UNIVERSITA' DEGLI STUDI DI PAVIA

FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL, COMPUTER AND BIOMEDICAL ENGINEERING

PHD IN ELECTRONICS, COMPUTER SCIENCE AND ELECTRICAL ENGINEERING
XXXV CYCLE - 2022

ARTIFICIAL INTELLIGENCE AND HIGH-PERFORMANCE COMPUTING FOR HUMAN-SENSIBLE APPLICATIONS: TOWARD PERCEPTIVE REAL-TIME DIAGNOSTIC SYSTEMS

PhD Thesis by

MARCO LA SALVIA

Advisor:
Prof. Francesco Leporati

PhD Program Chair:
Prof. Ilaria Cristiani

This one goes out to the oneS I love

This one goes out to the oneS I left behind

Acknowledgments

Here there will be the acknowledgments.

Abstract

This doctoral thesis presents a comprehensive study on the application of deep learning for a variety of vision tasks, with a focus on disease detection. The research covers developing and evaluating various deep learning architectures, ranging from ResNets to Vision Transformers, as well as traditional machine learning methods such as support vector machines and random forests. These approaches were applied to detect diverse diseases, including SARS-CoV-2, skin cancer, and brain cancer.

This doctoral thesis aims to deliver novel approaches for personalised medicine and human-sensible applications. A human-sensible application is designed to be easily understandable and interpretable by humans, allowing for more informed decision-making and improved communication between humans and machine learning systems.

The study's results demonstrate the potential of deep learning methods to achieve state-of-the-art performance in SARS-CoV-2 detection, epidermal lesions screening, and the assessment of brain cancer contours. In addition to developing these methods, this doctoral thesis also investigates the use of high-performance computing technologies to accelerate their implementation, including custom CUDA/C code. Real-time implementations of deep learning architectures for skin cancer and brain cancer were also developed and presented in this thesis.

The research presented in this thesis significantly contributes to artificial intelligence and deep learning for vision tasks. The development and evaluation of a wide range of deep learning architectures and the investigation of high-performance computing technologies provide valuable insights into the capabilities and limitations of these approaches. The real-time implementations developed in this study have the potential to significantly impact the speed and accuracy of disease detection in a clinical setting. Overall, the results of this research demonstrate the potential of deep learning and high-performance computing technologies to enable accurate and efficient disease detection for personalised medicine and human-sensible applications.

Contents

Summary of Notation

ACRONYM	EXTENDED NOTATION
AADO2	*ALVEOLAR-TO-ARTERIAL OXYGEN DIFFERENCE*
ABG	*ARTERIAL BLOOD GAS*
AI	*ARTIFICIAL INTELLIGENCE*
ALU	*ARITHMETIC-LOGIC UNIT*
ANNS	*ARTIFICIAL NEURAL NETWORKS*
ARDS	*ACUTE RESPIRATORY DISTRESS SYNDROME*
BCC	*BASAL CELL CARCINOMA*
BE	*BENIGN EPITHELIAL*
BGA	*BLOOD GAS ANALYSES*
BM	*BENIGN MELANOCYTIC*
CAD	*COMPUTER-ASSISTED DIAGNOSTIC*
CAM	*CLASS ACTIVATION MAPPING*
CCD	*CHARGE COUPLED DEVICE*
CE	*CROSS-ENTROPY*
CEIC/CEI	*COMITÉ ÉTICO DE INVESTIGACIÓN CLÍNICA- COMITÉ DE ÉTICA EN LA INVESTIGACIÓN*
CNNS	*CONVOLUTIONAL NEURAL NETWORKS*
CPAP	*CONTINUOUS POSITIVE AIRWAY PRESSURE*
CPU	*CENTRAL PROCESSING UNIT*
CT	*COMPUTED TOMOGRAPHY*
CUDA	*COMPUTER UNIFIED DEVICE ARCHITECTURE*
CUDNN	*CUDA DEEP NEURAL NETWORK LIBRARY*
CXR	*CHEST X-RAYS*
DCGAN	*DEEP CONVOLUTIONAL GAN*
DL	*DEEP LEARNING*
ECG	*ELECTROCARDIOGRAM*
ED	*EMERGENCY DEPARTMENT*
FDA	*FOOD AND DRUG ADMINISTRATION*
FID	*FRECHÈT INCEPTION DISTANCE*
FNRC	*FALSE NEGATIVE RATE PER CLASS*

FOV	*FIELD OF VIEW*
GA	*GENETIC ALGORITHM*
GANS	*GENERATIVE ADVERSARIAL NETWORKS*
GB	*GLIOBLASTOMA*
GPU	*GRAPHICS PROCESSING UNIT*
GUI	*GRAPHICAL USER INTERFACE*
HELICOID	*HYPERSPECTRAL IMAGING CANCER DETECTION*
HPC	*HIGH-PERFORMANCE COMPUTING*
HS	*HYPERSPECTRAL*
HSI	*HYPERSPECTRAL IMAGING*
ICCAS	*INNOVATION CENTER COMPUTER-ASSISTED SURGERY*
ICU	*INTENSIVE CARE UNIT*
IDE	*INTEGRATED DEVELOPMENT ENVIRONMENT*
IGS	*IMAGE GUIDED STEREOTACTIC*
IMRI	*INTRAOPERATIVE MAGNETIC RESONANCE IMAGING*
IOT	*INVASIVE VENTILATION*
IOU	*INTERSECTION OVER UNION*
IRCCS	*ISTITUTO DI RICOVERO E CURA A CARATTERE SCIENTIFICO*
KNN	*K-NEAREST NEIGHBOURS*
LDA	*LINEAR DISCRIMINANT ANALYSIS*
LUS	*LUNG ULTRASOUND*
LWIR	*LONG WAVE INFRARED*
MBFS	*MEAN BOUNDARY-F1 SCORE*
MDP	*MARKOV DECISION PROCESS*
ME	*MALIGNANT EPITHELIAL*
ML	*MACHINE LEARNING*
MLP	*MULTI-LAYER PERCEPTRON*
MM	*MALIGNANT MELANOMA*
MRI	*MAGNETIC RESONANCE IMAGING*
MSA	*MULTI-HEAD SELF ATTENTION*
MSC	*MELANOMA SKIN CANCER*
MSE	*MEAN SQUARE ERROR*
NIR	*NEAR INFRARED*
NLP	*NATURAL LANGUAGE PROCESSING*

NMSC	*NON-MELANOMA SKIN CANCER*
NNS	*NEURAL NETWORKS*
NPS	*NASOPHARYNGEAL SWABS*
NRES	*NATIONAL RESEARCH ETHICS SERVICE*
PCA	*PRINCIPAL COMPONENT ANALYSIS*
PSLS	*PIGMENTED SKIN LESIONS*
PTX	*PARALLEL THREAD EXECUTION*
PWM	*PULSE WIDTH MODULATION*
QTH	*QUARTZ-TUNGSTEN HALOGEN*
RBF	*RADIAL BASIS FUNCTION*
RELUS	*RECTIFIED LINEAR UNITS*
RFS	*RANDOM FORESTS*
RGB	*RED GREEN AND BLUE*
RNNS	*RECURRENT NEURAL NETWORKS*
ROC-AUC	*RECEIVER OPERATING CHARACTERISTIC AREA UNDER THE CURVE*
RT-PCR	*REVERSE TRANSCRIPTION-POLYMERASE CHAIN REACTIONS*
SAM	*SPECTRAL ANGLE MAPPER*
SCC	*SQUAMOUS CELL CARCINOMA*
SMS	*STREAMING MULTIPROCESSORS*
SSL	*SELF-SUPERVISED LEARNING*
SVMS	*SUPPORT VECTOR MACHINES*
SWIR	*SHORT WAVE INFRARED*
TL	*TRANSFER LEARNING*
US	*ULTRASOUND*
VIT	*VISION TRANSFORMERS*
VNIR	*VISIBLE NEAR INFRARED*
WHO	*WORLD HEALTH ORGANIZATION*

Chapter 1

Introduction

Nowadays, researchers all over the globe would undoubtedly agree on defining Artificial Intelligence (AI) as the buzzword of the last decades. This expression progressively flooded scientific journals, leading to technological advancements in various contexts and impressive experimental outcomes[1]. Academics started to conceive the mathematical models behind AI in the 40s, and its first definition only came in 1956 from John McCarthy. During those years, Alan Turing proposed the question: *can machines think?* Although we probably can still not answer, we refer to AI as the set of algorithms and models inspired by the human brain to mimic its intelligence[1–6]. We can offer two very well-known statements. The first, coined by Arthur Samuel, defines *AI as the field of study that allows computers to learn without being explicitly programmed*. Tom Mitchell provides a more modern and formal interpretation where *A computer program is said to learn from experience E concerning some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E*[2]. Currently, as Figure 1 reports, by mentioning AI, we might refer either to Machine Learning (ML) or Deep Learning (DL), depending on the specific algorithm or mathematical model involved in accomplishing the desired task.

Figure 1. Definitions of Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL)

Among the various research fields to which academic authors tried to apply AI, scientific investigators early identified medicine as one of the most promising. Its community has been taking advantage of these extraordinary developments. Indeed, researchers started designing AI applications that get the most out of medical images and clinical data, automating different healthcare steps to support clinical decisions[1,3–6]. Looking back at the 1970s, scientists first developed rule-based approaches to interpret medical imaging, diagnose diseases, and choose appropriate treatments aiming to assist physicians in generating diagnostic hypotheses in complex patient cases. Nevertheless, rule-based systems are costly to build and might present flaws, as they require an explicit definition of decision laws and human-computer interactions. Also, it was challenging to encode higher-order interactions among different pieces of knowledge authored by different experts, and the explainability of prior medical knowledge constrained the performance of the systems[7–10].

In recent decades, AI applications have experienced unprecedented breakthroughs, especially in computer vision, namely the field of artificial intelligence researching how to teach a machine to interpret and comprehend the visual world[2]. Notably, starting from any kind of imaging, we want to teach computers to identify visual patterns from data and react to what they see. In medicine, this translates to interpreting the visual data gathered from radiology examinations such as X-rays, ultrasound, and Computed Tomography (CT)[1,3,4,6]. Unlike the first generation of AI systems, which relied on the translation of medical knowledge into robust algorithmic rules by experts, recent AI research has leveraged machine and

deep learning methods, which can account for complex interactions and identify patterns from the data[1,3]. The ongoing crucial technological development supports this innovation. Healthcare research requires hardware capable of elaborating enormous amounts of data and mathematical models in real-time to comply with surgical operations or provide fast answers to patients to treat them according to the highest standards[3,11]. High-Performance Computing (HPC) technologies, namely systems comprising multi and many-core processors to spread the computational load and reduce elaboration times, play a significant role in this sense[11–13]. Indeed, it enables academic investigators and scientists to change how we think about medicine, designing the so-called *human-sensible applications* for *personalized medicine*[11]. Indeed, as Figure 2 reports, the number of scientific publications concerning AI projects in medicine astoundingly increased across the years thanks to the HPC advancements sustaining the complex computational elaborations required[1,5]. The researchers aim to design perceptive systems to diagnose diseases in real-time to provide specific therapies that better suit different patients[11,12].

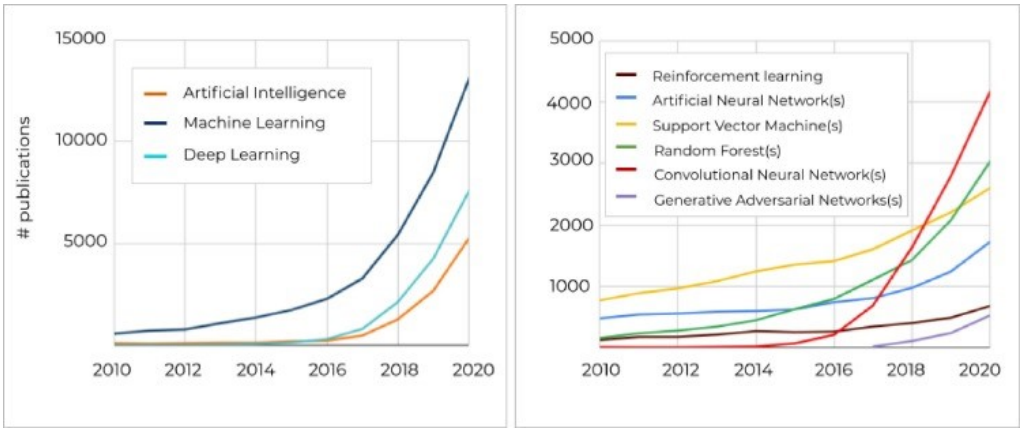


Figure 2. Number of publications from 2010 to 2020 containing keywords related to AI, ML or DL methods[1]

More recently, the pandemic further stressed the need for AI-based diagnostic systems and intelligent medical devices, accelerating the technological advances we all experience. Indeed, the SARS-CoV-2 (i.e., Covid-19) outbreak in 2020 challenged health systems worldwide, eliciting an urgent need for effective and highly reliable diagnostic instruments to help medical personnel. Researchers designed and developed different solutions ranging from Covid-19 positivity assessment to imaging analytics to evaluate patients' conditions[14–17].

Nonetheless, the data science process, from data collection to model deployment on HPC hardware, needs a rigid structure to allow AI instruments' fast-paced design, development, and release. Figure 3 reports

the process as composed of steps requiring diverse professional figures. Indeed, data engineers and analysts collect, clean, and explore the data to enable researchers in the pipeline to design different models later deployed on HPC hardware. The HPC hardware might also be embedded to fit better the ingenious devices supported with AI[11,18,19].

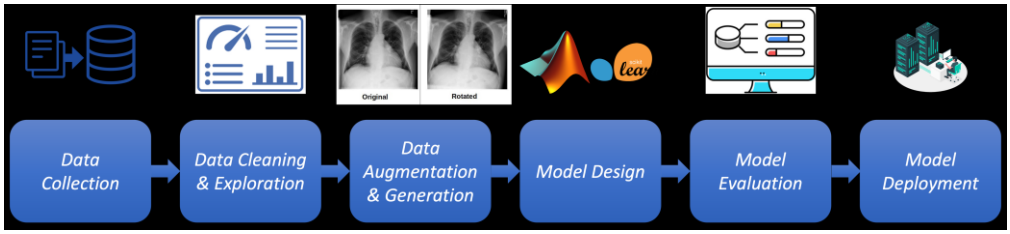


Figure 3. The basic steps of a data science project: from data collection to model deployment

The educational path reviewed in this doctoral thesis describes a collection of works following the data science process reported in Figure 3. Namely, we will address projects concerning different data types and sources, ranging from the HypErspectraL Imaging Cancer Detection (HELICoiD) brain cancer images[11,12] to the clinical and radiology data related to the SARS-CoV-2 disease[15]. We will describe how the data were first collected, cleaned, and explored to gather knowledge to support decisions. Then, we will understand how to enlarge the statistical variance of the information at our disposal, envisioning either standard *data augmentation* techniques or *generative models* that make AI systems robust, enabling disturbance rejection to adversarial attacks. The European Commission laid down harmonised rules on artificial intelligence, defining the so-called *AI act*. The AI act declares that cybersecurity is essential in guaranteeing that AI applications are resilient against endeavours to alter their service, behaviour, and performance or compromise their safety properties by malicious third players manipulating the system's vulnerabilities. Cyberattacks against AI systems can leverage AI-specific assets, such as trained models introducing *adversarial attacks*, which provide the optimised architecture with slightly different input and confound its behaviour. Accordingly, suitable measures should be taken by the providers of high-risk AI systems, considering the underlying infrastructure as appropriate, to ensure a level of cybersecurity appropriate to the risks. In this context, data augmentation is crucial since it prevents adversarial attacks on input data as much as possible.

This doctoral thesis analyses a collection of machine and deep learning models trained on various datasets using different algorithms and architectures. To determine their strengths and weaknesses, we will assess these models according to specific performance metrics, such as accuracy, precision, recall, and F1 score. We will also compare the results of these models with traditional machine learning models to gain a better

understanding of the benefits and limitations of deep learning. As we move forward, we'll also consider the importance of GPUs, especially during deep learning models' training and inference stages, as they are critical for their performance. With many cores that perform parallel computations, GPUs allow for faster training and inference times than a CPU alone, which is particularly important when dealing with large datasets and complex models that can take days or even weeks to train on a CPU.

Additionally, in medical contexts where policies are in place to ensure the privacy of patients, using a GPU or multi-GPU system is better than using cloud clusters. One of the main advantages of using a local GPU system is the ability to store and process patient data on-site, eliminating the need to transfer sensitive patient data to a cloud server. Furthermore, using a local GPU system is often more cost-effective than cloud clusters, especially when working with limited research budgets. Finally, in surgical scenarios where researchers must meet real-time constraints, using a local GPU system is critical, as latencies associated with data transfers can be detrimental to the outcome of the surgery. Using a local GPU system allows researchers to perform real-time analysis and make decisions quickly, which is crucial, for instance, to address the investigations in the brain cancer section of this doctoral thesis[11,13,18].

Medical AI applications, as they mature, face many challenges[3,5]. First, clinical contexts often offer poor datasets to work with, making it hard to exploit complex DL architectures. Not only do these architectures require vast amounts of data to extract functional patterns, but they also need HPC hardware. Indeed, the more complex the model, namely presenting larger structures with many parameters, the more we need to employ performing hardware to carry out computations. Likewise, medical data present challenges specific to its domain. For example, different experts may deliver contrasting opinions regarding a diagnosis. Consequently, we must set hierarchically structured and standardized evaluations to enable AI-based diagnosis. Finally, from a regulatory perspective, clinical AI systems must be certified before large-scale deployment[3,5,6]. Therefore, the manuscript proposes contemporary model deployment approaches employing hardware, Nvidia GPUs, and programming frameworks to accelerate the algorithms and allow the design of blueprints. Modern AI must translate to blueprints for regulatory entities such as the Food and Drug Administration (FDA) or the European Commission to evaluate to meet certifications currently under investigation in *AI acts*, including time-sensitive and performance criteria. Regulators have struggled to interpret existing frameworks concerning perceptive algorithms, whose functioning can change with ongoing training and optimization and whose output we often cannot clearly explain[3,6]. Many clinical applications of AI are seeking regulatory approval. For instance, a deep-learning system for diagnosing cardiovascular diseases using cardiac MRI images was approved by the FDA in 2018[1,3–6]. The government's new interpretation is a substantial step toward rules that protect patients without inhibiting innovation. These

represent only a few challenges we will encounter in this manuscript and proposed solutions.

This doctoral thesis addresses artificial intelligence applied to various medical data, especially Hyperspectral Images (HSIs), and matured around the state of the art. Notably, deep learning novel approaches were designed when literature only proposed standard machine learning processes. Not only was a robust artificial intelligence methodology applied to the medical context, but novel GPU approaches and frameworks were also engineered and implemented to embed or simply accelerate the designed models (i.e., HPC), complying with the time-sensitive criteria required for industry translation (e.g., real-time requirements). The work presented in this manuscript was carried out thanks to tight and robust collaborations outside the academy, particularly with the Fondazione IRCCS Policlinico San Matteo of Pavia, the University of Las Palmas de Gran Canaria and the Innovation Center Computer-Assisted Surgery (ICCAS) of the University of Leipzig.

Chapter 2 addresses medical imaging and the data types employed in this doctoral research. Furthermore, in Chapter 3, we explore artificial intelligence fundamentals, from the fundamental building blocks toward the architectures and frameworks used during the doctoral school. Chapter 4 presents the HPC technologies' relevance in developing the algorithms designed and employed to produce accelerated algorithms embedded into novel medical instruments. Specifically, we will address the description of hardware devices and the CUDA language together with its library extensions, providing syntax definitions and some examples. Finally, Chapters 5, 6 and 7 will go through the AI studies, grouped by disease type. Namely, we address studies to counteract SARS-CoV-2 in Chapter 5, epidermal tumours assessment in Chapter 6 and brain cancer contours delineation in Chapter 7. Finally, the thesis drives the conclusions from the works described in this manuscript and their comprehensive discussion.

Chapter 2

Healthcare applications: medical imaging and clinical data

AI lately re-emerged into public consciousness and applied science corporations and researchers revealed breakthroughs and new technologies at an extremely high rate[1,3,5,6]. Despite its science-fictional characteristics, AI is a branch of computer science attempting to understand and design perceptive entities written as software programs[3]. The booming growth of image classifiers has contributed to the recent renewal of AI since 2012, gradually transforming the geography of healthcare and biomedical research. Despite AI's progress during these years, it sorrowed from varying definitions. People in the 70s and 90s considered automated route planners and interpretations for electrocardiograms (ECGs) as examples of advanced AI[1,3]. Yet, they are so everywhere that most people would be surprised to think of them as true AI. Applications of medical-image diagnostic systems have expanded the frontiers of AI into areas previously mastered only by human experts. This boundary continues to expand into other areas of medicine, such as clinical practice and biomedical research[1,3,5].

Automated medical-image diagnosis is arguably the most flourishing discipline of healthcare AI. Many specialities, including radiology, ophthalmology, and dermatology, rely on image-based diagnoses. Notably, the radiological practice leans primarily on imaging for diagnosis and thus fits AI techniques, as images comprise the information needed to arrive at the proper treatment description of hardware devices and the CUDA language with syntax and some examples [1,3–6].

This chapter will go through the various data types addressed in this doctoral thesis. Indeed, data curation and exploration are the first steps toward the invention of AI technologies in healthcare. Chapter 2 will describe the data source, what application this manuscript used it for, how we interpreted the information, and the dataset gathered from its collection.

2.1. Lung UltraSound

An ultrasound (US), also known as a *sonogram*, is an imaging examination that operates sound waves to assemble a picture of organs and tissues. Ultrasound does not imply the presence of radiations compared to x-rays and can also show moving body parts, such as a beating heart[20].

In this thesis, we focus on the diagnostic ultrasound[15,20]. Namely, the examination physicians use to collect knowledge about internal body regions, including the heart, blood vessels, liver, bladder, kidneys, and lungs.

During the last decade, Lung US dramatically increased its popularity. Physicians routinely perform it at the patient's bedside, especially in the hospitals' emergency departments (EDs) and in intensive care units (ICUs) [15,20]. Radiation-free lung ultrasound (LUS) requires high expertise, and it is, therefore, underutilised. Indeed, it requires formal training, which radiology residence education does not often include. It demonstrated a strong correlation with CT scan results and high reliability in pneumonia detection, even in the early stages. Therefore, the diagnostic radiologist should be fluent in LUS execution and understanding [15,20].

Lung US is radiation-free, low-cost, fast, and portable, allowing real-time investigation of pulmonary regions. Data analyses indicate that it has higher sensitivity and similar specificity for detecting miscellaneous complications such as pneumonia and infections compared with chest radiography [15,20]. It is increasingly operated in the ICUs and EDs to detect these diseases. Critical care providers have adopted the bedside lung US protocol as a standardised approach to assessing patients quickly[21,22]. Indeed, many practitioners advocated for the regular use of LUS to decrease chest radiography employment, which is associated with increased cost and nontrivial cumulative radiation exposure, especially in pediatric patients. Lung US also has a well-established role in guiding interventional processes, especially in paediatrics where it is employed for the enhanced visualisation of irregularities in the thorax due to the small chest diameters of children and the absence of ionising radiation to assemble diagnostic imaging outcomes[20,21].

Concerning LUS's general functioning, it comprises several limitations[20,21,23]:

- It is operator-dependent, and its quality and interpretation vary by expertise. Indeed, advanced technical skills and clinical understanding increase the diagnostic yield. Nonetheless, studies have shown that healthcare providers can learn the essentials of the modality with relative ease
- Lung ultrasound requires up to 20 minutes to perform the examination, whereas physicians can complete chest radiography in fewer minutes

- Lung US bears artefact-based interpretation, and its findings, namely *A-lines*, *B-lines*, and *consolidations* can be observed in various conditions (Figure 4)

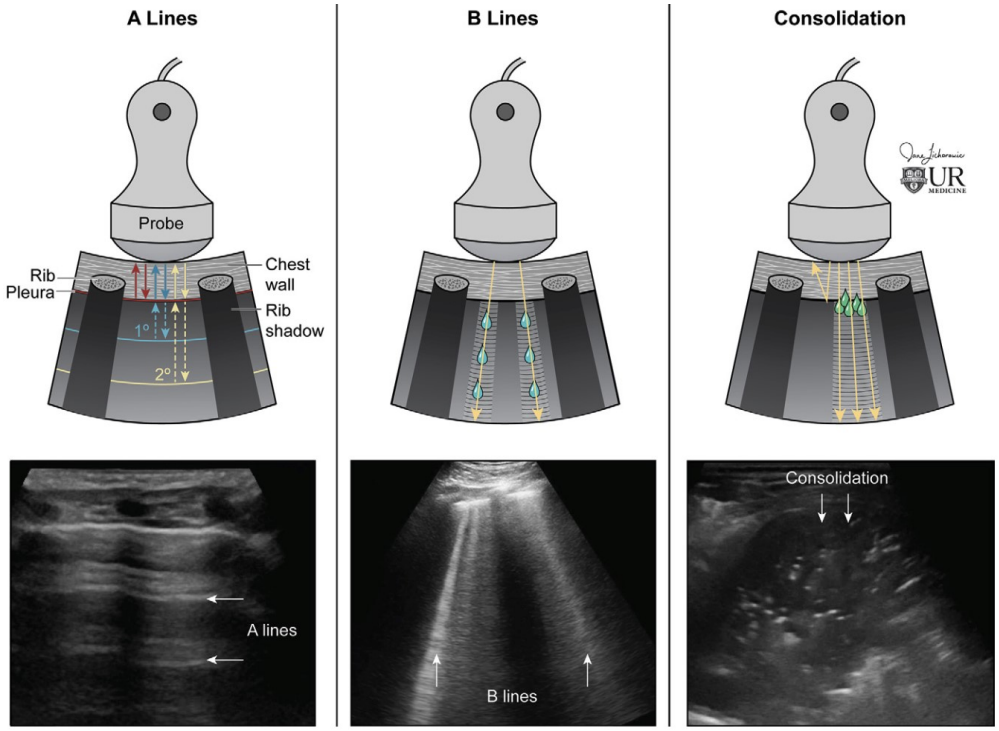


Figure 4. Artefacts derived from lung ultrasound (LUS) examination. From left to right we observe complete lung aeration towards no aeration[20,21,23]

Complete lung US involves examining each hemithorax in the anterior, lateral, and posterior lung zones (Figure 5)[23]. Physicians should also investigate all lung fields in transverse and longitudinal directions to avoid missed abnormalities. In this setting, patients can receive the examination both in the supine and upright positions.

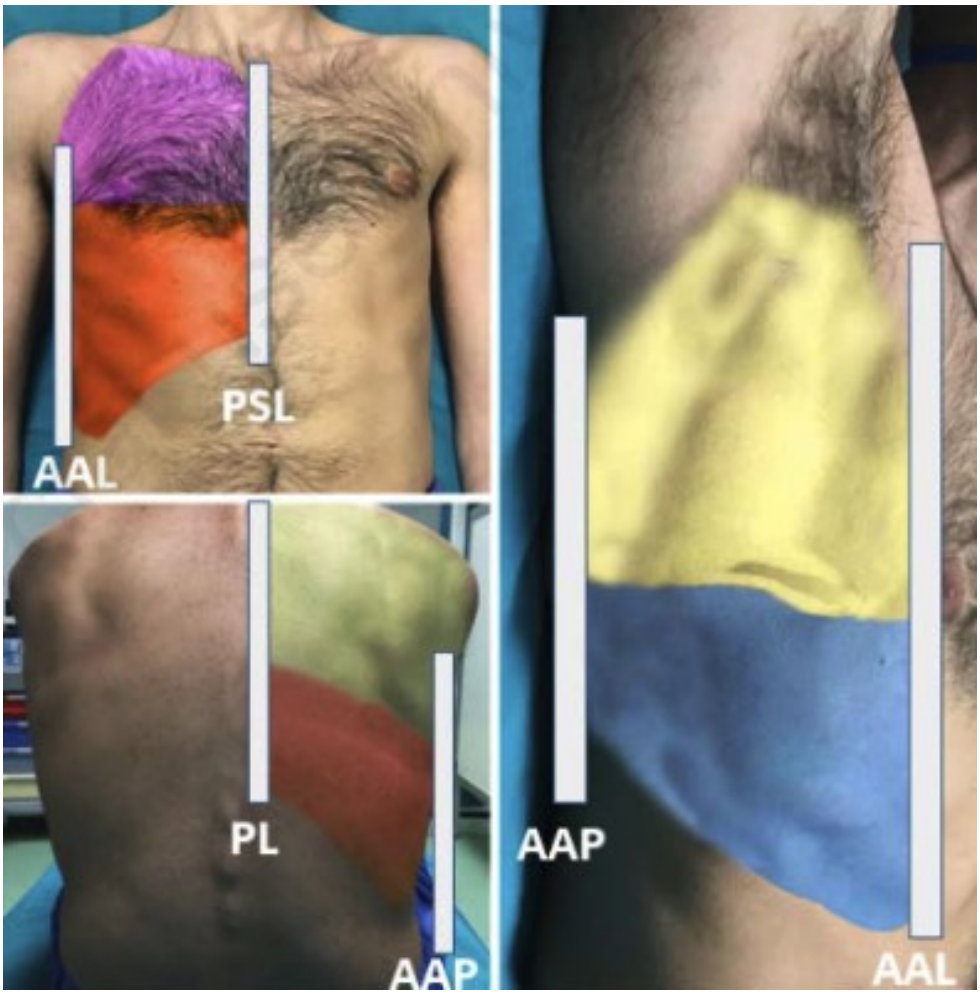


Figure 5. Hemitorax division into ParaSternal Line (PSL), Anterior Axillary Line (AAL) and Posterior Axillary Line (AAP)[23]

Proper technique is crucial to provide acceptable imaging, and since LUS is artefact-based, investigators must hold the probe on the skin to ensure perpendicular orientation to the pleural line to collect them for interpretation[20,21,23]. Indeed, chest ultrasound is unique among other US examinations because it generates artefacts instead of direct anatomy observation[20,21,23]. Most ultrasound waves are reflected at the pleura in an air-filled lung owing to the acoustic impedance mismatch at the air and soft-tissue interface, resulting in a hyperechoic pleural line. Consequently, observers can report A-lines for interpretation, namely horizontal reverberations of the hyperechoic pleural line reflected from the air-filled lung (Figure 4). The lung interstitium might present thickening, for instance, due to pulmonary oedema, interstitial inflammation, or infection [20,21,23]. In this case, B-lines, hyperechoic vertical lines traversing the imaging field below the pleural line, replace the normal A-lines (Figure 4). Lung collapse or consolidation removes the A-lines and allows for direct visualisation of the parenchyma (Figure 4). Physicians can also directly report pleural effusions, both complex and straightforward. Regardless, if a

pulmonary irregularity does not touch the pleural line, it does not generate artefacts due to the acoustic impedance mismatch between the air and soft-tissue interface. Fortunately, most clinically significant anomalies, especially life-threatening ones, border the pleural line, allowing its detection.

2.2. Lung UltraSound in the SARS-CoV-2 pandemic

SARS-CoV-2 (Covid-19) originated in China and has abruptly scattered in Europe since February 2020[15,24,25]. It rapidly dispersed worldwide and still challenges health systems. It manifests after a long incubation period and a high contagion rate, thus necessitating the development of fast and cheap diagnostic tools to detect infected subjects[15,16,23]. Moreover, it can cause bilateral multifocal interstitial pneumonia, rapidly evolving into acute respiratory distress syndrome (ARDS), responsible for generating hundreds of thousands of deaths worldwide[26]. Subjects infected by SARS-CoV-2 may present an evolving clinical picture ranging from focal to multifocal interstitial pulmonary involvement that LUS may visualise in the so-called white lung pattern, as well as by bilateral submantellar-subpleural consolidations. The high contagion rate adds a further level of complexity because patient care, according to the highest healthcare standards, must be combined with strict pandemic protocols that healthcare professionals must follow for their safety[14,26].

Currently, the main diagnostic tools for detecting infected people comprise reverse transcription-polymerase chain reactions (RT-PCR) in nasopharyngeal swabs (NPS) and IgM-IgG integrated antibody tests. However, both these tools present drawbacks[24,27,28]. The first does not reach a 100% sensitivity, introducing false-negative outcomes, one of the causes of the inaccurate partition of patient streams in hospitals. Likewise, it is time-consuming, and when the number of infected subjects increases, unavoidable shortages in reagents and other laboratory reserves occur, precluding test completion. IgM-IgG tests not only exhibit the same poor sensitivity, with a slight increase only after a specific duration following symptom manifestation but also may result in false negatives in the early phases of the infection. Covid-19 forms with mild or no manifestations but can rapidly transform into highly critical conditions with possibly fatal consequences due to multi-organ failure. Therefore, it is vital to promptly and reliably detect infected subjects to apply the appropriate treatments early and prevent the virus from spreading. Moreover, no swab tests can describe the presence or severity of lung engagement[14,26,29].

First-line diagnosis of pneumonia comprises chest X-rays (CXR) for first-aid treatment of patients exhibiting symptoms of pneumonia[30]. Potential alternatives to CXR include computed tomography (CT) scans and lung ultrasound (LUS)[20,21,23]. Breakdowns concerning these procedures state that LUS and CT scans are significantly better first-line diagnostic tools than CXR, whose main drawback is poor sensitivity. However,

although ultrasonography is a cost-effective, radiation-free, and promising tool, highly skilled radiographers must perform it to achieve accurate results. LUS effectively performed at the bedside in approximately 13 min yielding a higher sensitivity than CXR. Thus, it is comparable to but cheaper than CT imaging tools. Moreover, LUS is easier to disinfect and can be repeated even with short time intervals between two observations, while the same is not true for the other methodologies[20,21,23]. However, it has certain drawbacks, such as operator dependency and high expertise requirements, resulting in underutilisation, and it may not be useful for Covid-19 asymptomatic patients.

In the following, Chapter 2 analyses the main patterns arising from LUS examination, especially in conditions such as SARS-CoV-2[20,21,23]:

- **The A-line artefact in healthy air-filled lungs**: as Figure 4 reports, the pleural line is continuous and regular, and A-lines arise as horizontal artefacts owing to the high reflectance of the aerated lung surface. Hence, multiple reflections appear between the probe and the lung surface. In addition, during respiration, the sliding visceral and parietal pleura causes the shimmering motion of the pleural line, referred to as lung sliding. However, pathologic conditions like asthma and pulmonary embolism with air-filled lungs also have A-line artefacts. In addition, a pulmonary infarct appears as a consolidation abutting the pleural surface
- **Interstitial Thickening**: when the pulmonary interstitium thickens, B-lines replace the normal A-lines, consisting of well-defined, laserlike, vertical, echogenic beams arising from the pleural line and extending to the bottom of the image. Scattered B-lines, namely fewer than two per intercostal space, can be present in healthy lungs. The number of B-lines directly correlates to disease severity
- **Infection**: LUS is excellent for assessing suspected pulmonary disease. Indeed, pneumonia has several imaging formations depending on the consolidation area or interstitial involvement. For instance, a thoroughly consolidated lung emulates the solid appearance of a liver (Figure 4). In a consolidation, fluids fill the alveoli, removing the normal A-lines. Therefore, observers report lungs characterised by dense and broadly extended white lung areas with grand coalitions. Physicians report this severity level when the lung presents tissue-like patterns, namely wide, thick, and dark consolidations

During the SARS-CoV-2 pandemic, healthcare professionals reported B-line artefacts of varying severity, consolidations, and pleural irregularities in Covid-19 infection (Figure 4). In areas of focal ground-glass opaqueness, diffuse B-lines arise with a casualty of A-lines.

2.3. The LUS SARS-CoV-2 database

Since March 2020, the Fondazione IRCCS San Matteo Hospital's ED of Pavia has been collecting LUS data to assess patients affected by Covid-19. The personnel utilised the ultrasound machine Aloka Arietta V70 (Hitachi Medical Systems), providing convex and linear probes at 5 MHz and 12 MHz. They standardised the acquisition process through abdominal settings, focusing on the pleural line, reaching a depth of 10 cm with the convex probe. Moreover, they accommodated the gain to acquire the best possible imaging of the pleura, vertical artefacts, and peripheral consolidations with or without air bronchograms. Physicians performed complete longitudinal and transversal scans to explore the entire pleural length, disabling all harmonics and artefact-erasing options[15,23].

Physicians performed LUS on people with clear clinical circumstances due to the RT-PCR test introducing many false negatives. Namely, the artefacts comprised either pulmonary oedema or non-cardiac causes of interstitial syndromes[15,23]. Although many people presented a negative RT-PCR test, subjects manifesting lung involvement are highly likely to be Covid-19 positive. Physicians are used to differentiating suspicious from healthy subjects following a triaging procedure involving LUS investigation.

Hereafter, we define a clip as the result of an LUS examination. It consists of frames (i.e., images) that this doctoral thesis operated. The proposed definition grant continuity regarding observations in other similar works[15].

Table 1. Lung UltraSound scores description[15,23,31]

Severity Score	**LUS Score**
Score 0	A-lines with at most two B-lines
*Score 0**	A-lines, and at most two B-lines, with a slightly irregular pleural line
Score 1	Artefacts occupy at most 50% of the pleura
*Score 1**	Artefacts occupy at most 50% of the pleura and present a damaged pleural line
Score 2	Artefacts occupy more than 50% of the pleura, while consolidated areas may be visible
*Score 2**	Artefacts occupy more than 50% of the pleura, while consolidated areas may be visible. The pleura is either damaged or irregular
Score 3	Tissue-like pattern

The hospital's medical personnel collected 12 clips for each patient, all assigned with a standardised LUS score (Table 1), one for each chest portion, as depicted in Figure 5. The ED collected data from 450 patients, whose clinical information is in Table 2, and were treated in Pavia, and gathered 5400 clips. Table 2 lists the subjects split into Covid-19 positive and negative, and the clinical data through median and 25th–75th percentile values. The LUS Score entry indicates the sum of the values collected for each patient who received 12 examinations[15,23,31].

Table 2. Fondazione IRCCS Hospital patients' clinical information[15,17]

	Negative (172)		**Positive (278)**		**Total (450)**	
	Median	25 - 75 P	Median	25 - 75 P	Median	25 - 75 P
Age (years)	54	37.0–67.5	63	51.0–75.0	60	47.0–73.0
Systolic blood pressure (mmHg)	135	125.0–150.0	130	115.5–144.0	130	120.0–145.0
Diastolic blood pressure (mmHg)	80	70.0–90.0	80	70.0–85.8	80	70.0–90.0
Respiratory rate	20	16.0–22.0	20	16.0–26.0	20	16.0–24.0
Oxygen saturation (%)	97	94.0–98.0	94	90.0–97.0	95	91.0–98.0
Body temperature (°C)	36.7	36.2–37.6	37.1	36.5–38.0	37	36.3–37.9
Hemoglobin (g/dL)	13.5	12.2–14.9	13.9	12.8–14.9	13.7	12.6–14.9
White blood cell (10^9/L)	8.2	6.3–11.5	6.3	4.8–8.1	6.92	5.1–9.2
Lymphocytes (10^9/L)	1.555	0.9–2.2	0.8	0.6–1.1	1	0.7–1.6
Platelets (10^9/L)	224.5	179.5–272.5	184	146.0–239.0	204	157.0–256.7
C-reactive protein (mg/dL)	1.325	0.1–10.5	7.97	2.6–15.2	5.29	0.9–14.4
Lactate dehydrogenase (U/L)	222	182.0–290.0	326	243.5–428.0	286	211.2–399.7
Creatine phosphokinase (U/L)	86	51.0–143.0	113	68.0–293.5	99	62.0–217.7
PH	7.4	7.4–7.4	7.4	7.4–7.4	7.44	7.4–7.4
PaO2/FiO2	392.1	317.5–462.9	299.5	226.4–352.7	323.8	256.0–405.8
Alveolar-arterial gradient of O2 (mmHg)	22.4	9.5–42.5	47.3	33.6–93.1	40.4	20.8–60.8
LUS Score	2	0.0–7.5	11	6.0–16.0	7	2.0–13.0

First, physicians assigned a clip with Score 0 when the pleural line was continuous and regular, and A-lines were present as horizontal artefacts due to the high reflectance of the aerated lung surface. Hence, multiple reflections appeared between the probe and the lung surface.

Next, the personnel defined Score 0* as any clip evaluated as Score 0 but with an irregular or slightly damaged pleural line.

Furthermore, the severity level increased when either vertical areas of white or consolidations were visible (Score 1). These white regions are due to local alterations in the acoustic properties of the lung. Namely, the lung volume, previously aerated and healthy, transformed into a tissue or water-like entity. This process demonstrates the formation of perpendicular artefacts. The physicians ranked the clip with a Score of 1 when observing these artefacts for less than 50% of the pleura.

In addition to the introduction of Score 0*, they defined Score 1* as a recording that would have typically been assigned Score 1 but had an irregular or shattered pleural line. Clearly, the higher the score, the greater the injury detected upon pleural investigation.

Furthermore, specialists generally estimate a patient's lung as Score 2 if more significant consolidated areas (i.e., dark portions) appear along associated regions of white below solidifications. This pattern typically leads to the white lung[20,21,26,31]. Dark and dense partitions suggest a shift in the tissue and its acoustic characteristics toward a situation commonly reported when scanning soft tissue. Regardless, the formation of white and large zones indicates a not fully ventilated lung: air is still present but embedded in tissue-like compounds. The medical personnel marked a recording with this score when the artefacts scattered on more than 50% of the pleura, and they observed both small and bounded consolidations, demonstrating a more acute stage of the infection.

Similarly, the specialists designated a Score 2* when the pleura was either irregular or injured in a lung that would have typically received a Score 2.

Finally, lungs characterised by dense and vast developed white lung regions with abundant consolidations received a Score of 3. This severity level describes lungs presenting tissue-like patterns, namely widely thick and dark consolidations.

Nevertheless, not all clips obtained an LUS score from the same medical practitioner. Therefore, we further reviewed the collection to validate the classifications and avoid incorrect severity-scoring problems. This process was mandatory to ensure that each clip had a standardised LUS score and that there were no discrepancies in the scores assigned to different clips, which are problems stressed in other studies[15].

Fondazione IRCCS Policlinico San Matteo ED's physicians observed the methodological procedure and ensured that the labelling was correct. During the first part of the collection and annotation process, they manually selected all clips from each patient, assessed the quality of each clip, and proceeded to evaluate it according to the aforementioned scoring. They reviewed each clip to assign a score and verify that SARS-CoV-2 pneumonia patterns were present.

2.4. Hyperspectral images

Hyperspectral imaging (HSI) is a non-invasive, non-ionising and label-free technique conceived originally for remote-sensing and military intents[32–34]. Thanks to technical refinements, HSIs evolved in applications in different fields such as archaeology and aerospace. At first, only a few corporations and academic institutes operated HSIs because acquisition equipment and computational systems were costly. Contemporary technological progress allowed the widespread use of hyperspectral images in many fields, becoming popular, especially in medicine, for cancer detection[33–36]. Hyperspectral (HS) images measure the reflected and transmitted light, gathering light-matter relations values associated with several bands (i.e., wavelengths) of the electromagnetic spectrum[13].

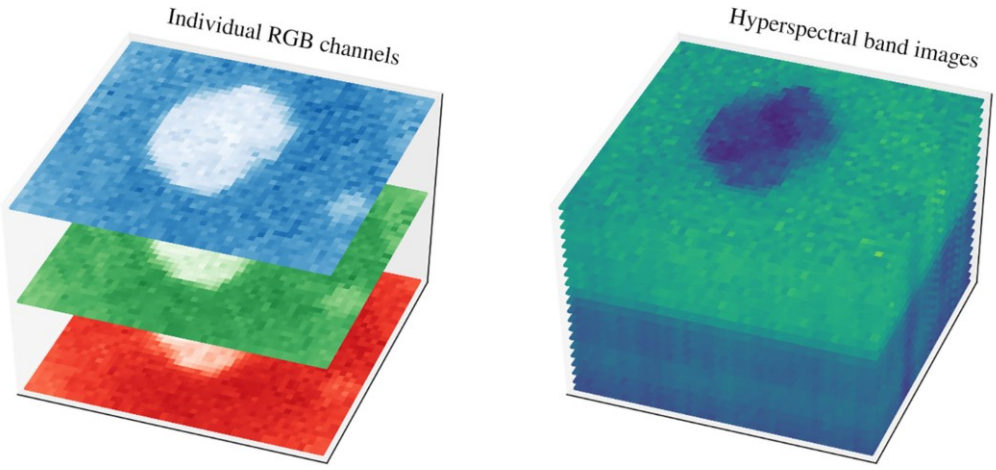


Figure 6. The difference between the information richness in a hypercube and an RGB image[37]

Multiple shots aligned in neighbouring narrow wavelengths form the HS image, constituting a reflectance spectrum of all the pixels[37]. Therefore, the outcome is the HS cube in Figure 6 retaining both the spatial and spectral information of the analysed sample. Several studies discussed tumour cells exhibiting unique molecular spectral signatures and reflectance values[32,34,37,38]. Researchers exploit the light-matter physical interaction, causing each material or tissue to react differently to the beam radiation on its surface, owing to its molecular structure, allowing the discrimination between healthy and tumour tissue. Therefore, the hyperspectral image, also known as a *hypercube*, is a three-dimensional dataset having a two-dimensional image at each wavelength. Researchers appreciate the light-matter reaction in the pixel electromagnetic spectrum, which provides detailed information concerning the observed area represented by the pixel[34].

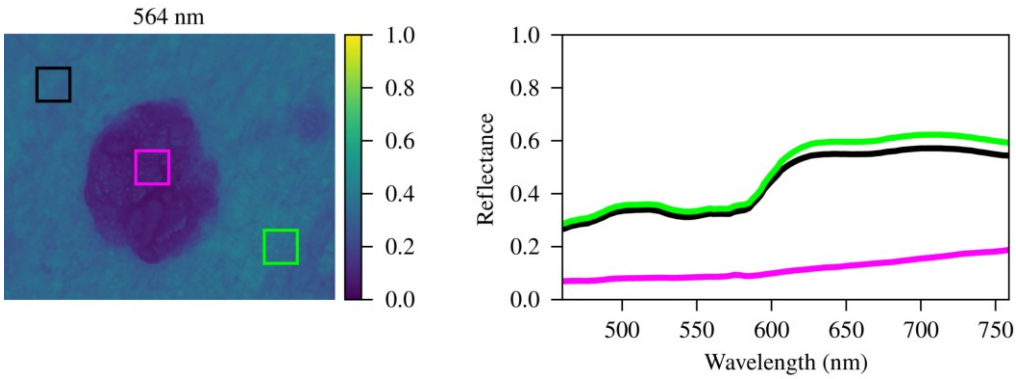


Figure 7. Examples of spectral signatures. The image on the left displays an epidermal lesion whose coloured squares relate to the average spectral signatures in the right plot. The mean curves are calculated based on all pixels in each region[37]

HS cubes differ from classical Red, Green and Blue (RGB) images due to the higher number of bands characterising them. Indeed, RGB photographs present only three bands, whilst hypercubes comprise hundreds of bands per pixel. Figure 7 highlights their differences in evaluating the reflectance curve (i.e., spectral signature)[11].

During the last decade, machine and deep learning (ML, DL) solutions emerged as a tool to analyse and cluster different cancer types in HSIs. Academics hardly interpret HSIs as they are structured, so researchers usually carry out hyperspectral image analysis via ML approaches. Among the medical subjects concerning ML and HSIs, literature focused on brain, skin, colon, and oesophageal cancer[11,13,32,34,39]. This chapter explores why this procedure presents medical advantages, especially for cancer detection. Namely, biochemical and morphological changes associated with lesions modify the optical characteristics of tissues, such as light absorption, scattering and fluorescence, providing valuable diagnostic information and allowing automatic cancer detection through HSIs. Despite the presence of other imaging techniques, such as optical spectroscopy, hyperspectral cameras can capture larger areas and deliver more accurate results in detecting cancers in the cervix, breast, skin, and brain cancer[11,13,18]. The different techniques mainly differ in the acquisition system setup, the nature of the samples, whether they are in-vivo, ex-vivo or in-vitro, the considered disease, and the classification algorithm.

2.5. Hyperspectral cameras

A spectrometer is an instrument that measures the electromagnetic field, namely an object that divides the collected light into a *spectrum*. Hyperspectral imaging uses a spectrometer to collect spectral information, and this device is called a hyperspectral camera.

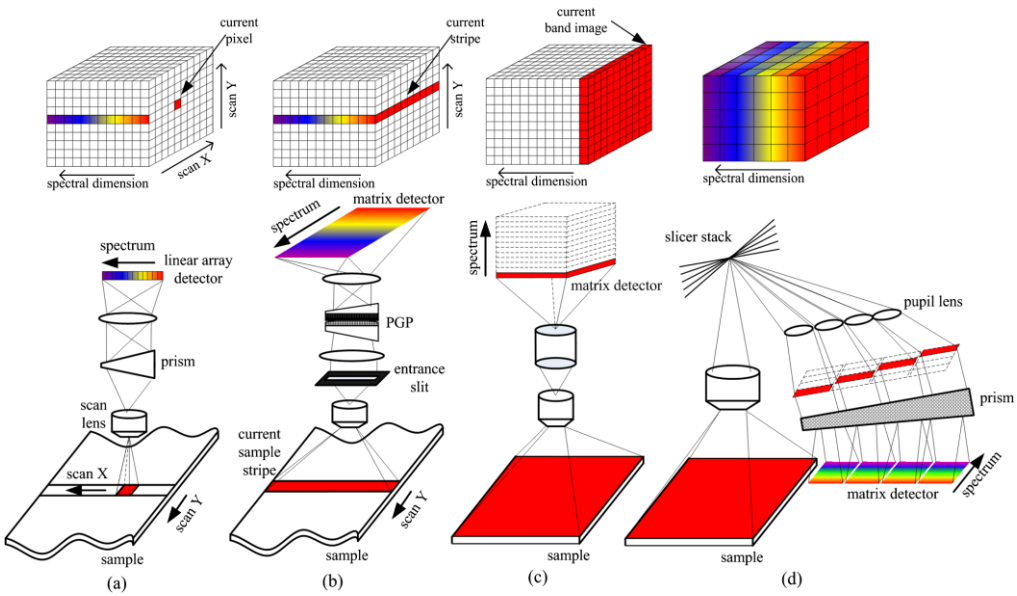


Figure 8. Typical spectral imaging approaches. (a) Whiskbroom. (b) Pushbroom. (c) Staring. (d) Snapshot[40]

These cameras collect information concerning hundreds of spectral bands with continuity over the entire spectrum of interest. Various acquisition techniques exist to obtain a hyperspectral image, and the most used are whiskbroom, pushbroom, staring and snapshot.

The whiskbroom mode, used initially for satellites and known as the *point scanning method*, allows the reflected light's guidance through rotating mirrors towards a group of detector sensors. As shown in Figure 8.a, the single point is scanned along the X or Y direction by moving the sample or detector, a prism scatters the reflected light, and a linear array detector records the spectrum. The hypercube (X, Y, λ) originates by acquiring the scene in the X and Y dimensions while collecting the wavelength domain (λ). Since it involves separate image acquisition along the two spatial dimensions, the whiskbroom technique requires a complex hardware configuration and a high scanning time.

The pushbroom method, also known as *line scanning*, allows the grouping of no longer a single point but a line, with one spatial and one spectral dimension at a time. In this technique, light gathers through a collimated slit, then scatters on a 2D matrix detector, displaying spatial information along one axis and wavelength information along the other (Figure 8.b). The 3D data cube forms by moving the sample or the camera by scanning along the other spatial direction: the relative movement must be synchronous with the acquisition rate of the detector frames to produce a uniform image. A Pushbroom scanner can fetch more light than a whiskbroom, owing to its stay in a precise area for a longer time, providing an extended exposure on the array detector, hence higher spectral resolution.

The staring mode (Figure 8.c), also known as the band sequential method, uses filters instead of a prism in front of a detector matrix to gather a single-band 2D grayscale image with spatial information X and Y at once. After passing through focusing optics, a filter splits light to collect a small narrow band segment of the spectrum at a time. The 3D hypercube originates from defining the filter's wavelength as a function of time. Unlike the whiskbrooms and pushbrooms, the camera collects the scene from a spatial point of view, but one spectral band at a time, and the operator can freely select the number of bands to capture.

Finally, the snapshot mode allows spatial and spectral information recording with a single exposure without scanning. This technique, as shown in Figure 8.d, allows the acquisition of the complete 3D datacube in a single integration time thanks to pixels remapping and the simultaneous scattering of the corresponding light through a prism on a detector. The great advantage is acquiring the entire scene in a single shot in terms of spectral and spatial resolution since the total number of pixels on the CCD detector limits the latter.

Hyperspectral cameras also feature different sensors and detectors which characterise the wavelengths to which they are sensitive.

Literature divides HS cameras into[40,41]:

- **VNIR (Visible Near Infrared)**: wavelengths from 400 nm to 1000 nm
- **NIR (Near Infrared)**: wavelengths from 900 nm to 1700 nm
- **SWIR (Short Wave Infrared)**: wavelengths from 1000 nm to 2500 nm
- **LWIR (Long Wave Infrared)**: wavelengths from 8000 nm to 12000 nm

2.6. Skin cancer detection via hyperspectral imaging

Skin cancer affects the body's largest organ, thus representing one of the most frequent malignancies[42,43]. Physicians usually divide epidermal lesions into melanoma and non-melanoma skin cancer (MSC - NMSC).

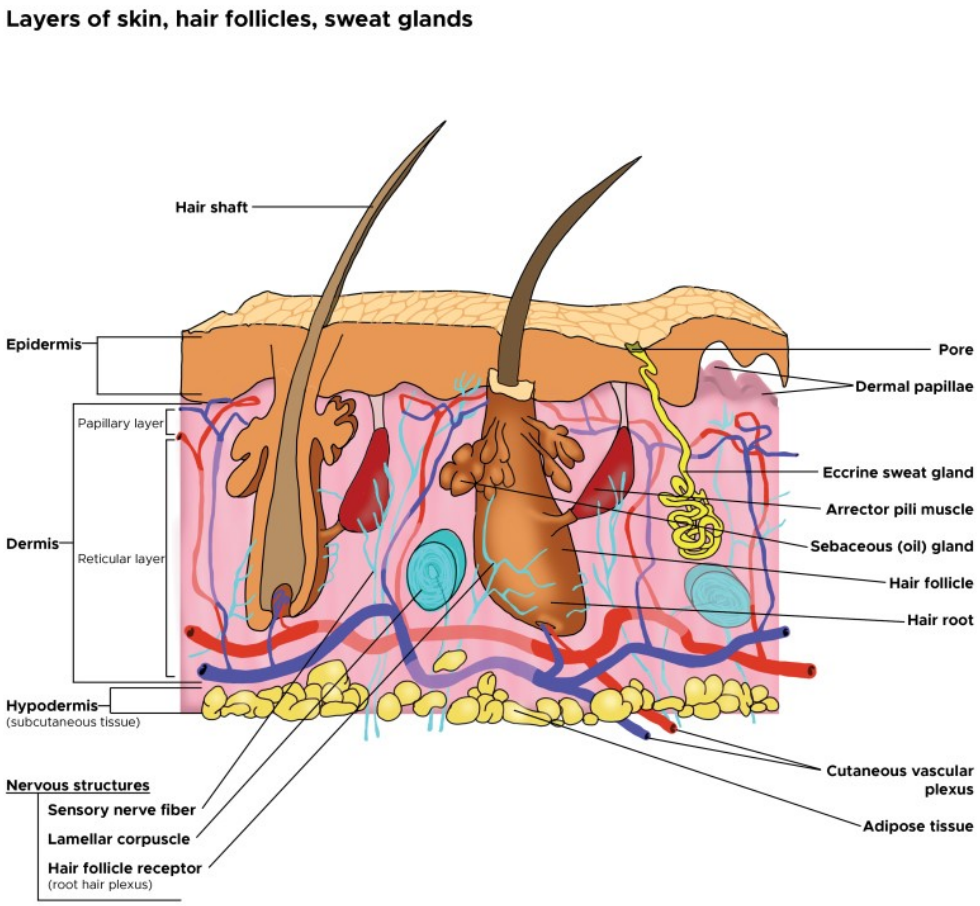


Figure 9. Epidermal layers descriptions and schema[44]

Most skin cancers begin in the epidermis (Figure 9) and can affect three enclosures: squamous cells, basal cells, or melanocytes. The MSC originates from any cell capable of forming melanin and comprises three subtypes: superficial extension, lentigo maligna, and nodular[18,42,44,45]. Some types of skin cancer present genetic modifications that, if left untreated, grow, and spread over the body, yielding potentially metastasising outcomes.

Although MSC is the rarest skin tumour, it causes the highest mortality rates because it lacks adequate early detection. NMSC lesions represent more than 98% of the known skin lesions in the United States of America, of which 75–80% are basal cell carcinoma (BCC), 15–20% are squamous cell carcinoma (SCC), and around 1.6% is MSC, the most lethal type of cancer[18]. Regardless, healthcare professionals must consider BCC and SCC malignant as they might degenerate and induce death[18,19,45,46]. Therefore, sorting epidermal tumours into benign and malignant categories is more accurate. Currently, a person has a 4% chance of developing melanoma, which is responsible for 75% of all skin cancer-related deaths [18,37,42,43].

Dermatologists visually inspect melanocytic tumours to determine the presence of malignancies during routine clinical practice. They operate a handheld instrument incorporating magnifying lenses and constant polarised illumination. The process relies upon the so-called ABCDE rule, where A stands for asymmetry, B for border irregularity, C for colour, D for diameter, and E for evolution[37,45,47]. Nevertheless, this procedure introduces false positives, namely benign lesions classified as malignant. Consequently, the gold standard is a biopsy with surgical lesion excision and histopathological assessment. Nevertheless, this process is painful, time-consuming, slow, and expensive[18,45,47]. The worldwide incidence of skin cancer is rapidly rising, bearing heavy health and economic commitment for diagnosis and treatment. Early skin cancer detection effectively enhances the 5-year survival rate and is correlated with 99% of the overall healing likelihood[42,43]. Hence, the escalating rate of skin cancers and the lack of adequate expertise and innovative methodologies present an immediate demand for systems based on artificial intelligence (AI) and novel optical technologies to assist clinicians in this domain[18].

Researchers investigate hyperspectral imaging for cancer detection in this context thanks to recent specialised advancements.

Chromophores, such as melanin and haemoglobin, are organic molecules that characterise epidermal lesions' spectral properties and vary among skin lesions of diverse etiologies. Consequently, HSI systems should capture such information, enabling AI algorithms to automatically detect and cluster tumours of various categories[18,19,44]. Traditional imaging techniques are limited to the visible light spectrum, leading to limited diagnostic results. However, HS images set the stage for broadband information acquisition, overcoming inter-class similarities and intra-class dissimilarities of various diseases considered in the visual domain[34,37,48]. Researchers strived to develop AI solutions to detect skin cancer early on and strengthen current diagnostic performances, whose efficacy leans heavily on healthcare professionals' expertise[33,34].

Likewise, research should not be limited to the learning methodology but also to conceiving an instrument to overcome existing challenges, such as data availability, interpretability, computational power, operating recent algorithms and real-world clinical scenario applicability. Although present AI algorithms are still in the very early phases of clinical application and are not always ready to aid clinicians, they can be scalable to multiple devices, transforming them into modern medical tools[3,49]. Such novel devices will also store the acquired data, overcoming the data availability issues.

2.7. HS dermatologic acquisition system and database

In this doctoral thesis, we exploited a database of HS images acquired via the custom solution in Figure 10[38]. The system comprises a snapshot camera (Cubert UHD 185, Cubert GmbH, Ulm, Germany) capable of capturing the visual and near-infrared (VNIR) spectrum. The spectral range covered 450 to 950 nm, resulting in a spectral resolution of 8 nm and a spatial resolution of 50 × 50 pixels, whose pixel size was $240 \times 240\ \mu m^2$. The camera has a Cinegon 1.9/10 lens (Schneider Optics Inc., Hauppauge, NY, USA) with a 10.4 nm focal length. The acquisition system employed a Dolan-Jenner halogen source light (Dolan-Jenner, Boxborough, MA, USA) and the lamp employed was a 150 watts quartz-tungsten bulb. A fibre optic ring light guides the HS camera to illuminate the skin surface with cold light, avoiding the high temperature of a halogen lamp on the subject's skin[38]. The authors embedded a dermoscopic lens with a human skin refraction index in a 3D-printed contact structure and attached it to the system. The system allows HS image capturing in 250 ms when controlled by the acquisition software[38].

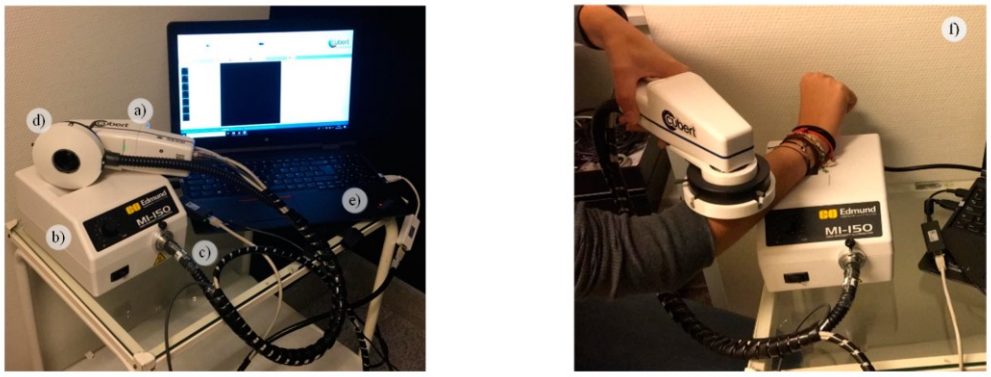


Figure 10. University of Las Palmas' HS dermatologic acquisition system. (**a**) HS snapshot camera; (**b**) QTH (Quartz-Tungsten Halogen) source light; (**c**) Fiber optic ring light guide; (**d**) 3D printed customized dermoscopic contact structure attached to the ring light; (**e**) Acquisition software installed onto a laptop; (**f**) System employed during a data acquisition campaign[38]

The data acquisition campaign occurred from March 2018 to June 2019 at the Hospital Universitario de Gran Canaria Doctor Negrín (Canary Islands, Spain) and the Complejo Hospitalario Universitario Insular-Materno Infantíl (Canary Islands, Spain). The database comprises 76 HS images, 40 benign and 36 malignant skin lesions, from 61 subjects[38]. Pathologists and dermatologists diagnosed suspected malignant lesions

through biopsy-proven histological assessment to evaluate the tumour aetiology, categorising each lesion in the taxonomy described in Figure 11.

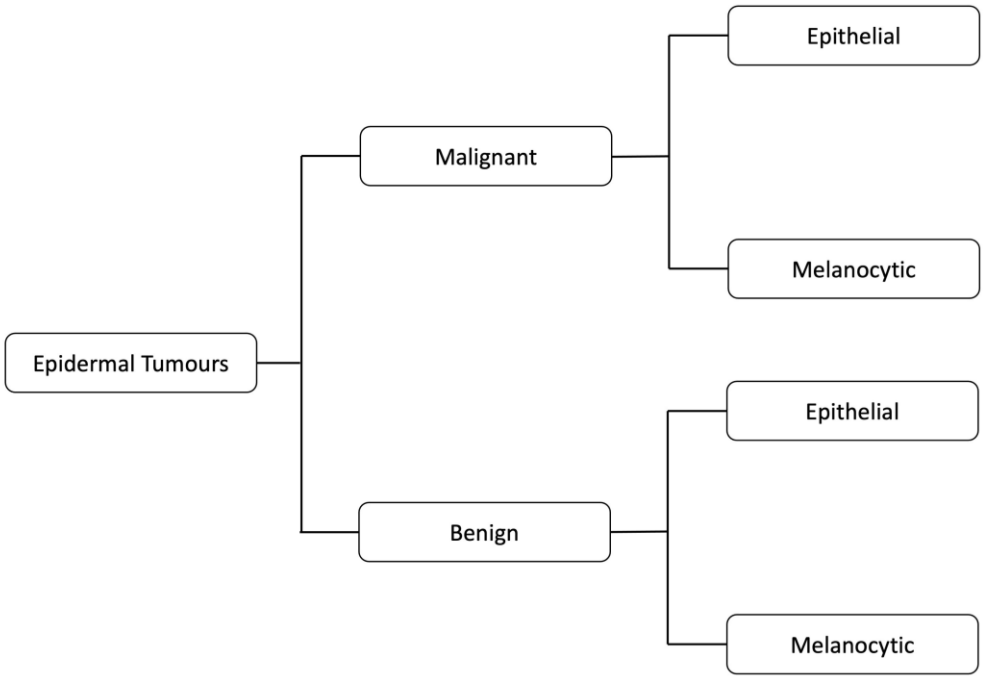


Figure 11. Skin cancer aetiologies

Figure 12, instead, describes the dataset in detail. As depicted in Figure 11, we arranged the dataset in a tree structure with two root nodes representing benign and malignant lesions. This thesis considers only one other level except for the primary root. Remarkably, the root node splits into melanocytic and epidermal tumours. This taxonomy represents a trade-off between other classification approaches, introduced as medically relevant, complete, and well-suited to ML classifiers[18]. Figure 11's taxonomy is well-suited to treat patients according to the highest healthcare standards. The first validation approach uses the primary layer nodes and represents the broadest partition. On the other hand, the children layer represents disease classes sharing similar clinical treatment strategies. Consequently, dermatologists can diagnose more severe lesions earlier and improve patient survival rates. Pathologists and dermatologists diagnosed suspected malignant lesions through biopsy-proven histological assessment to evaluate the tumour aetiology. They assigned each epidermal lesion a category from the taxonomy proposed. They also produced a mask highlighting the tumour borders by visually inspecting the synthetic RGB images generated from the HS cubes[18,19,38].

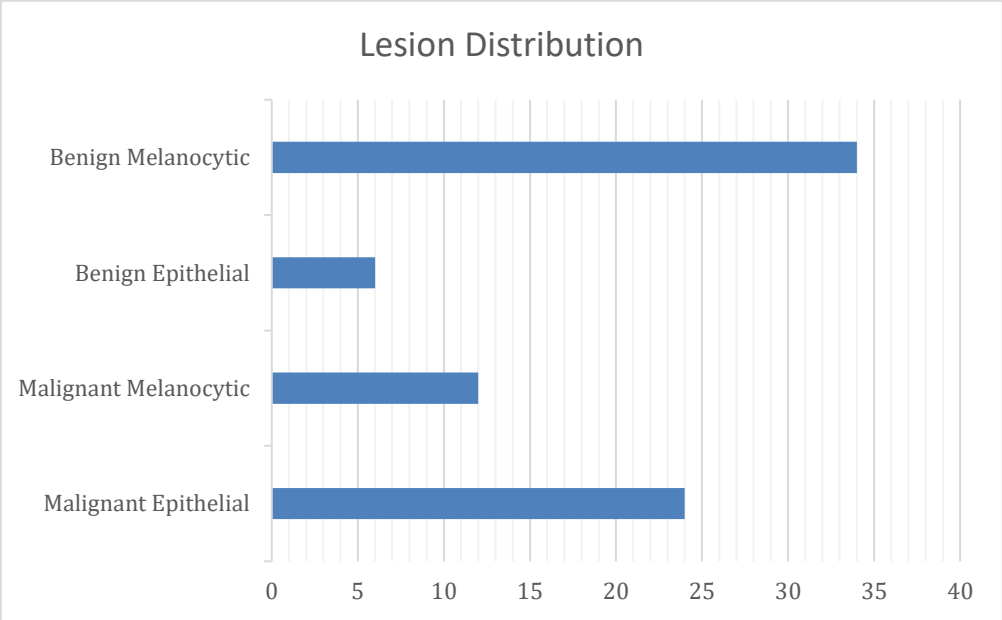


Figure 12. Bar charts of epidermal lesions distributions in the dataset[38]

2.8. Brain cancer contour delineation via hyperspectral imaging

Brain cancer denotes the most common central nervous system malignancy, causing subjects' death and morbidity. It forms directly in the brain or the spinal cord. Healthcare professionals usually cluster nervous system tumours into primary, if cancer arises in the brain, and secondary, or metastasis, if it begins elsewhere in the body, flaring up to the brain[42,50]. Doctors rank brain tumours depending on their nature, source, rate of growth and progression stage. First, they can be either benign or malignant lesions. The harmless cells rarely intrude on neighbouring healthy ones, show distinct borders, and have a slow progression rate. On the other hand, malignant cells readily attack adjacent ones in the brain or spinal cord, presenting fuzzy contours and a rapid progress pace[11,41,50]. The World Health Organization categorises brain tumours into four grades (I, II, III and IV)[11,41,50]. The higher the grade, the faster the rate of growth. Physicians characterise brain tumours according to their progression stages, from 0 to 4. Stage 0 refers to irregular cancerous biological structures that do not distribute to nearby compartments. Stages 1, 2 and 3 denote cancerous cells spreading rapidly. In Stage 4, cancer extends throughout the body[39,50–52]. Glioblastoma (GB - grade IV) is the deadliest brain tumour retaining a 5.5 % 5-year survival rate[11,42,50]. Early and total resection of grade-II increases the overall 5-year survival rate to 81%. Medical practitioners could preserve numerous lives if they detected cancer early via prompt, cost-effective diagnosis procedures. Still, it is challenging to treat cancer at higher stages[11,39,50,52].

Brain cancer diagnosis can be either invasive or non-invasive. The gold-standard biopsy is an invasive strategy. Namely, it is the histopathological examination of a tissue specimen to confirm the malignancy. On the other hand, non-invasive approaches comprise a body and brain scanning. The modalities include computed tomography (CT) and magnetic resonance imaging (MRI) of the brain. These imaging procedures help radiologists uncover brain diseases, monitor disease advancement, and prepare for surgery. Nevertheless, these modalities exhibit inter-reader variability and accuracy owing to the physicians' proficiency[11,12,39].

Meningiomas represent the non-malignant primary tumour, whose resection can discourage further disease progression, improving the survival probabilities. Nonetheless, complete resection is not always feasible and might lead to neurological damage. Consequently, surgeons must balance tumour reduction and neurological conditions[11,12].

At present, neurosurgeons operate several intraoperative guidance tools for cancer resection assistance. Namely, they broadly employ Image Guided Stereotactic (IGS) neuronavigation, Magnetic Resonance Imaging (iMRI), or fluorescent tumour markers like 5-aminolevulinic acid (5-ALA)[11]. However, these medical procedures indicate limitations, namely cost and time, and do not outline precisely lesions borders. Undoubtedly, the procedure course must be reduced as much as possible, being the patient operated on with an open craniotomy. Furthermore, craniotomy and brain shift alter the tumour volume in the intraoperative imaging-guided tools. Therefore, there is a demand to research new imaging modalities that could overcome such limitations[11,41,50].

Hence, hyperspectral imaging plays a significant function in this scenario, and during the last decade, machine and deep learning (ML, DL) solutions emerged as a tool to analyse and cluster different cancer types using HSI[32,34].

2.9. The HELICoiD database of glioblastoma images

Concerning intraoperative glioblastoma segmentation of HS images, research mainly emerged within the European project HELICoiD (HypErspectraL Imaging Cancer Detection)[39]. Researchers gathered an in vivo human-brain HS database during surgical procedures in open craniotomy. The main challenge is retrieving a target ground truth to supervise the ML algorithms. Neurosurgeons can only partially identify the tumour and its boundaries when diagnosing them with traditional imaging systems. Therefore, HELICoiD-based ML studies comprised unsupervised algorithms to overcome this problem and automatically segment the intraoperative-captured HSIs[11,12,39].

The HELICoiD intraoperative HS acquisition system comprised a VNIR pushbroom camera (Hyperspec® VNIR A-Series, Headwall Photonics Inc.,

Fitchburg, MA, USA) to collect data[11,12,39]. It captured HS images ranging from 400 to 1000 nm in 826 spectral bands and a spatial resolution of 1004 pixels. The HS camera acquired via the push-broom method explained in Section 2.5, which provided high spectral and relatively high spatial resolution. Nonetheless, the sensor only captures one spatial dimension of the scene while still capturing its entire spectral signature. Hence, the authors designed a spatial scanning to obtain the complete HS cube with a maximum image size of 1004×1787 pixels ($129 \times 230\,mm$). The system also included an illumination device capable of emitting cold light between 400 and 2200 nm[11,12,39]. The engineers combined a Quartz Tungsten Halogen (QTH) lamp to the cold light emitter via a fibre optic cable to avoid brain surface vulnerability to high temperatures[11,12,39]. The University Hospital Doctor Negrin of Las Palmas de Gran Canaria (Spain) and the University Hospital of Southampton (UK) installed the intraoperative HS acquisition system. Both the Comité Ético de Investigación Clínica- Comité de Ética en la Investigación (CEIC/CEI) of the University Hospital Doctor Negrin and the National Research Ethics Service (NRES) Committee South Central-Oxford C for the University Hospital of Southampton approved the study and its consent procedures signed by all participating patients[11,12,39].

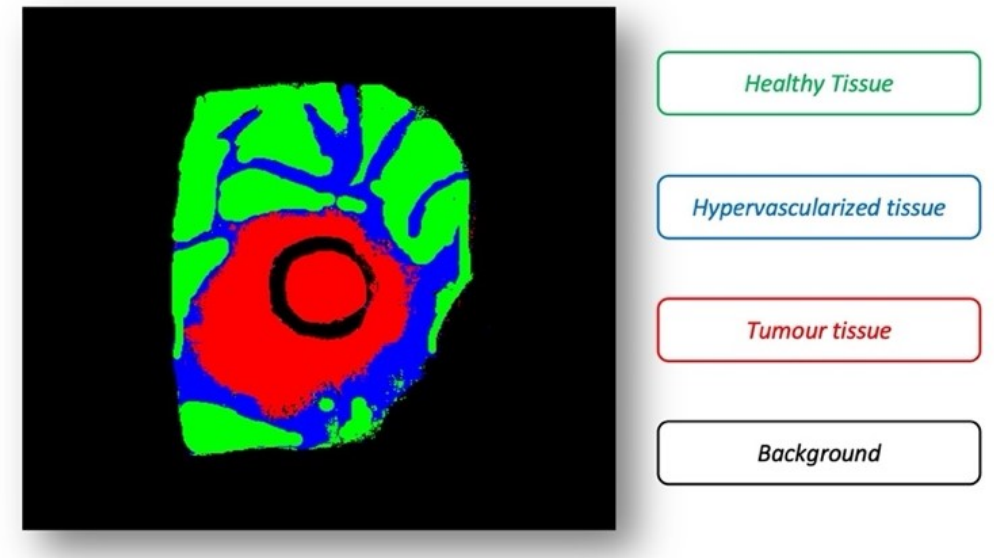


Figure 13. Intraoperative glioblastoma ground truth taxonomy[11,12,39]

Physicians provided each HS pixel with a class concerning Figure 13's taxonomy. The intraoperative GB images present black rubber ring markers employed for the pathological assessment of the image labelling in correspondence with either healthy or tumour tissue. In the latter case, a histopathological biopsy reevaluation confirmed cancer's presence. During the classification, the investigations assigned the markers to the

background class[11,12,39]. Researchers recorded the tumours located in a deeper brain layer after superficial resection.

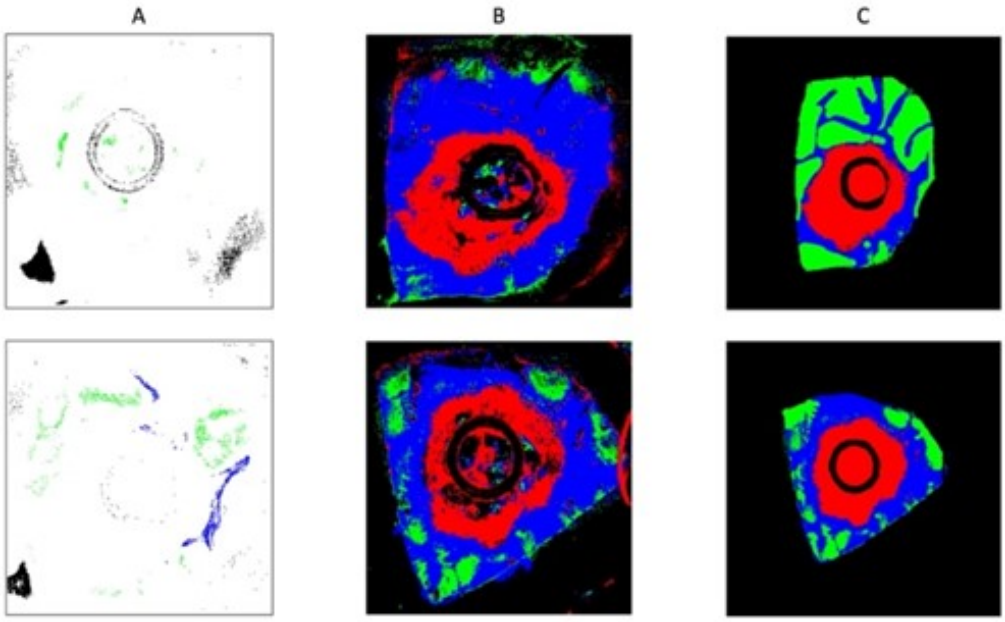


Figure 14. A is the initial ground truth derived from the Neurosurgeon with SAM labelling. B is the HELICoiD ML pipeline result, whilst C is it cleaned version[11,12,39]

Hence, the HELICoiD processing pipeline begins with the Spectral Angle Mapper (SAM)[11,12,39] to segment the entire intraoperative GB hypercube (Figure 14.A). Neurosurgeons operated the SAM to form the initial segmentation dataset. They selected reference pixels from healthy and tumour classes inside the circular markers. Accordingly, pixels having a similar spectrum to the reference pixels received the same categories. Neurosurgeons labelled tumour pixels depending on the histopathological assessment. Likewise, physicians labelled healthy tissue, blood vessels and background by visual inspection according to their experience.

Figure 14.A is the initial ground truth derived from the Neurosurgeon with SAM labelling. Figure 14.B is the HELICoiD ML pipeline result, whilst Figure 14.C represents the cleaned version via filtering and thresholding.

Once the first SAM step ended, the HELICoiD ML pipeline produced the result in Figure 14.B. Since the ML pipeline produced spurious results, we cleaned the segmentation maps via adequate thresholding and filtering, obtaining the smooth mask depicted in Figure 14.C.

The original in-vivo human-brain HS database consists of twenty-six images from sixteen adult patients[11,12,39,52]. Nine patients had histopathologically verified Grade IV glioblastoma, while the remaining seven were affected by other types of tumours or other pathologies requiring a craniotomy. Regardless, this thesis operated on only fifteen

among the original twenty-six images because these offered the required ground truth quality obtained from the HELICoiD ML pipeline.

2.10. Medical data beyond images

Moving beyond image classification, deep learning models can learn from diverse input sources, including time series, tabular, text or even combinations of input types. Healthcare data is intrinsically *multimodal*, and all information produced during a patient's lifespan retains relevant knowledge to provide personalised healthcare[3,5,6]. Data sources such as blood analyses, ECGs, handwritten notes, and histopathological and radiological images inform a physician's therapy judgment. Nevertheless, most medical machine-learning applications focus on a single data source. It is especially true in radiology, whose information complexity, accessibility, and computational interpretability, constitute the core attraction of artificial intelligence applications in medicine[3,5]. In the future, Computer-Assisted Diagnostic (CAD) architectures should process and interpret multimodal information, thereby emulating physicians' reasoning.

2.11. SARS-CoV-2 clinical dataset

This doctoral thesis focused on machine learning applications to counteract the Covid-19 pandemic, discussed in Section 2.2, which elicited an urgent need for reliable diagnostic tools to minimise viral spreading[15] to avoid cross-contamination between subjects and detect their disease positivity to cluster them by prognosis and manage the emergency department's resources. Fondazione IRCCS Policlinico San Matteo Hospital's ED of Pavia let us evaluate the exploitation of machine learning algorithms on a clinical dataset gathered from laboratory-confirmed rRT-PCR test patients, collected from March 1st to June 30th, 2020[17]. Doctors evaluated routine blood tests, clinical history, symptoms, Arterial Blood Gas (ABG) investigation, and lung ultrasound quantitative examination. The personnel collected the ABG samples from the Radiometer ABL 825 (Radiometer Medical ApS, Åkandevej 21, DK-2700, Brønshøj, Denmark). We designed two diagnostic AI-based instruments for Covid-19 detection and oxygen therapy prediction: the need for ventilation support due to lung involvement[15,17].

The main goal was to quickly stratify patients and employ cross-contamination procedures, avoiding extensive swab testing and leveraging physicians' workload. The investigations on the dataset relied on gathering patients' data based on two primary principles. First, we engaged features readily available in every ED triage, such as anamnesis, symptoms, and vital signs. Moreover, physicians collected data concerning patients' respiratory failures, routine blood tests, arterial blood gas (ABG) analysis,

and lung ultrasound quantitative evaluations[16,17]. Fondazione IRCCS Policlinico San Matteo's Emergency Department of Pavia specified a stringent protocol during triage functions to assess patients whom SARS-CoV-2 might have potentially contaminated. The procedure stratified people and avoided cross-contamination during daily clinical operations. This thesis mainly scrutinised clinical characteristics available in any ED triage, such as history taking, symptoms, and vital signs, to apply ML and aid physicians during the pandemic. Furthermore, physicians collected knowledge associated with patients' respiratory malfunctions that satisfied the constraint of being promptly available and cheap, such as routine blood tests, ABG examination, and lung ultrasound quantitative evaluation[15,17,53]. They gathered information from people complaining about probable SARS-CoV-2 symptoms, whom the physicians swabbed for diagnosis.

Eventually, the clinical dataset comprised the list of features shown in Figure 15 for 1355 patients, of which we illustrated the clinical information in Section 2.3's Table 2, where there is the correlation coefficient between each element and the outcome to be predicted, namely both Covid-19 positivity assessment and oxygen therapy potential need.

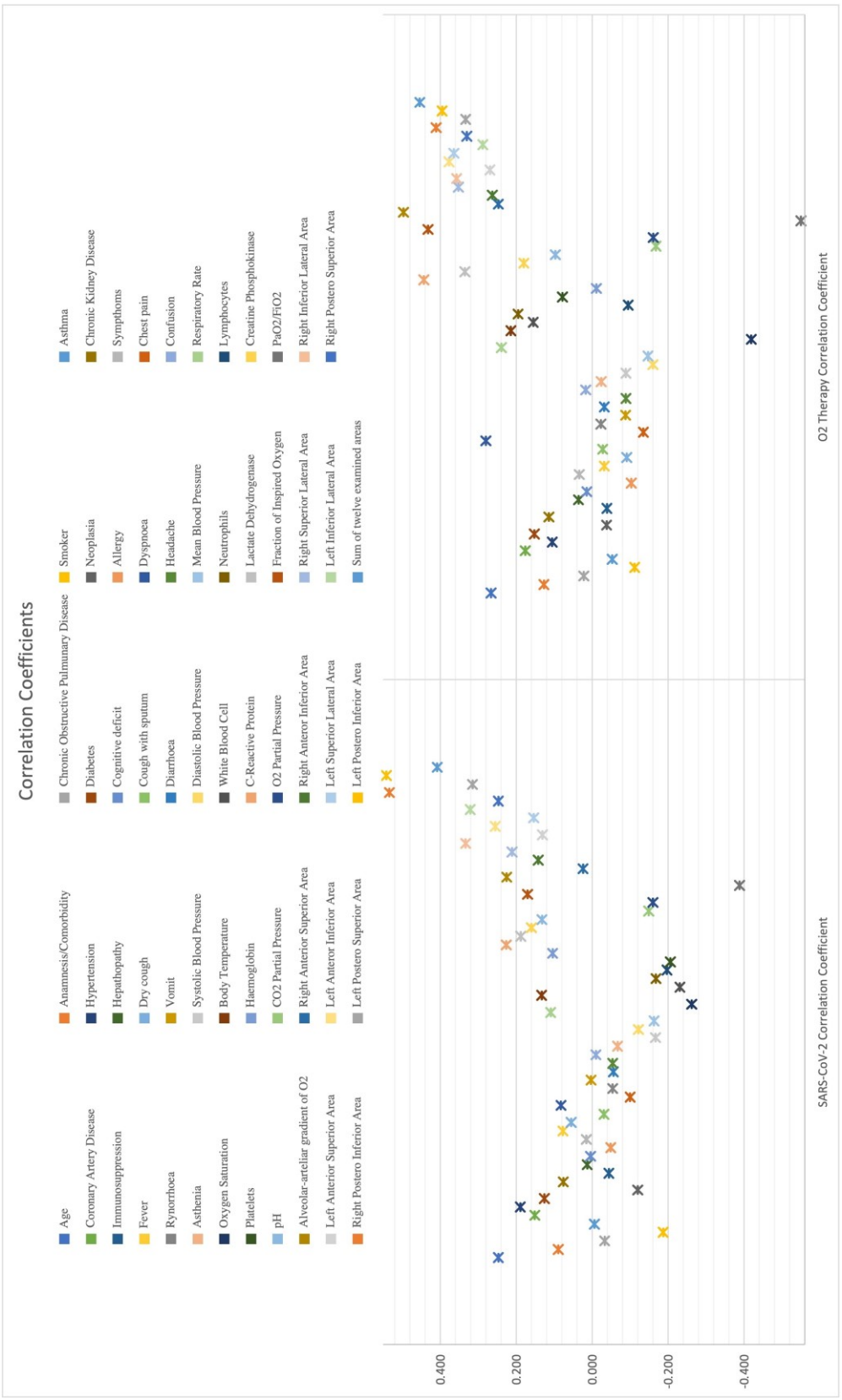


Figure 15. SARS-COV-2 clinical dataset examined features and their correlation with targets[17]

2.12. Challenges and opportunities

Even though AI vows to change medical practice completely, many technological challenges lie ahead[3,6,11]. Artificial intelligence leans heavily on the availability of high-quality large-sized training data. Hence, academic researchers must care about data collection and its representativeness of the target patient population. Different data sourced from various healthcare environments might yield bias and noise, causing generalisation failure on a separate dataset. Likewise, diagnostic tasks with an imperfect inter-expert likelihood exist, and the literature reports consensus diagnoses improving the performance of the machine-learning models[1,3–6]. Hence, good data curation is essential for addressing heterogeneous information. Clinicians must review their notes to allow Natural Language Processing (NLP) procedures, and researchers must focus on artificial intelligence explainability to enable AI in clinical environments. Consequently, it is not yet straightforward to communicate intuitive notions explaining the models' outcomes, to determine model weaknesses, or to extract biological insights from these so-called black boxes[3,5,6].

Implementing a computing domain for collecting, storing, and sharing sensitive health data is necessary. Hence, privacy-preserving methods that can permit secure data sharing are one of the challenges AI in healthcare presents researchers with. Also, smooth data integration across healthcare applications and locations is complex and slow[3,5].

Since most of the medical applications of artificial intelligence concern retrospective data collected for research and proof of concepts, academics should validate the real-world utility of medical AI systems. Furthermore, as clinical AI approaches evolve, there will be an inevitable increase in their clinical use and deployment, which will lead to new social, economic, and legal issues[3,5]. AI will likely enhance healthcare quality, reducing human error and physician fatigue from everyday clinical duties. Careful design of these clinical applications and their implementation is necessary. Regulator entities, such as the FDA or the European Commission, must certify AI systems before large-scale deployment. Accordingly, the FDA should foresee increasing approval submissions: it should evaluate the safety and effectiveness of medical instruments that present a potential and unreasonable risk of illness or injury. Policymakers must set specific criteria for demonstrating the validation process and the quality and representativeness of the validation data[3,5].

Furthermore, research should not be limited to the learning system but also to designing a device to overcome current challenges, such as data availability, interpretability, and computational power, employing recent algorithms, HPC hardware, and having real-world clinical scenario applicability. Indeed, although current AI algorithms are still at the very early stages of clinical application and not always ready to aid clinicians, they can be scalable to multiple devices, transforming them into modern

medical instruments[11,18]. Such novel devices will also store the acquired data, overcoming the data availability issues. Both the hardware-software challenges will lead to economic investments and specialised personnel that will help overcome the challenges mentioned above.

Chapter 3

Fundamentals of Artificial Intelligence

Artificial intelligence is a common buzzword in scientific contexts, born as the consequence of technological advances and experimental results, notably in image analysis and processing. Researchers seized considerable effort and opportunity to deploy AI in medicine, specifically in specialities where images are central, like radiology, pathology and oncology, and the key to safe and efficient use of clinical AI applications relies on informed practitioners[1,3–5]. Despite AI's fast evolution, several central concepts have settled for good. Hereafter, we present AI's building blocks, which are extensively described in well-known books, focusing on medical imaging[1–5]. This chapter aims to define and describe AI's fundamental pillars, state-of-the-art machine and deep learning methods and models, and their application to medical imaging. Specifically, we describe the fundamental AI reasoning operated in this doctoral thesis concerning the miscellaneous data described in the previous chapter, but this doctoral thesis leaves the description of advanced topics to their dedicated chapters. In the end, we discuss the new future research directions.

3.1. Artificial intelligence, machine learning, and deep learning

AI extensively refers to any method or algorithm mimicking human behaviour and cognitive processes. Historically, academics approached AI from two directions: computationalism and connectionism[1]. The former emulates formal reasoning and logic directly, regardless of its biological designs, operating hardcoded axioms and rules combined to deduce new conclusions. Computationalism is comparable to computers, which store and process symbols.

On the other hand, connectionism observes a bottom-up strategy, starting from models of large networks of interconnected biological neurons, from which intelligence emerges as learning from experience.

Expert systems from the 80s are classic examples of computationalism. Nonetheless, their bottleneck concerns the complex process of gathering the required knowledge formulated as production rules. Consequently,

interest in computationalism-based algorithms has faded since the 90s in favour of connectionism-based systems[1–5].

Connectionism and learning-based approaches rely on data concerning performance and information exhaustiveness instead of humans, who might be poorly available in specific world areas or error-prone and biased. Data abundance enhances learning techniques in this scenario, specifically regarding medical images. The scientific community focused on two nested subfamilies of artificial intelligence: *Machine Learning* and *Deep Learning*.

Data drive ML approaches which can learn from it, extracting patterns and structured information without explicit programming. ML operates in two stages: *training* and *inference*. The former allows pattern extraction in previously collected and usually unstructured data. In contrast, inference compares these patterns to new data to make predictions or aid decision-making. Artificial intelligence algorithms, better known as maximum likelihood estimators in the 90s, have continuously matured and improved thanks to high-performance computing enhancements, evolving into more sophisticated hierarchical structures and giving birth to DL. Researchers first used deep learning in the 2000s, referring to a subset of hierarchically structured ML algorithms arranged on multiple levels[1,3,4]. Hence, deep concerns about the high number of levels these structures base on to automatically extract meaningful features from data. Hereafter we will often use the term *feature*, as a unique, measurable characteristic of an event or information. For instance, borders, shapes and colours are features of images[1–5].

Although ML encloses DL, the latter is usually opposed to classical shallow ML, which instead relies on algorithms with flatter architectures and depends on previously engineered features to extract patterns. This antagonism reflects the evolution from ML to DL, from detailed feature engineering to generic feature extraction[1–5]. On the one hand, human experts have always taken part in ML algorithm design, adding domain knowledge and expertise to define relevant features. On the other hand, DL entangles generic, trainable features. Therefore, despite the modelling capability of ML, implementation is limited by the hand-picked features. Alternatively, DL replaces the technical characteristics chosen by human experts with generic, trainable, low-level features involved in the learning procedure, which offer better performance capabilities. Deep learning achieves complex structures, hence better pattern extraction, by stacking layers of shallow features, leading to a hierarchical model network. Academics often refer to DL as an end-to-end method since it involves low-level features and higher-level model training[1–5].

Nowadays, DL models' diagnostic performance has proven to be equivalent to that of healthcare professionals for specific applications, such as skin cancer or breast cancer detection[1–5,14,15,18,45].

3.2. Learning frameworks and strategies

We can split machine learning and deep learning into two complementary categories, namely supervised and unsupervised, which derive from human learning (Table 3).

Table 3. The table displays diverse learning approaches with some of their popular algorithms, as well as a few examples of common applications in medicine. The table is divided in three parts: the basic learning frameworks (supervised, unsupervised and reinforcement learning), the hybrid learning frameworks, and common learning strategies that solve consecutive learning problems or combine several models together[1]

Learning approach	**Typical algorithms**	**Use cases**
	Standard Frameworks	
Supervised Learning	Linear or logistic regression Decision trees and random forests Support vector machines Convolutional neural networks Recurrent neural networks	Cancer diagnosis Organ segmentation Conversion between image modalities
Unsupervised Learning	(Variational) Auto encoders Dimensionality reduction (e.g., Principal component analysis) Clustering (e.g., K-means)	Classification of patient groups Image reconstruction
Reinforcement Learning	Q-learning Markov Decision Processes	Tumor segmentation Image reconstruction Treatment planning
	Hybrid Frameworks	
Semi-Supervised Learning	Generative Adversarial Networks	Tumor classification Organ segmentation Synthetic image generation
Self-Supervised Learning	Pretext task: distortion (e.g., rotation), color- or intensity- based, patch extraction	Image classification or segmentation
	Learning Strategies	
Transfer Learning	Inductive Transductive Unsupervised	Adaptation to different clinical practices Improving model generalization

Ensemble Learning	Bagging - Bootstrap AGGregatING - (e.g., random forests) Boosting (e.g., AdaBoost, gradient boosting)	Estimation of uncertainty Stratification of patients

Supervised learning is the most straightforward method, providing a tight framework with the highest guarantees of success. The term supervision refers to the training stage formalisation: training data consists of labelled input and output pairs, and the model is optimised to yield the desired output (i.e., a diagnosis or numerical estimation) when presented with a specific input. On the other hand, we operate unsupervised learning when we deal with unlabelled data, also known as self-organisation, aiming to discover data patterns (Figure 16)[1–5]. Typical supervised tasks comprise function approximation, like regression and classification. Classification, like pathology presence assessment in radiology, can be binary or address multiple classes, as in determining a particular pathology among several labels. When classification does not concern the whole image but each pixel, we refer to image segmentation[1–5]. Table 3 contains detailed examples and definitions of different learning methodologies.

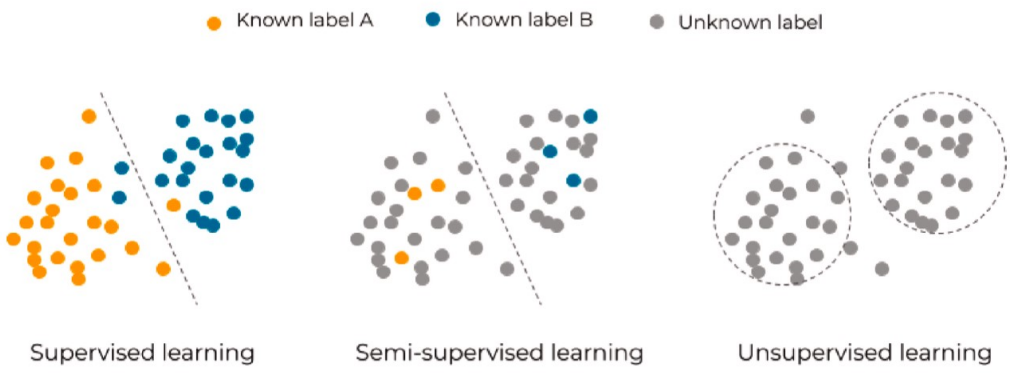


Figure 16. Supervised, semi-supervised, and unsupervised learning[1]

Instead, unsupervised tasks convey probability density estimation, finding separated groups of similar data items, also known as clustering, anomaly detection, and dimensionality reduction, among others. Regardless, unsupervised learning utilisation has been more limited than its supervised counterpart due to the higher complexity involved in the algorithms. Table 3 also offers the main unsupervised ML methods and descriptions.

Apart from supervised and unsupervised approaches, other methodologies enable continuous feature extraction and deep pattern understanding. Accordingly, the third we overlook in this chapter is called

semi-supervised. It is a hybrid framework comprising supervised and unsupervised characteristics involving partially labelled data. The unsupervised part identifies clusters which represent possible class labels (Figure 16). Examples of clinical semi-supervised learning retain the generation or translation of images from a specific class to another (e.g., generation of synthetic CTs from MR images), and segmentation or classification of images[1–5,54–56].

So far, supervision has been the most used framework for medical imaging, as it is straightforward to use. Regardless, medical data labelling is highly time-consuming and subject to inspection by human experts, who at worst might only sometimes agree on the same diagnosis. Consequently, researchers are now transitioning to semi-supervised learning strategies because they represent an excellent alternative to complement small sets of carefully labelled data with large amounts of cheap unlabelled data collected automatically. Indeed, the current limitations of artificial intelligence algorithms come from labelled data. Namely, we might find labelling errors and limited-size databases[1–5,54–56].

The fourth variety of learning is called reinforcement and involves an *agent* interacting with an *environment* where an agent gets feedback from its actions over time and the problem is defined as a Markov Decision Process (MDP)[2,57]. The environment can either reward or punish the agent who has then to best predict the longer-term outcomes of future actions in a trial-and-error manner. Reinforcement learning usage in medicine is not ubiquitous yet but has recently increased, with promising applications concerning physicians' behaviour mimicking for typical tasks such as treatment design (Table 3)[1–5].

On top of these essential frameworks, other strategies enable us to reuse previously trained models (*transfer learning*) or combine models (*ensemble learning*). Transfer learning reuses blocks and layers from a pre-trained model with some data for a specific task and fine-tunes it to pursue different data or tasks. The best common practice is using architectures pre-trained on similar domains to overcome small-sized dataset problems and poor classification performances. For example, a classification model pre-trained on ImageNet[58], which is an extensive collection of natural images, can be partly reused and fine-tuned for medical imaging applications, such as organ segmentation or treatment outcome prediction. Transfer learning allows us to exploit knowledge from different but related domains, mitigating the necessity of an extensive dataset for the target task, and improving the model performance[1–5,59–61]. Accordingly, academics proved that deep architectures always learn similar simple features in their most shallow layers, forming the complex ensembled feature in a deeper one[3,59]. Consequently, pre-trained models offer the same shallow features but come from a different, more comprehensive dataset. For instance, shallow features comprise shape recognition.

Ensemble learning also improves a model's overall performance and stability by combining the output of multiple models or algorithms to perform a task[1,57].

Finally, self-supervised learning is a contemporary hybrid framework embodying the state-of-the-art mostly in Natural Language Processing (NLP) and heavily researched in vision applications, including medical imaging. It could be essential in future research directions for medical computer-aided diagnosis applications. Self-supervision is a variant of the unsupervised framework because it operates with unlabelled data. Regardless, the scheme exploits *free labels* that come for the data, namely, those we can extract from the data structure itself. Generally, self-supervised algorithms function in two stages. First, the model is pre-trained to solve a pretext task that aims to obtain the data's supervisory signals (i.e., the free labels). Secondly, we transfer the acquired knowledge and fine-tune the model to unravel the main task[1–5]. Literature on self-supervision for medical imaging is still inadequate, mainly due to the variety of challenges we already mentioned in this doctoral thesis in Section 2.12. The existence of hybrid-learning frameworks reveals that the borders between supervised and unsupervised learning have progressively blurred to accommodate mixed approaches (Table 3), which can address real-world scenarios and datasets consistently (Figure 17).

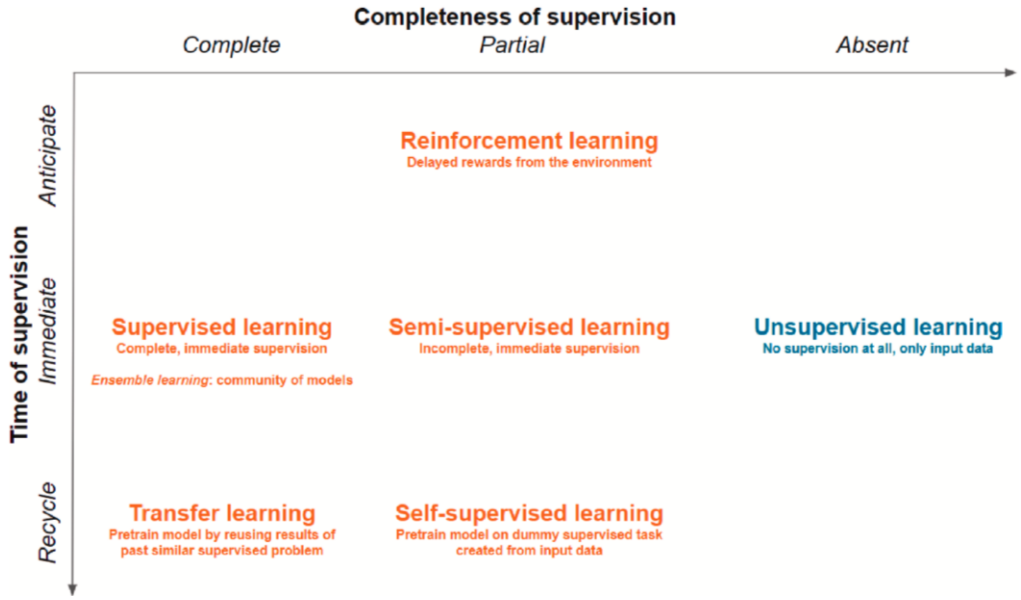


Figure 17. Trade-off between the degree of supervision and the time required for it[1]

3.3. Typical AI-based medical imaging analysis workflow

AI literature reports the presence of standard stages in most workflows for medical imaging processing (Figure 18)[1,3]. Data drives ML and preliminary steps comprise relevant feature extraction and selection.

Consequently, predictive models like classifiers or regressors operate this information to perform a specific task.

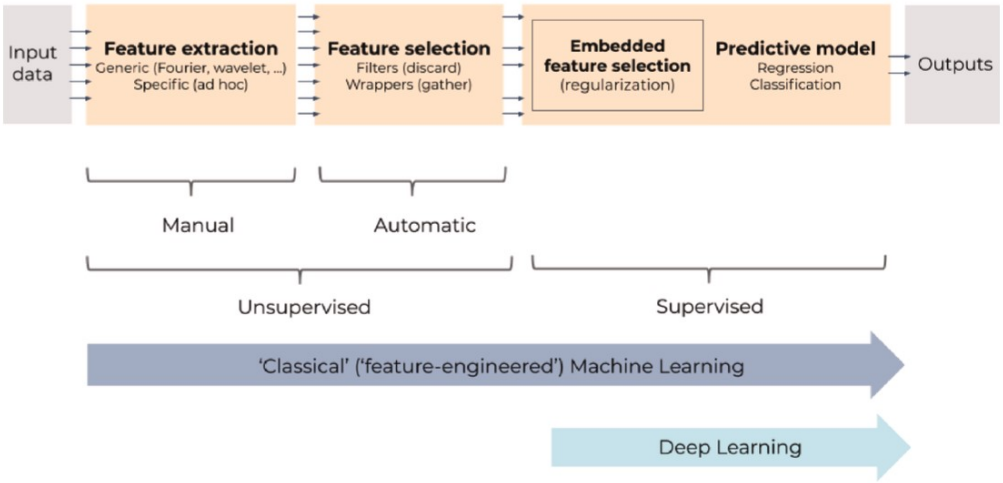


Figure 18. General ML pipeline for supervised learning[1,3,5]

3.4. Feature engineering, extraction, and selection

Before deep end-to-end learning was born, critical steps towards channelling data into AI comprised: feature engineering, extraction, and selection. Feature engineering crafts features by hand. Concerning skin cancer, before academics researched deep architectures to classify images directly, feature engineering comprised characteristics extraction to resemble the ABCD rule: colour, contours, shape, and dimension measurements[45]. Indeed, researchers often classify image features in low-level or high-level features: the former refers to a specific small group of pixels, whilst the latter characterises the full image.

Alternatively, we can extract higher-level features in a more data-driven form by operating dimensionality reduction. Methods like Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA)[57] can reduce the number of input variables (i.e., features) according to specific criterions. While dimensionality reduction might improve the predictor's performance, they strongly reduce data interpretation and explainability due to the translation to other geometrical spaces. In this chapter, we will overlook similarities in computer vision algorithms. Specifically, the convolutional filters involved in convolutional networks bear similarity with the feature engineering above: they extract local characteristics from data later stacked together to allow global higher-level features to emerge[1–5].

It might happen to manage redundant or irrelevant features, and feature selection addresses this issue: we can discard some of those to focus on a reduced set of components. Usually, the optimisation process involves

terms in the cost function that handles this issue directly. This approach is called *regularisation* and involves each feature having weights associated with it which, if set to 0 or 1, perform the selection process. Indeed, features' weights regularisation (i.e., L1 or L2 norm regularisation) can prefer sparse configurations, where irrelevant features get null weights[1–3,57].

3.5. Predictive models

We can generally divide AI everyday tasks into *regression* or *classification*. The former retains models estimating continuous values, like a dosage, whilst the latter predicts class probabilities, such as pathology assessment into benign and malignant aetiologies[57]. This section describes the models' main methodological aspects and the state-of-the-art examples.

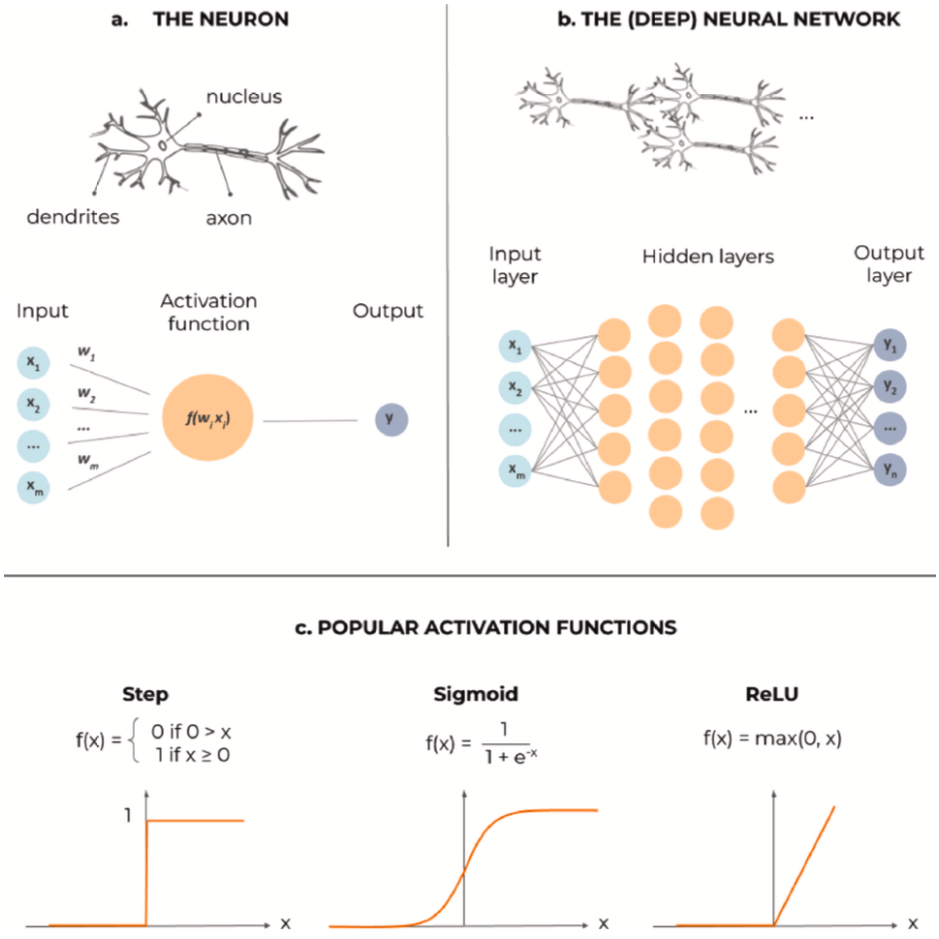


Figure 19. Brief recap on artificial neural networks: (a) The formal neuron, processing several dendritic inputs through a nonlinear activation to produce its actional output; (b) The neurons can form a network in a feed-forward fashion, and specific activation functions can deliver the output to achieve regression or classification. (c) Examples of nonlinear activation functions[1]

Regression represents the most generic task in supervised learning, and comprises well-known statistical approaches like linear regression, but other mathematical approaches exist involving exponential or polynomial functions. ML generalises this concept to universal approximators that can fit data sampled from almost any smooth function and possibly many input and output variables. Artificial neural networks (NNs) are universal approximators (Figure 19)[1]. They consist of interconnected formal mathematical models of neurons, a cell combining several dendritic inputs into a weighted sum that triggers an axonal output through a non-linear activation function. Examples of activation functions comprise step, sigmoid, or special functions like Rectified Linear Units (ReLUs). The universal approximation theorem states that when a NN possesses at least a hidden layer of neurons with non-linear activation functions, the model can fit any input-to-output mapping[2]. Regardless, the more neurons the hidden layer counts, the more complex functions we can resemble. This capacity is roughly proportional to the number of synaptic weights in the NN and resembles the polynomial order of a regression. As we mentioned, the term *deep* of the learning process refers to the high number of hidden layers present in a NN, which makes it a universal approximator. Interest in deep NN lies in trading the width of a single hidden layer for depth. As we stack hidden layers, we enable hierarchical processing and higher generic and complex feature ensembling starting from the shallow ones of earlier layers[1,2].

Most NNs are feed-forward, meaning data flows unidirectionally from inputs to outputs. Recurrent NNs (RNNs) add feedback loops, namely memories, allow sequential data processing (i.e., text, videos)[1,2].

Independently from the learning framework, the AI model's training consists of minimising a loss function between the target output and the one the NN predicts in its current parameter configuration. The minimisation process produces partial derivatives, namely the gradient, of the loss function concerning these parameters, indicating the direction in which tuning the parameters is likely to decrease the loss. In a feed-forward NN, this gradient information flows back from layer to layer towards the input, yielding the backpropagation algorithm[1,2].

Typical loss functions depend on the problem to be solved (i.e., regression or classification). They comprise the mean square error (MSE), the Cross-Entropy and general logit-based functions. Also, different optimisation techniques exist to perform backpropagation concerning the memorisation of past directions yielded by the gradients. The learning rate is a very important hyperparameter because it determines the learning speed of the network. Indeed, the learning rate scales the gradient before the weights update. Besides the gradient descent algorithm, other weights update algorithms were born over the years: RMSprop, Adam, AdaDelta, AdaMax, Adagrad, and Nadam. Each of them has advantages and disadvantages[1,2].

3.6. State-of-the-art AI methods

In the last decade, research attention has moved from ML methods such as Support Vector Machines (SVMs) and Random Forests (RFs) to vision architectures such as Convolutional Neural Networks (CNNs) (Figure 2, right). Furthermore, since 2018 the exploitation of other more complex DL methods, including Generative Adversarial Networks (GANs), is rapidly advancing[1,2].

The following sections briefly review all the fundamental models and strategies employed in this doctoral thesis, whilst we leave mentioning more advanced topics to their dedicated chapters that will appear later in the manuscript.

3.7. Random forests (RFs)

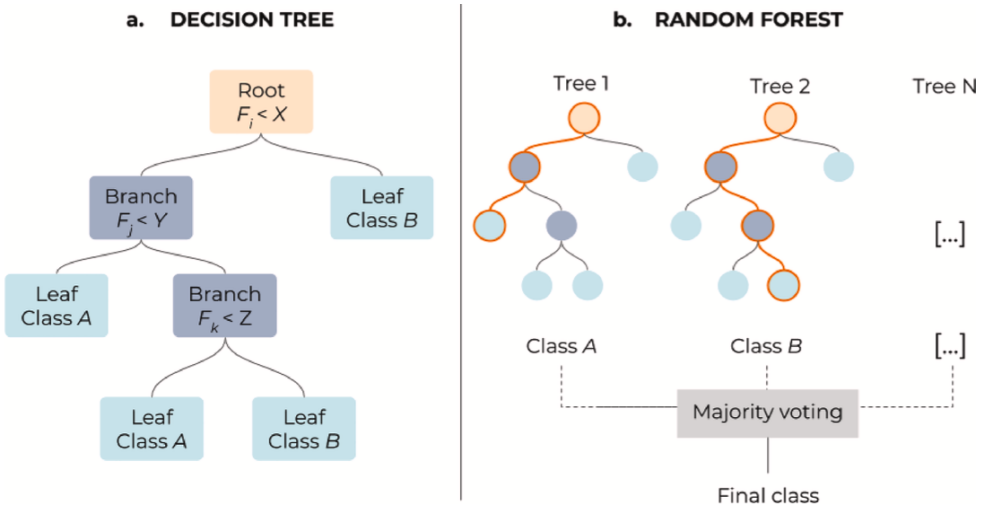


Figure 20. Decision Tree and Random Forest[1]

Random Forest is an ensemble learning methodology that achieves classification by designing a cluster of decision trees throughout the training (Figure 20)[1,2,57]. Each decision tree defines a base model, namely a binary classifier, with its respective decision. The assortment of such decisions leads to the final output. RFs meet ensembling using internal feature selection and voting. The RFs algorithm extracts many low-level feature representations and uses the selection mechanism earlier described to find the most informative ones. After extraction, a majority vote on selected classifiers yields the final decision[1,2].

The hyperparameters describing an RF algorithm are the number of estimators composing the forest, the tree's maximum depth, the highest number of levels we let each tree reach, and the estimator's minimum number of data points placed in a node before splitting it. Likewise,

academics usually tune the maximum number of features to be considered for splitting a node and the minimum number of data points allowed in a leaf. Eventually, we also choose whether to bootstrap our data, namely resampling it. Data scientists usually recommend exploiting bootstrap when the dataset size is small.

RFs are easy to implement and less computationally expensive than CNNs. Accordingly, they can work on regular CPUs. Consequently, they still play an essential role in the ML toolbox for medical applications[1,2].

3.8. Support Vector Machines (SVMs)

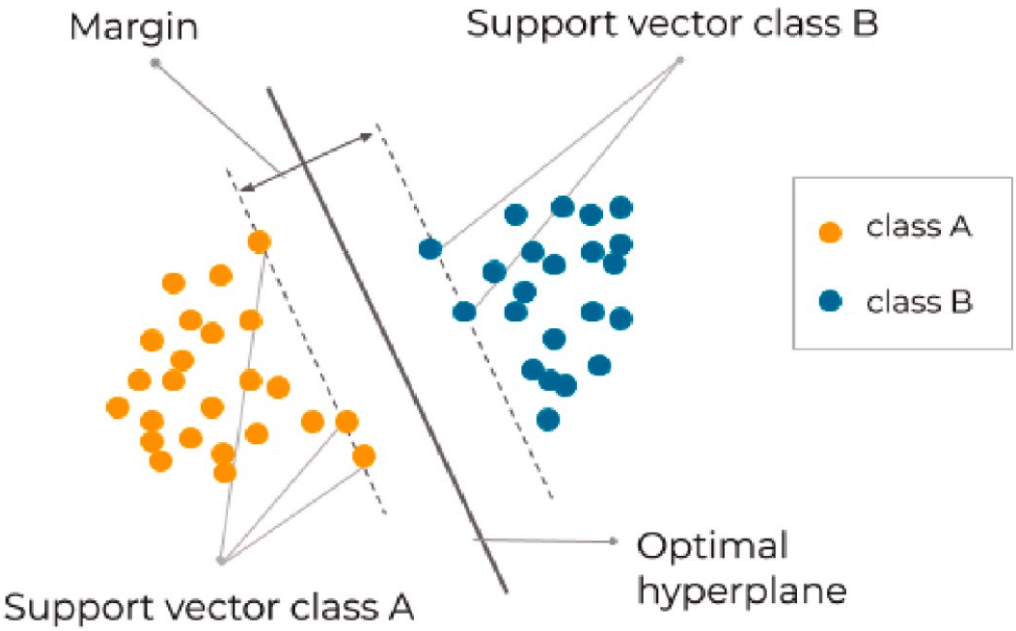


Figure 21. Principles of the linear Support Vector Machines[1]

SVM also conveys a supervised learning methodology, denoting one of the most robust prediction algorithms (Figure 21). An SVM acts by projecting the features in a p-dimensional space and dividing them such that exists a hyperplane clustering the points. The gap between the hyperplane and the data points belonging to each class must be the widest[1,2,57]. There exist fewer hyperparameters compared to the RF algorithm. First, the kernel function is the non-linear function that maps the data into the p-dimensional space, enabling us to fit the maximum-margin hyperplane. Data points could not be linearly separable. Accordingly, we alter them through a kernel function acting as a degree of closeness, mapping the data into another feature space. One of the most used functions is the Radial Basis Function (RBF), featuring the hyperparameter γ. The hyperparameter γ controls the distance of influence of each training

point. The lower its value, the higher the similarity radius, resulting in more points grouped. Likewise, it exists also for different kernels[1,2,57].

Moreover, we choose the C hyperparameter, a penalty we assign for each misclassified data point. The minor C, the lower the error penalty. Typically, we look for $\gamma \subset$ [1e-4,10] and C $\subset$ [1e-1, 100].

3.9. Convolutional Neural Networks (CNNs)

We already mentioned neural networks as computational models with a layered structure composed of interconnected nodes in Section 3.5.

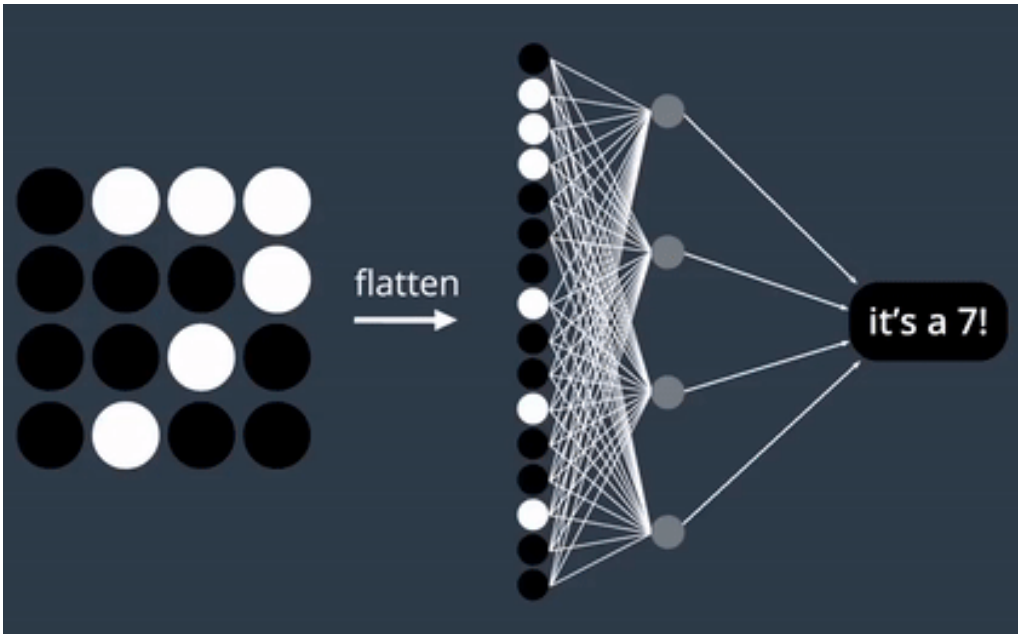


Figure 22. Pixel unrolling (content provided by Udacity Inc. Creative Common License)

Before convolutional neural networks were born, academics processed images by performing *pixel unrolling* (Figure 22). Instead, CNNs take inspiration from the human visual system and exploit the spatial arrangement of data within images. Namely, we no longer have Figure 22's redundant connections, but we arrange neurons in kernels (Figure 23). Their unique ability to glimpse hierarchical data representations has made CNNs the most popular architecture for existing medical image processing applications.

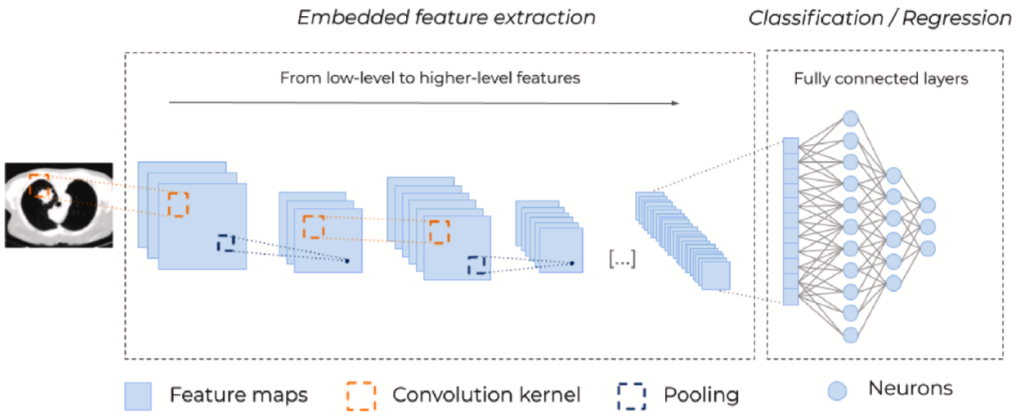


Figure 23. Typical architecture of a Convolutional Neural Network (CNN)[1]

CNNs are feed-forward architectures whose information propagates in only one direction: it starts from the input layer, flows along the hidden layers, and comes out of the output layer, providing the final estimate. The architecture of a CNN can vary considerably depending on the task it is required to perform, but in general, it receives an input image to be classified, and the central part comprises convolution and pooling layers organised in blocks[1,2,57]. These, stacked on top of each other, contribute to the network's depth expansion. Fully-connected layers occur at the stack's end, and the last has as many neurons as the number of classes to be identified. The network returns a label at the output, referring to the class to which the image under analysis belongs (Figure 23)[1,2,57].

CNNs vary according to layer types and their arrangement into a topology. Hereafter, this chapter describes the most common layers building up CNNs:

- Convolutional layer
- Batch normalisation layer
- Pooling layer
- Transposed convolution layer
- Dropout layer
- Fully-connected layer

The convolutional layer is the fundamental building block of a CNN, and scientists named it after the mathematical operation. It extracts input images' fundamental characteristics and can appear immediately after the input layer or after the previous convolutional layer's output. The first convolutional layers identify specific features, such as image edges and corners, while deeper ones can identify more complex features up to accurate object recognition[1,2,57].

The convolutional neurons organise in feature maps (e.g., matrices), each having a receptive field related by weights to neighbouring neurons

positioned in the previous layers. The weights linking different convolutional layers' neurons constitute a kernel or filter matrix.

This layer performs the convolution operation shown in Equation 1, returning a new feature map:

$$\boldsymbol{Z} = < \boldsymbol{w}, \boldsymbol{x} > + \boldsymbol{b} \qquad \text{Equation 1}$$

Z is the output of the convolution, w is the weights matrix, x is the input matrix, and b is the bias.

The convolution operation consists of the scalar product (i.e., point-by-point multiplication and overall sum) between the input matrix and the kernel. The stride drives the filter scrolling over the input matrix. In some cases, we might add a frame of zeros to the input matrix (padding) or expand the kernel size by introducing a parameter called dilatation. Therefore, convolution down-samples the input image, producing an output image smaller than the original one (Figure 23). Equation 2 displays the convolution output size.

$$\boldsymbol{OutputSize} = (\boldsymbol{InputSize} - \mathbf{1}) \cdot \boldsymbol{Stride} + \boldsymbol{FilterSize} - \mathbf{2} \cdot \boldsymbol{Padding} \qquad \text{Equation 2}$$

The convolution output is then passed to a non-linear activation function, as shown in Figure 19, to extract the data's non-linear characteristics. Traditionally, the activation functions from Figure 19 are the most used due to the increased performance in the learning process after their usage. Additionally, researchers also employ the Tanh, and Leaky ReLU[1,2,57].

Besides convolution, batch normalisation makes the network faster and more stable by normalising the data it receives from the kernels. Equation 3 defines the normalisation:

$$\widehat{\boldsymbol{x}}_{\boldsymbol{i}} = \frac{\boldsymbol{x}_{\boldsymbol{i}} - \boldsymbol{\mu}_{\boldsymbol{B}}}{\sqrt{\boldsymbol{\sigma}_{\boldsymbol{B}}^{\mathbf{2}} + \boldsymbol{\epsilon}}} \qquad \text{Equation 3}$$

x_i is the input mini-batch, μ_B and σ_B are the mean and variance vectors over the entire mini-batch, respectively, and ε is a factor that improves numerical stability.

A *mini-batch* is a limited sample of input data. There may need to be more to CNNs than normalisation to make the data optimal for further processing. For this reason, Equation 4 introduces a further variation in the data where the parameters γ and β are defined as scale and offset, respectively [1,2,57].

$$\boldsymbol{y}_{\boldsymbol{i}} = \boldsymbol{\gamma}\widehat{\boldsymbol{x}}_{\boldsymbol{i}} - \boldsymbol{\beta} \qquad \text{Equation 4}$$

Afterwards, academics usually introduce pooling layers between successive convolutional layers. It aims to reduce feature maps' spatial resolution and achieve spatial invariance to input scale changes and offsets.

Pooling partitions the input image or features coming from deeper layers into non-overlapping sub-regions, and for each region, it computes an output value according to specific rules. The two most used pooling techniques are max pooling, which calculates the maximum value of each sub-region, and average pooling, which calculates the average value of each sub-region (Figure 24)[1,2,57].

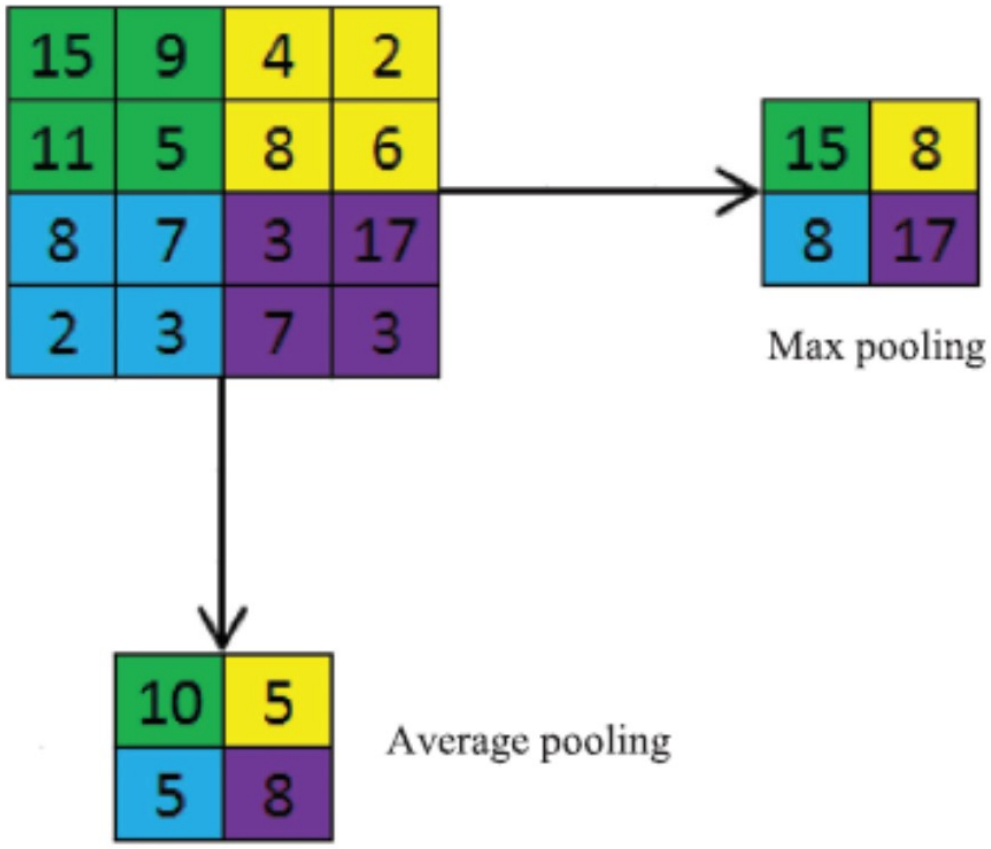


Figure 24. Average and Max Pooling operations[62]

The output of a pooling layer has dimensions calculated by referring to Equation 5.

$$\boldsymbol{OutputSize} = \mathbf{1} + \frac{\boldsymbol{InputSize} + \mathbf{2} \cdot \boldsymbol{Padding} - \boldsymbol{WindowDim}}{\boldsymbol{PoolingStride}} \qquad \text{Equation 5}$$

On the other hand, transposed convolution up-samples the input's spatial resolution. It operates in the same way as a convolution layer but by making changes to the input feature map. It is necessary to follow the steps listed in Equation 6 to build a transposed convolution layer: given an input image, a kernel, a padding value (p) and a stride value (s), we can compute the new parameters z and p', then the input image is modified by spacing each row and column with many zeros equal to the z dimension. This way,

the input image size increases, and we can compute the standard convolution between the modified image and the kernel[1,2,57]. The result of the transposed convolution is given by Equation 6:

$$\boldsymbol{OutputSize} = (\boldsymbol{InputSize} - \mathbf{1}) \cdot \boldsymbol{Stride} + \boldsymbol{FilterSize} - \mathbf{2} \cdot \boldsymbol{Padding} \qquad \text{Equation 6}$$

The dropout layer makes the network more robust in the classification task and is only active during the training phase. It randomly selects precise activations belonging to a specific network layer and temporarily removes them from the model, setting them equal to zero (Figure 25)[63].

Each node remains in the network with a specific and arbitrarily chosen probability. By doing so, one trains the network whose model varies slightly between iterations of the training process and obtains a generalisable result[1,2,57,63].

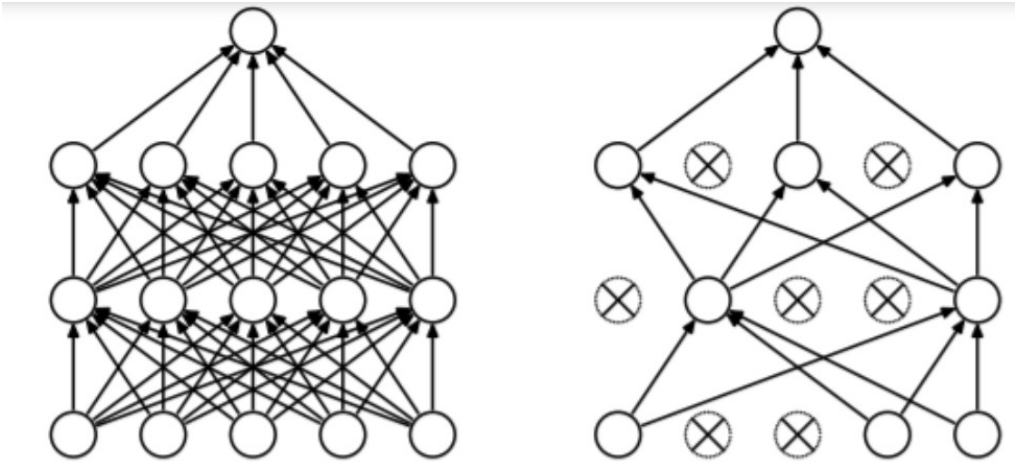


Figure 25. Dropout example, before and after[63]

Eventually, a fully-connected layer consists of neurons connected to all activations of the previous layer. When placed at the network's dropout, it has as many neurons as the number of existing classes. For classification, it operates the softmax function given in Equation 7:

$$\boldsymbol{Softmax}(\boldsymbol{z_i}) = \frac{\boldsymbol{e^{z_i}}}{\sum_{\boldsymbol{j=1}}^{\boldsymbol{K}} \boldsymbol{e^{z_j}}} \qquad \text{Equation 7}$$

K equals the number of classes, and the *Softmax* operator transforms the output of the fully connected layer into a vector of elements with values between 0 and 1, representing the probability referring to each class. In the case of binary classification, the fully connected layer instead operates the sigmoid function[1,2,57].

In conclusion, CNNs stack layers of convolutions and down-sampling and fully connected layers towards the output (Figure 23). Sequential

applications of multiple convolutions enable the network to pull first shallow features, like edges, in the earlier layers, which are next combined and refined into richer, more elaborated, hierarchical features, like whole organs[1,2,57].

We mentioned how each convolutional layer has a feature saliency determined by scanning a fixed-size convolution kernel (typically 3x3) over the input to yield a map. This procedure authorises parameters economy called weight sharing and more accessible training. Downsampling layers are inserted between convolutional layers to reduce the size of feature maps by applying pooling operations. Consecutive pooling allows for shift invariance concerning image content, as the salient maximum or average from the pooling might originate from anywhere in the block. Downsampling trades resolution for number, as more convolution filters operate smaller maps within the identical memory footprint. Eventually, fully connected layers generate the outputs, where all neurons are interconnected[1,2,57].

The following sections analyse the specific CNN architectures employed in this thesis.

3.10. CNNs topologies

Due to their ability to represent abstractly complex concepts such as images and words, CNNs affect many fields and have experienced a natural evolution. Indeed, since 2012 academics have researched many variants of these networks, and we can group them into four different categories, each including multiple models with more assorted characteristics[1,2,57].

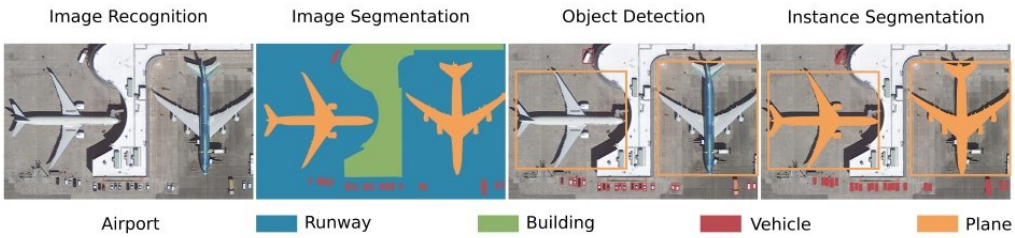


Figure 26. Examples of CNNs' tasks[64]

CNNs grouping relies on the tasks each network can perform (Figure 26). Accordingly, we have:

- Networks for image recognition predict a label that is representative of the class to which the image belongs
- Networks for image segmentation, also called semantic segmentation networks, deal with the assignment of a label for

the classification of each image pixel, resulting in an image of the same size as the source image, known as a mask
- Object detection networks assign a label to the objects in the image and identify their position through bounding boxes that delimit the object.
- Instance segmentation networks, they simultaneously perform the detection and segmentation of the objects in an image

Each of the groups aforementioned contains real families of network architectures whose evolution over time is documented[64]. The description of all the possible types of Convolutional Neural Networks that exist is beyond the scope of this doctoral thesis work, which focuses on specific image recognition and segmentation models, which concern the following paragraphs.

3.11. Training phase

To meet the desired network output, CNNs use learning algorithms to adjust network weights and parameters. The Backpropagation algorithm is the most used for this purpose. It relies on the Gradient Descent Method to find the minimum error made by the loss function with respect to various weights and parameters. Likewise, the most popular loss function is the Cross-Entropy (CE).

The training begins with the pseudo-random initialisation of network's weights and biases. Consequently, during the *feed-forward* phase, each node in the network calculates its own activation function and its respective derivative, the former is propagated forward, whilst the latter stays in memory. When the data flow reaches the end of the network, the loss function (i.e., the cross-entropy) is calculated. At this point, starting from the error committed and measured by the CE, we retrace the network in the opposite direction and calculate the gradient of the loss function with respect to the weights and biases, at each level l, using the chain rule, as shown in Equation 8 and Equation 9 where i and j are the dimensions of the matrix of weights and bias vector belonging to the layer l.

$$\frac{\partial L}{\partial w_{ij}^{[l]}} = \frac{\partial L}{\partial z_i^{[l]}} \times \frac{\partial z_i^{[l]}}{\partial w_{ij}^{[l]}} \quad \text{Equation 8}$$

$$\frac{\partial L}{\partial b_i^{[l]}} = \frac{\partial L}{\partial z_i^{[l]}} \times \frac{\partial z_i^{[l]}}{\partial b_i^{[l]}} \quad \text{Equation 9}$$

Equation 10 and Equation 11 can then be used to update the weights where l is the reference level and α is a hyperparameter called learning rate, appropriately chosen.

$$w^{[l]} = w^{[l]} - \alpha \frac{\partial L}{\partial \mathbf{w}^{[l]}} \quad \text{Equation 10}$$

$$b^{[l]} = b^{[l]} - \alpha \frac{\partial L}{\partial b^{[l]}}$$ Equation 11

The cycle repeats until a certain number of iterations are reached or when the loss function stabilises.

Finally, the batch size is a number of samples processed before the model is updated. The number of epochs is the number of complete passes through the training dataset. The size of a batch must be more than or equal to one and less than or equal to the number of samples in the training dataset.

3.12. Data augmentation

Data augmentation in data analysis is a technique that enlarges the amount of available information by adding slightly modified copies of existing data or synthetic data created from scratch using methodologies similar to the ones we will encounter in Section 3.19 of this chapter. It acts as a regulator and helps reduce overfitting during the training of a machine learning model. It is closely related to oversampling in data analysis[15].

The augmentation process helps neural networks focus on meaningful information, hence providing disturbance rejection to the adversarial attacks we mentioned in the introduction of this doctoral thesis. Standard augmentation procedure comprises geometric, filtering, random centre cropping, and colour transformations to the training instances. This method produces effective results in DL classification tasks, significantly reducing overfitting[65]. Furthermore, researchers usually add salt-and-pepper white noise to enlarge the training set. Accordingly, handling RGB enables colour augmentations.

Data augmentation numerically modifies the training images, introducing statistically diverse samples, and allowing the architectures to robustly classify new data: moving the point of interest in the image and slightly modifying its shape or colour together with noise, prepares the architectures not to perceive relevant features always in the same place. Consequently, the models learn to reject disturbances such as probe sensor movements or measurement errors[15].

Table 4 contains a list of augmentations. The recursive application of each Table 4's entry broadens the training set exponentially.

Table 4. Data augmentation operations used during the investigations. We list both the augmentations names and descriptions[15]

Augmentation Name	Augmentation Description
Image noise	Adds salt-and-pepper noise to image. Namely, random pixels get randomly coloured towards white. Spreading power of modified pixels can be set by a parameter; hence, different augmentations can be considered as being more or less noisy.
Colour jittering	Adjusts the colour of RGB image I with a randomly selected value of hue, saturation, brightness, and contrast from the HSV colour space. Specify the range of each type of adjustment using name-value pair arguments. Four augmentations can be retrieved.
Flip	Images are flipped either from left to right or upside down.
Centre cropping	Images are centre cropped using a 150 × 150 window to ensure that image patterns are selected during operation.

3.13. Train-test split, k-fold cross-validation and evaluation metrics

Cross-validation is a resampling procedure to evaluate AI models having limited data samples. It is a statistical approach whose outcomes in metrics estimations offer lower bias than other procedures. Researchers usually adopt performance evaluation in unique measurements such as ROC-AUC, Accuracy, Precision, Recall and F1 Score (Equation 12 to Equation 18).

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP} + \boldsymbol{TN}}{\boldsymbol{P} + \boldsymbol{N}} \quad \text{Equation 12}$$

$$\boldsymbol{Sensitivity} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}} \quad \text{Equation 13}$$

$$\boldsymbol{Specificity} = \frac{\boldsymbol{TN}}{\boldsymbol{TN} + \boldsymbol{FP}} \quad \text{Equation 14}$$

$$\boldsymbol{Precision} = \boldsymbol{PPV} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FP}} \quad \text{Equation 15}$$

$$\boldsymbol{F1\ Score} = \frac{\boldsymbol{2 \cdot Precision \cdot Recall}}{\boldsymbol{Precision + Recall}} \quad \text{Equation 16}$$

$$\boldsymbol{IOU} = \frac{|\boldsymbol{A} \cap \boldsymbol{B}|}{|\boldsymbol{A} \cup \boldsymbol{B}|} \quad \text{Equation 17}$$

$$\boldsymbol{DICE} = \frac{\boldsymbol{2} \cdot |\boldsymbol{A} \cap \boldsymbol{B}|}{|\boldsymbol{A} \cup \boldsymbol{B}|} \quad \text{Equation 18}$$

Cross-validation has a single parameter called k, which refers to the number of groups in which we split the dataset. When k is as big as the data

sample size, the procedure is called leave-one-out cross-validation. As such, the method is named k-fold cross-validation[1,2,57].

Cross-validation is primarily used in applied ML to estimate the performance of a model on unseen data not used during the training stage.

This doctoral thesis operated K-fold cross-validation in two ways. Indeed, each model was trained k times, recording its estimate for each test set. Then, we could either evaluate the metrics mentioned above (Equation 12 to Equation 18) on the aggregated group of predictions, namely the union of each k-fold test set or on each separate group detecting statistical variation in each measurement (mean, variance, or percentiles).

Besides cross-validation, data scientists usually perform one-time standard train-test-validation splits with data split percentages varying depending on the dataset size[1,2,57].

3.14. Models hyperparameters

Every model we encounter in this Chapter exploits hyperparameters to classify the data. Hence, researchers usually adopt hyperparameter tuning procedures to boost classification performance, evaluated in well-known metrics (Equation 12 to Equation 18)[1,2,57].

The first procedure is the grid search cross-validation. We list the values of the hyperparameters which we would like to test our models with, and we evaluate every combination. At the end of the process, we choose the values attaining the best classification performance on the K-fold cross-validation.

The second one is called random search cross-validation. The process is like the grid search. Nonetheless, we pick the hyperparameters utilising a heuristic search over random values.

The hyperparameter tuning processes rely upon pseudo-random number generation, such as selecting the data points belonging to training and test sets or the K-fold cross-validation. Hence, scientists are used to setting the random seed to make the experiments reproducible and to look at the improvements derived from tuning the hyperparameters[1,2,57].

3.15. Transfer learning

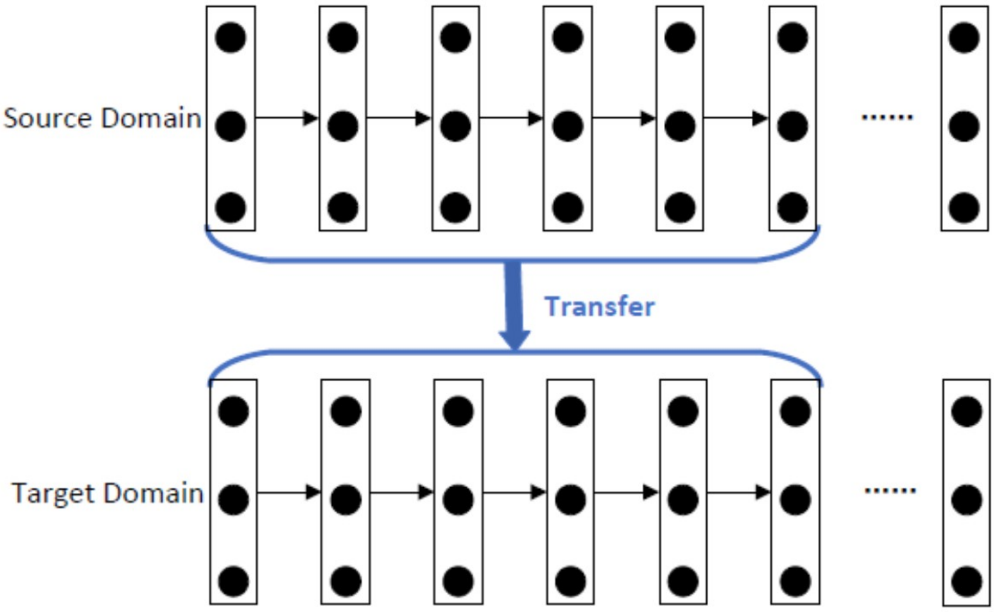


Figure 27. The transfer learning process[66]

Specific contexts, like medicine, present difficulties in collecting data in sufficient quantity to train a learning algorithm due to the need for more human experts or the time required. Transfer Learning (TL) overcomes these difficulties, starting from the assumption that the training data must be independent and identically distributed concerning the test data and belong to the same domain as the test data[59,66]. Thus, one can train a CNN in a given domain and reuse it in whole or part. Accordingly, we perform pre-trained structure and parameters transfer to a different domain. The result is a network retaining prior knowledge that, once fine-tuned, can learn new tasks efficiently. By domain, we mean both the type of data used (e.g., different image types) and the information in it, such as tissues belonging to different categories (e.g., brain, skin and lungs). This idea relies on the assumption that neural networks function similarly to the human brain, which continuously implements abstraction processes in different domains (Figure 27)[1,2,57]. Therefore, it is possible to find different open-source models of pre-trained neural networks in the literature, such as LeNet, AlexNet, VGG, Inception and ResNets, and adapt them to other domains[2,67].

TL was extensively adopted in this doctoral thesis to overcome small-sized dataset challenges.

3.16. The ResNet architecture

ResNets are deep CNNs that classify an image by returning a label as output. They owe their name to specific blocks that characterise their structure, consisting of residual connections. They were born to counteract the problem of vanishing gradient or explosion that occurs in the presence of many layers[67]. One speaks of gradient explosion when, during backpropagation, the gradient assumes values so large that the CPU cannot manage them, causing an overflow. The vanishing gradient is the opposite case, and the gradient assumes minimal values such that there is a slowdown in the training phase: one immediately reaches convergence without being able to advance in learning [67].

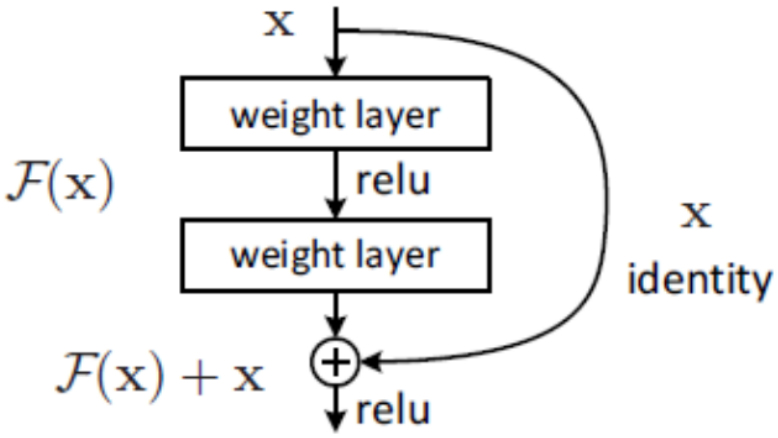
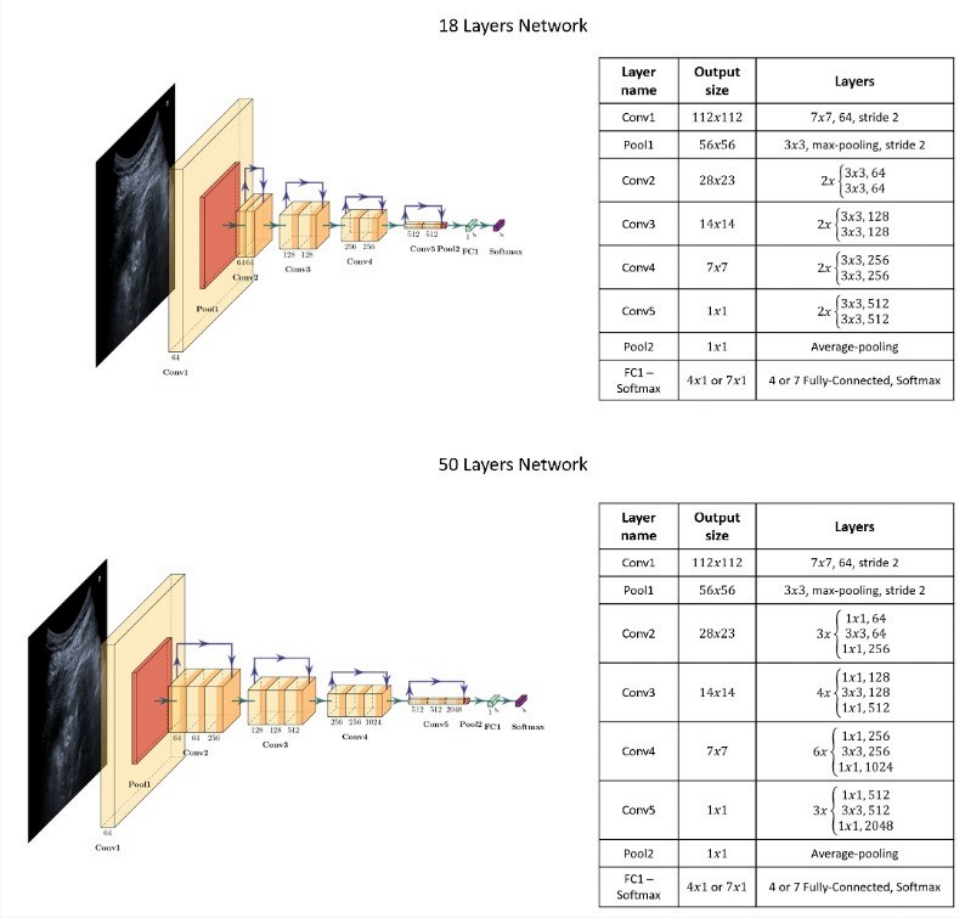


Figure 28. Residual block[67]

As shown in Figure 28, a residual block approximates the output H(x) by summing the identity function x to that derived from the underlying layers F(x), using a *skip-connection*. In this way, during the backpropagation phase, the gradient propagates backwards no longer one layer at a time, but by travelling through the residual connections, it reaches the initial layers more efficiently and quickly, skipping the intermediate layers. It is possible to stack hundreds of blocks of residual connections and obtain ResNet with different depth levels[67].

This doctoral thesis adopted deep residual networks, among others, to achieve the best and most reliable classification performance, avoiding vanishing gradient problems and allowing for deeper architectures than the commonly used ones, which do not exploit residual connections. The manuscript introduced residual architectures and skip-connections in this chapter because they represented a breakthrough in AI. Researchers have described using already proven models as a more rational approach for initiating DL model development from scratch[59,61,66]. Remarkably, we

selected two residual networks with 18 and 50 layers each and structured them as reported in the original paper[67]. Likewise, we extensively exploited transfer learning (Section 3.15) to significantly improve the classification results by exploiting features belonging to pre-trained networks. Accordingly, we selected ResNet-18 and ResNet-50 architectures, which had already undergone optimisation based on the ImageNet dataset[58]. Regardless, we made a few modifications to these networks before using them; we changed the last fully connected layers because they had as many neurons as the number of classes to be detected.



Layer name	Output size	Layers
Conv1	112x112	7x7, 64, stride 2
Pool1	56x56	3x3, max-pooling, stride 2
Conv2	28x23	$2x\begin{cases}3x3, 64\\3x3, 64\end{cases}$
Conv3	14x14	$2x\begin{cases}3x3, 128\\3x3, 128\end{cases}$
Conv4	7x7	$2x\begin{cases}3x3, 256\\3x3, 256\end{cases}$
Conv5	1x1	$2x\begin{cases}3x3, 512\\3x3, 512\end{cases}$
Pool2	1x1	Average-pooling
FC1 – Softmax	4x1 or 7x1	4 or 7 Fully-Connected, Softmax

Layer name	Output size	Layers
Conv1	112x112	7x7, 64, stride 2
Pool1	56x56	3x3, max-pooling, stride 2
Conv2	28x23	$3x\begin{cases}1x1, 64\\3x3, 64\\1x1, 256\end{cases}$
Conv3	14x14	$4x\begin{cases}1x1, 128\\3x3, 128\\1x1, 512\end{cases}$
Conv4	7x7	$6x\begin{cases}1x1, 256\\3x3, 256\\1x1, 1024\end{cases}$
Conv5	1x1	$3x\begin{cases}1x1, 512\\3x3, 512\\1x1, 2048\end{cases}$
Pool2	1x1	Average-pooling
FC1 – Softmax	4x1 or 7x1	4 or 7 Fully-Connected, Softmax

Figure 29. Residual Network Structure Diagrams: plot of each ResNet employed together with their structure and exploited layers[15]

Figure 29 displays the two architectures employed in this study. ResNet-18 and ResNet-50 take input images undergoing a first step consisting of a 7×7 convolution with a feature size of 64 and a stride of 2, followed by a 3×3 max-pooling step with the same stride. Next, each of the following layers performs either 3×3 or 1×1 convolutions with a fixed feature map dimension for the first residual network, namely $F_{ResNet-18} = [64,128,256,512]$, and with an increasingly repeated pattern for the second residual network, that is, $F_{ResNet-50} = [F, F, 4F]$ with F following the fixed

feature map order mentioned above. The input is bypassed every two convolutions for ResNet-18 and every three convolutions for the other residual architecture[15]. Width and height remain constant throughout the section because padding and stride are equal to 1 during these operations, allowing the connection to skip. The residual models exploit batch normalisation to improve regularisation together with the pooling layers. ReLU is the activation function. Eventually, the 18-layers residual network has 11.174 M parameters, while the 50-layers network consists of 23.521 M parameters.

3.17. The U-net architecture

The U-net is a CNN performing semantic segmentation, named after its peculiar U shape. The architecture comprises a contraction path, namely the encoder, which encrypts the input image by reducing it to a feature vector, and a symmetrical expansion path, the decoder, which decrypts the previously contracted information to restore its initial dimensions[68]. The contraction path consists of repeated blocks made up of two convolutions, each followed by a ReLU and a max pooling layer. At each down-sampling step, the number of feature-maps channels doubles. In the expansion path, each up-sampling step involves a transposed convolution halving the number of channels of the feature maps, and a skip-connection[67] (i.e., residual connection) with the feature map of the symmetrical contraction step. Then, it has two convolutions, each followed by a ReLU. Following this path, from left to right, given an input image, we obtain in output a segmented image of the exact dimensions as the original (Figure 30)[68].

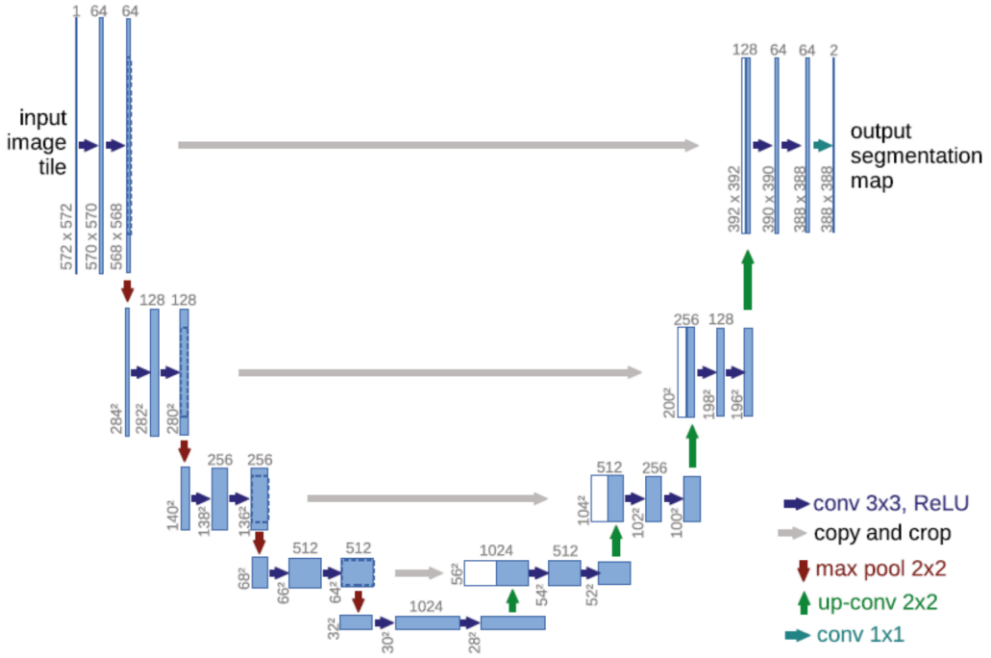


Figure 30. U-Net architecture[68]

The U-net++ is an evolution of the U-net and is essentially a supervised encoder-decoder network in which the contraction and expansion parts bond through dense, nested skip-connection and convolutional paths (Figure 31.a)[69]. This dense network of connections optimises the network training process and enables the generation of segmented masks with a high level of detail accuracy, even in chaotic backgrounds. Academics compared U-net++ with U-nets of various sizes for the segmentation of different images in the medical field, and they reported that U-net++ provided better results than the basic model[69]. Skip connections between semantically similar feature maps enhance the semantic segmentation procedure, unlike U-nets, where high-resolution characteristics directly link from the encoder portion to the decoder portion. The idea is thus to bridge the semantic gap between the encoding and decoding portions to facilitate the backpropagation process. For example, as shown in Figure 31.b, the semantic gap between block $x_{0,0}$ and block $x_{0,4}$ is bridged by inserting three convolutional blocks ($x_{0,1}$, $x_{0,2}$, $x_{0,3}$). The graphs show in black the U-net that makes up the skeleton of the network, in green the dense convolutional blocks positioned on the path furrowed by the skip-connections (blue lines), and in red the supervision. The red, green, and blue components differentiate the U-net++ from the U-net. Furthermore, in inference, the U-net++ can be reduced when trained with deep supervision (Figure 31.c)[69]. Deep supervision allows the model to operate in two ways[69]:

- In accurate mode, the final segmentation map is calculated as the average of the outputs of all branches
- In fast mode, the final segmentation map is selected from a single segmentation branch. The branch choice determines the model's pruning extent and training speed

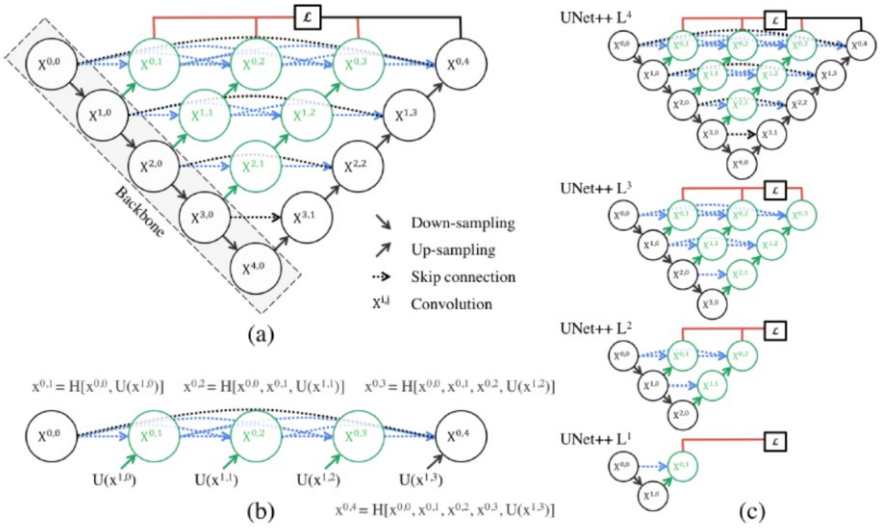


Figure 31. U-Net++ architecture[69]

3.18. The DeepLab architecture

The DeepLab architecture comprises naïve decoder networks. This term describes the type of up-sampling used for feature map generation. The basic idea is to extract features by employing a convolutional encoding structure and then restore the original image's spatial resolution through bilinear interpolation[70]. In DeepLab networks, the segmented image is obtained by up-sampling the output of the last convolution layer and calculating the loss function for each pixel. The up-sampling occurs after the encoding phase, during which the resolution of the input image significantly decreases. Oversampling, namely the enlargement of the image's spatial resolution, is carried out through unique atrous convolutions derived from the French *à trous* (i.e., with holes). The atrous convolution makes it possible to enlarge the field of view of the filters without increasing the number of parameters or the computational cost (Figure 32)[70]: zeros convey the space between kernel elements.

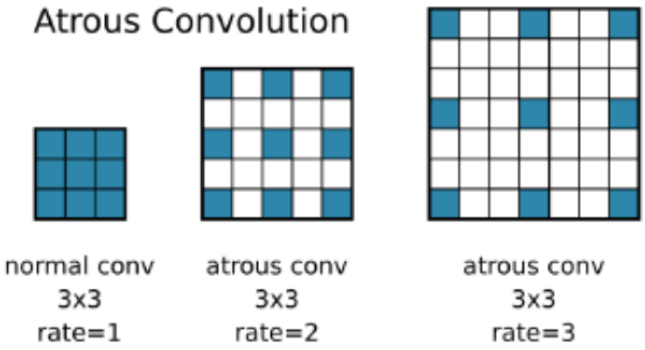


Figure 32. Generic filters featuring atrous convolutions[70]

DeepLabs come in different versions: DeepLabV1, DeepLabV2, DeepLabV3, and DeepLabV3+. Each version represents an evolution of the previous model[70–73].

The DeepLabV1 network, compared to a traditional Deep Convolutional Neural Network, overcomes the problem of excessive feature resolution reduction, and improves segmentation accuracy. It takes an input image, passes it to a Deep Convolutional Neural Network, followed by one or two layers performing hole convolution, and obtains a coarse score map[70]. This map is subsequently over-sampled through bi-linear interpolation, resulting in an image larger than the original. In the end, a fully connected Conditional Random Field is applied to improve the segmentation results. This model can couple neighbouring nodes, favouring the assignment of the same label to pixels that are spatially close to each other (Figure 33)[70].

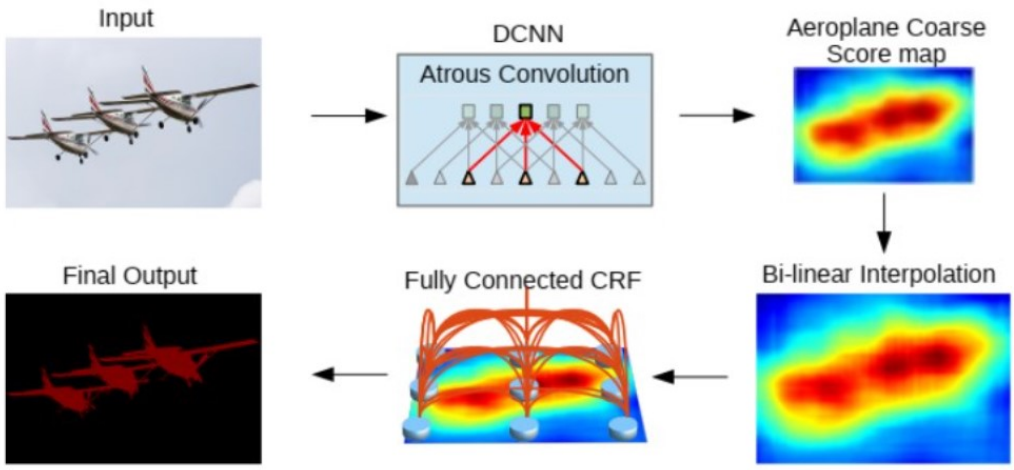


Figure 33. Steps followed by the DeepLabV1 network[70]

The DeepLabV2 network aims to improve the performance of the previous version and achieve a more robust segmentation. To this end, Atrous Spatial Pyramid Pooling occurs[71]:

- Several hole convolutions occur in a feature map
- Each atrous operation has a kernel that differently expands
- The result derives from the merging of the two previous steps

This method makes it possible to improve accuracy when several instances of an object in an image belong to the same class but with a different scale factor (Figure 34)[71,72].

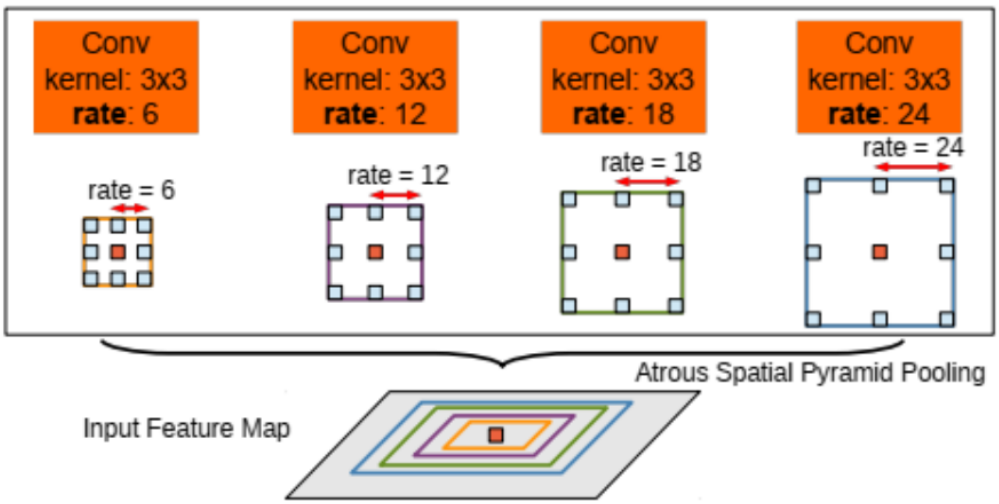
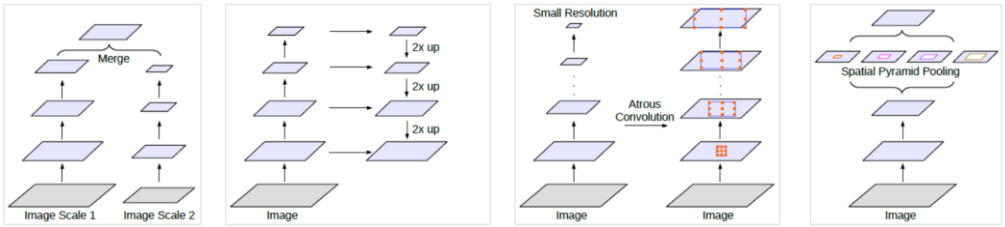


Figure 34. Atrous Spatial Pyramid Pooling[71,72]

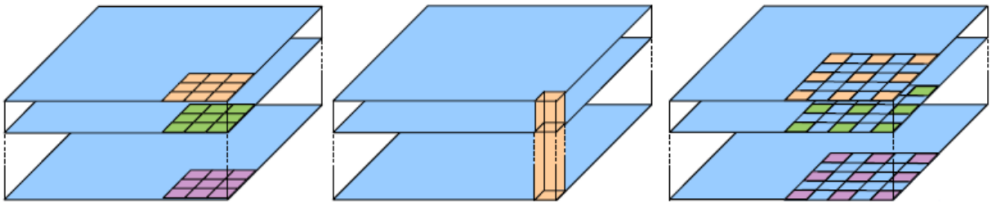
DeepLabV3 further improves performance by attempting to delineate sharper object boundaries, especially in the presence of objects on multiple scales (Figure 35.a). It experiments with new encoder-decoder type architectures in which feature maps' size gradually decreases, capturing higher semantic information, and equally gradually, spatial information is recovered (Figure 35.b). It makes more extensive use of atrous convolutions, testing their operation both in cascade (Figure 35.c) and parallel (Figure 35.d)[73].



(a) Image Pyramid (b) Encoder-Decoder (c) Deeper w. Atrous Convolution (d) Spatial Pyramid Pooling

Figure 35. DeepLabV3 architectures[73]

Finally, the DeepLabV3+ extends the DeepLabV3: it retains the DeepLabV3's encoding module and adds a decoding module to refine segmentation results, especially along the edge of objects[72,73]. For this purpose, it employs the atrous depthwise separable convolutions that decompose a standard convolution into a depthwise convolution (Figure 36.a) and a pointwise convolution (Figure 36.b). The former applies the same filter to each input channel, combining the outputs of the depthwise convolution across the various channels. In addition, depthwise convolution occurs through a perforated filter so that one can refer to it as atrous depthwise convolution (Figure 36.c).



(a) Depthwise conv. (b) Pointwise conv. (c) Atrous depthwise conv.

Figure 36. Atrous depthwise separable convolution[73]

In contrast to DeepLabV3, in the encoding phase, atrous depthwise separable convolutions replace all max pooling operations, whilst in decoding, there are a series of fixes on some layers to obtain an output mask that had the exact spatial resolution as the input image (Figure 37)[73].

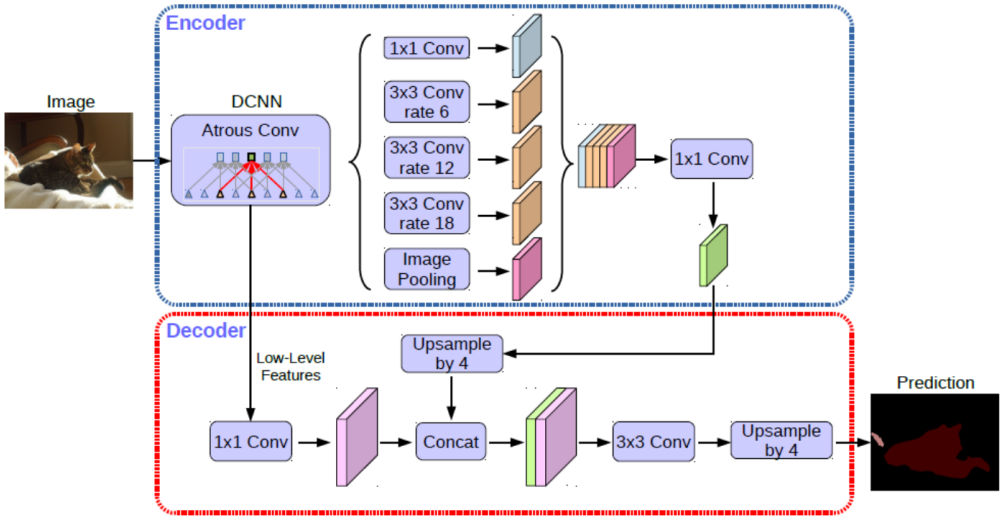


Figure 37. DeepLab V3+ architecture[73]

3.19. Generative Adversarial Networks (GANs)

Generative adversarial networks (GANs) are famous architectures used for generative modelling. GANs consist of two networks: a generator (G) and a discriminator (D) (Figure 38)[74].

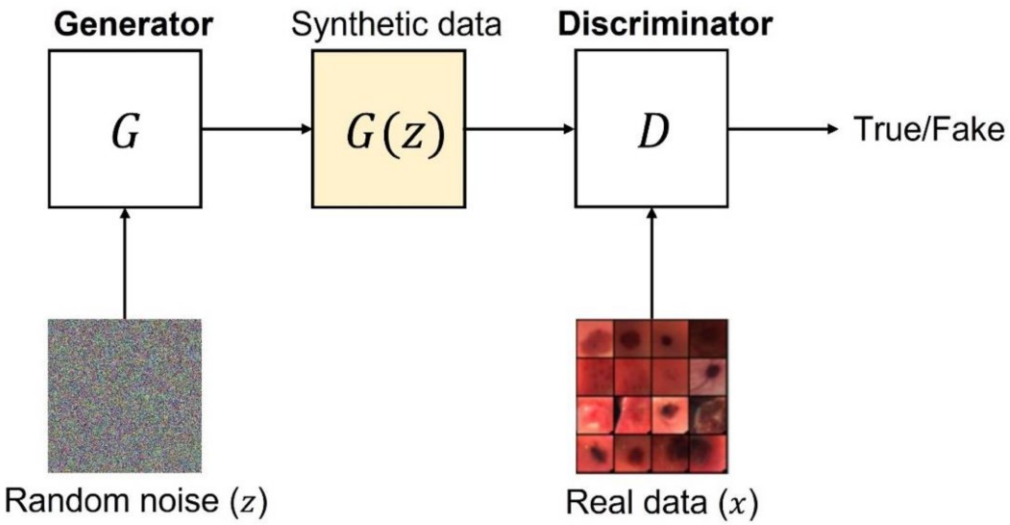


Figure 38. GAN standard structure[56]

The intuition is that G iteratively tries to map a stochastic input distribution to a target data distribution to generate new data, which D assesses as real or fake. Depending on the feedback from D, G tends to minimise the loss between the two distributions, thus generating similar samples as input data. The goal is to trick D into classifying generated data

as real. Both networks train simultaneously to get better at their respective tasks: while G is learning to fool D, D is concurrently learning to distinguish better fake from real data. D and G are generally CNNs trained in an adversarial setup[56,74].

Unlike CNNs, adversarial learning is a relatively recent idea. Regardless, it rapidly spread in medical imaging to overcome the small-sized dataset problems discussed in Section 2.12.

The original GAN architecture suffered several disadvantages, such as irregular training[74]. Consequently, intensive research in computer vision brought substantial progress by either modifying the architecture of D and G or exploring new loss functions[2,74,75]. A manner to nicely handle the data generation process in GANs is to supply extra information about the desired output properties, such as examples of the desired target labels. This knowledge supply is known as conditional GANs (cGANs), and it represents a form of supervision since it demands aligned training pairs[1,2,56,74,75]. However, the real strength of GANs relies on their ability to learn semi-supervised or fully unsupervised. Mainly, where aligned and properly annotated image couples are rarely available in medical imaging, GANs play a crucial role.

So far, in the medical imaging field, GANs have been chiefly applied to synthetic image generation for data augmentation[75–77] and multi-modality image translation. Concerning data augmentation, literature believes that GAN-based models have the potential to better sample the whole data distribution and generate more natural images than traditional approaches (e.g., rotation and flipping), which may contribute to higher models generalizability and more efficient training[1,3,54].

The present doctoral thesis adopted the original GAN model proposed by Goodfellow et al. in 2014[74]. The generator G inputs a latent space vector z from a standard Gaussian distribution and produces a sample G(z) representing the mapping from a latent space z to the actual data space.

On the one hand, G trains to estimate the training data distribution and generate synthetic samples with the same real data distribution.

On the other hand, discriminator D receives the synthetic data produced by G or a sample (x) from the real dataset as input. D outputs the probability estimate concerning the input data source. Specifically, it estimates whether the sample came from the training data or G. G and D play a *minimax game*, where G tries to minimise the probability that D will predict its outputs as fake, whilst D tries to maximise its probability to discriminate between real and fake samples correctly[2,74,75].

Researchers proposed several network architectural topologies to implement G and D, including Vanilla GAN, BiGAN, infoGAN, variational autoencoder network GAN (VAEGAN), and deep convolutional GAN[56]. As we mentioned, deep convolutional neural networks have recently emerged as stable and affordable architecture for synthetic image generation[56]. This architecture adopts two convolutional networks, G and D. Remarkably, G consists of transposed convolutional layers, while D comprises standard convolutional layers.

Considering HS images, the conversion from z to the data space performed by G consists of creating synthetic HS images with the training images' exact spatial and spectral dimensions. Since we employed the skin cancer dataset described in Section 2.7 as a training set, G should generate an image whose sizes are $50 \times 50 \times 116$. Figure 39.a shows the G architecture and the sizes G adopted in this thesis. A batch normalisation and ReLU activation function follow the deconvolutional layers from 1 to 6. Finally, the last deconvolutional layer adopts the Tanh activation function.

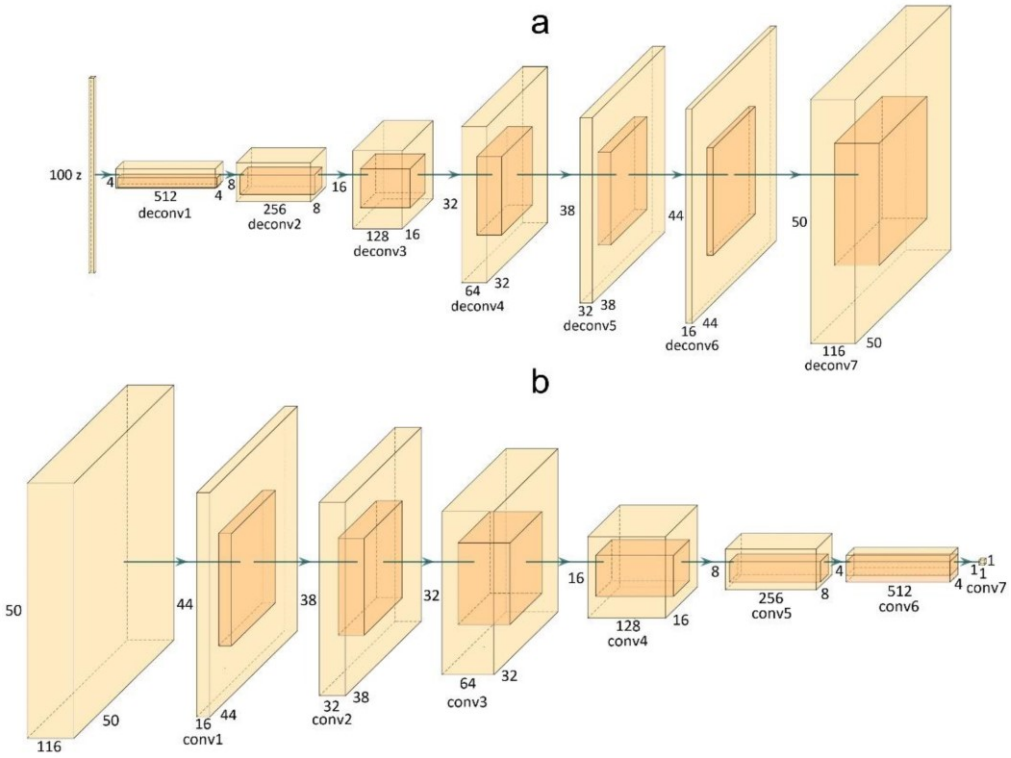


Figure 39. Proposed generator (**a**) and discriminator (**b**) architectures[56]

On the other hand, D receives as input an HS image with the same size, $50 \times 50 \times 116$, and performs a binary classification to determine if the input image is real or fake. For this reason, this network comprises standard convolutional layers. Figure 39.b depicts D's architecture detailing each convolutional layer's size. The leaky ReLU activation function characterises the first convolutional layer. The layers from 2 to 5 feature batch normalisation and leaky ReLU activation function. All the leaky ReLU functions adopt a negative slope equal to 0.2. The sigmoid function characterises the final convolutional layer[56].

3.20. Concluding remarks

This chapter provided an overview of AI focusing on medical imaging analysis, paying attention to the critical methodological concepts adopted in this doctoral work. All the models and methodologies suffer from the challenges described in Section 2.12 in medical environments.

The following chapters will dive into the details of the works carried out during the educational path described in this manuscript, exploiting the algorithms and technological advancements described in this manuscript.

Chapter 4

CUDA programming basics

This chapter covers the basics of the CUDA Graphical Processing Unit (GPU) programming language and its main libraries, which took part in this doctoral thesis. CUDA is a programming language designed and developed by Nvidia engineers to provide access to a CUDA-capable GPU device's hardware to run general-purpose code that can utilise the massively parallel architecture to accelerate applications. The general term for this is GPGPU which stands for General Purpose Graphical Processing Unit to distinguish them from the types of operations that run on graphics cards, including 3D rendering and video decoding.

The ongoing technological High-Performance Computing (HPC) growth, mainly concerning hardware like GPUs, drives traditional healthcare innovation and transformation into personalised medicine, capable of managing the big data and the models described in the previous chapters. The challenge involves experimenting with new approaches to gathering, managing, and transmitting data to deliver a renewed direction to medical research. In this innovative context, computational aspects are crucial since they concern every aspect of healthcare. For instance, simulators let patients better comprehend the surgery or the therapy they will face. They also provide surgeons with a more practical breakdown and operation preparation tool. Regardless, all the goals met by such settings are only possible thanks to the high-performance computing hardware provided with them[11,12].

Another example of the research supported by HPC stands in what we can consider another popular buzzword of the last decade besides AI, the digital twin. It consists of a detailed mathematical model allowing the real-time simulation of its natural counterpart, which could consist of a living organ or an electrical machine for autonomous driving. In healthcare, the digital twin improves therapy personalisation and better data-driven decisions, precluding medical difficulties before they may occur[11].

Two main factors contribute to these substantial visionary systems' continuous design and development: fast data availability and technological infrastructure.

GPU architectures comprise thousands of cores, spreading the workload among these processors that work in parallel. Researchers operate multi-GPU systems or supercomputers to increase the number of available cores,

reducing elaboration time. This doctoral thesis extensively operated GPUs for model training and their embedded deployment onto low-power hardware[78].

4.1. Accelerated computing

The number of AI medical applications leaning on more than a single processor's computational capacity and power is rising. Fast or even real-time reaction is essential in healthcare, and AI computational complexity needs many cores architecture to grow and manage big data. This element leads to assessing efficient technologies to manage vast amounts of data in constrained computational times. Consequently, a different way of programming exists to transform a sequential into a parallel software program, where various threads cooperate to complete the elaboration rapidly. Two philosophies exploit parallel hardware: *multi-core* and *many-core*. In multi-core programming, the processor elaborates both the code's serial and parallel parts. A single thread executes the former, whilst many-core comprises several threads for elaboration. In the many-core philosophy, each core manages one thread since these technologies host hundreds or thousands of cores[11].

4.2. Comparison between CPU and GPU

GPUs are many-core architectures, representing the dominant device for parallel computing. The first dedicated graphics chips were produced to output the 2D display and assist with bitmap rendering operations. This production coincided with the introduction of graphically driven operating systems of the late 80s and early 90s. Resolutions and colour palettes gradually improved with improved display technology. As hardware became powerful, companies produced dedicated 3D graphics workstations to spread in government, defence, scientific and technical industries, as well as producing visuals and special effects for the media and creative industries. By the mid-90s, consumer applications and games employing 3-D graphics were becoming popular, and companies like Nvidia released their competing products around this time.

The first standardised and platform-independent 3D graphics API, OpenGL (Open Graphics Language), was released in 1992 by Silicon Graphics. Developers no longer had to utilise different proprietary standards to support competing devices. Regardless, they could develop code obliging for one standard able to run on all compliant cards with a guaranteed set of features defined by the version compliance, with all graphics card manufacturers eventually supporting both DirectX and OpenGL. Eventually, as hardware became faster and more complex, the graphics devices moved from a fixed function pipeline to a more programmable pipeline that included components like pixel shaders and

vertex shaders, which allowed for greater versatility in 3D rendering techniques. Designers intended pixel shaders to work on textures in 3D coordinates, but developers soon found a way to harness the computational power of GPU for non-graphical algorithms. This process gave rise to the term GPGPU, short for the general-purpose graphical processing unit. Nvidia took notice of this, and the next evolutionary step on the way to full GPGPU was to deliver the programmable functionality and parallel computing power of the GPU. The company released a dedicated parallel computing API named CUDA, which allowed developers to utilise the GPU for computational work without going through the graphics API or being restricted to using textures for data. It also allowed arbitrary memory reads and writes and a full suite of debugging and profiling tools included in the software development kit. Applications of GPGPU include medical imaging, raytracing, fluid dynamics, cryptocurrencies, and AI.

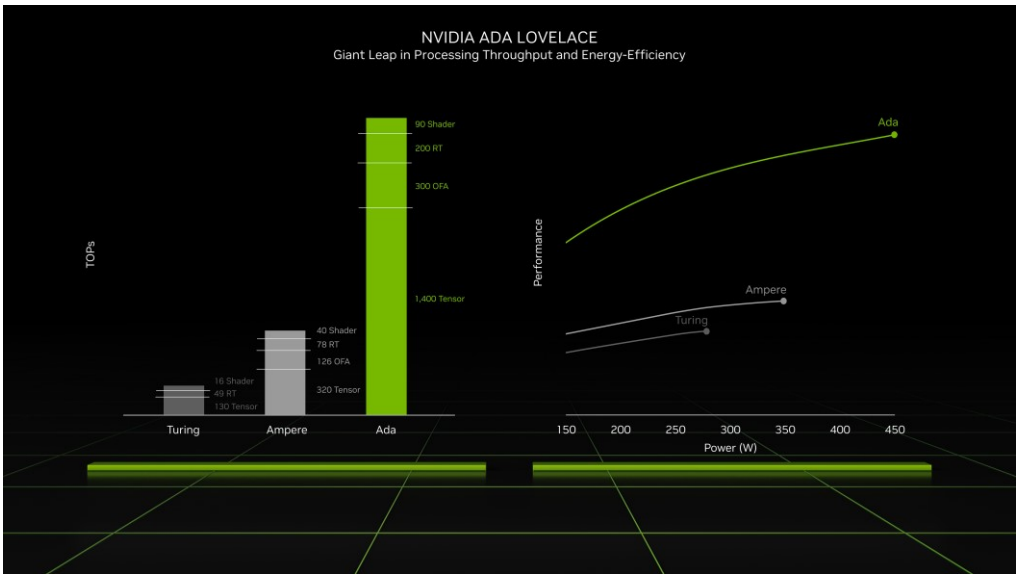


Figure 40. Nvidia ADA Lovelace Tera Floating Point Operations Per Second

To better explain the computational capabilities of CUDA-programmable GPUs and CPUs, Figure 40 shows the evolution of the peak double-precision floating-point operations per second (TOPS) on recent GPU architectures associated with power consumption. This graph is not an accurate performance comparison, but it is possible to appreciate how much the performance gap has grown over the years. This difference led developers to harness the GPUs' power to execute their programs' most computationally intensive parts[11,78].

Designers produced many GPU architectures over the years, representing one of the main reasons for the computational gap concerning CPUs (Figure 41). The CPU maximises sequential programs' performance and comprises a sophisticated control unit that distributes the workload to

threads but maintains a serial aspect. Big cache memories ease the latency of memory and instruction accesses. Another difference for CPUs is backward compatibility with several operative systems, applications and I/O procedures. Since GPUs do not have to meet these constraints, their memory model is more straightforward. GPU developers increase memory bandwidth, speeding up data and instruction transfers, and most chip area is dedicated to several ALUs to maximise the computational throughput. Indeed, while CPUs optimise latency, namely the time required for operation completion, GPUs seek throughput, namely the number of operations completed concurrently[11,78]. This arrangement makes the control units and the cache memories smaller than the CPU ones (Figure 41). GPU architecture allows threads to elaborate data while others perform memory accesses, and several small-sized caches increase memory bandwidth. Although the GPUs optimise the computational performance, some tasks are more efficient if the CPU runs. For this reason, most applications use the CPU for the sequential part of the code and the GPUs for the most intensive parts.

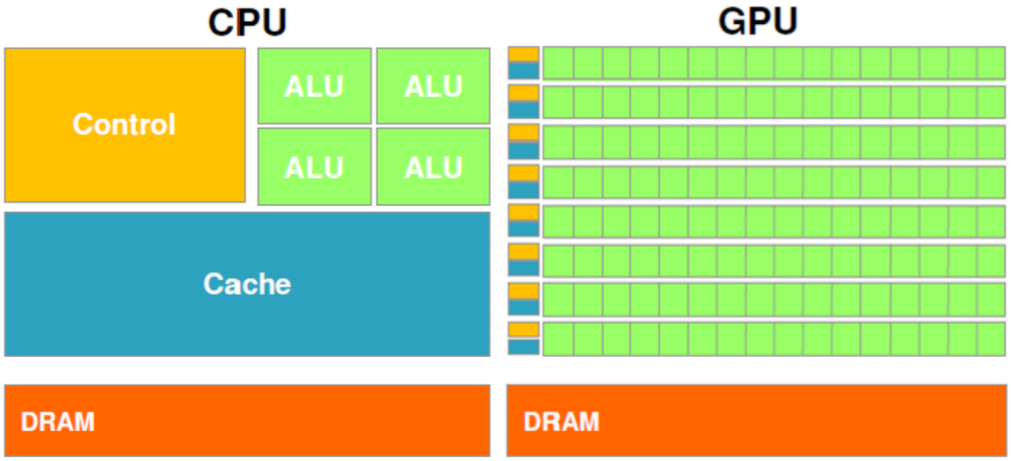


Figure 41. CPU and GPU architecture comparison[78]

Nvidia equips different GPUs generations with hundreds of cores organised in the so-called Streaming Multiprocessors (SMs). They can be considered similar to a separate CPU core, with each SM having access to its local memory pool, cache and registers (Figure 42). Furthermore, SMs comprise several processing cores called CUDA cores under the control of several warp schedulers, which fetch instructions and execute them on the multiprocessor. Different graphics card models from the same family will often differ by the number of SM they contain. However, developers usually fix the number of CUDA cores in SMs for a given micro-architecture. For example, Maxwell and Pascal families contain 128 CUDA cores per multiprocessor. Table 5 displays how the CUDA cores number has historically changed between different designs. For example, in the

previous Kepler architecture, the number of CUDA cores per multiprocessor was 192; on Fermi, before that, it was 48.

Table 5. GPU architectures evolution across the years

CUDA Architecture Codename	Release year	CUDA cores per SM
Tesla	2006	8
Fermi	2010	48
Kepler	2012	192
Maxwell	2014	128
Pascal	2016	128
Volta	2017	64
Turing	2018	64

Two types of memories characterise GPU architectures. The global memory can be accessed by a thread independently from its block and is used to exchange data between CPU and GPU through suitable CUDA functions. Since it takes a long time to access this memory due to the low access bandwidth, developers must design an efficient strategy during the data transfers between GPU and CPU. The on-chip memories, shared memory and registers have a low access latency and a high bandwidth. Registers are private to each thread, whilst shared memory is private to all block threads. The usage of the on-chip memories allows for reducing the high cost of accessing the global memory[11,78].

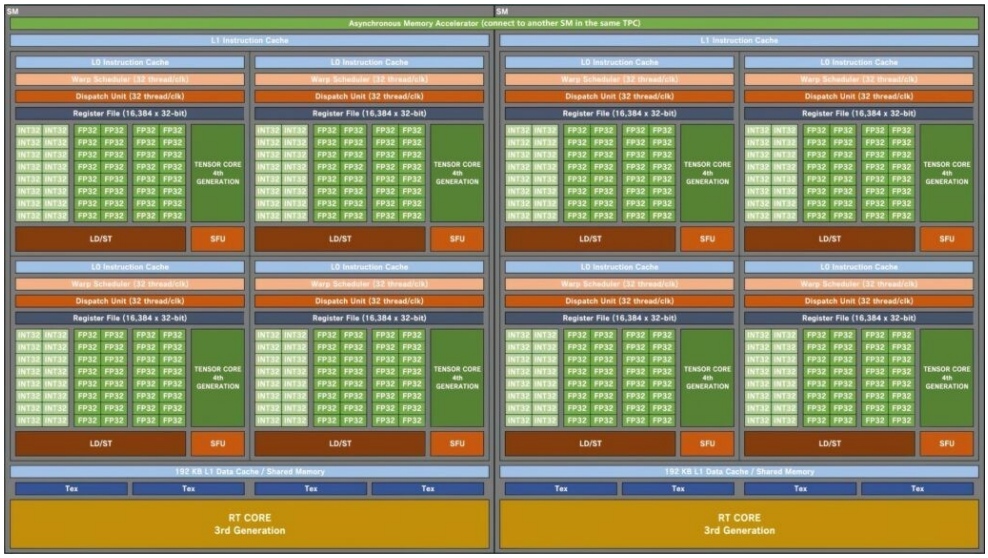


Figure 42. Most recent Nvidia Ada Lovelace's Streaming Multiprocessors scheme

4.3. Computer Unified Device Architecture (CUDA) basics

CUDA is an acronym for *Computer Unified Device Architecture*, and Nvidia first introduced Version 1.0 of their proprietary CUDA C programming language with the Geforce 8 series of cards: the GTX 8800. It was the first graphics card released to be CUDA capable under a new micro-architecture called Tesla in June 2007 with a designated Compute Capability 1.0, which is different from the software version. Nvidia assigns a compute capability version number to each new micro-architecture it releases to designate the hardware features that the device supports. When Nvidia designed the CUDA language, it relied on the C language with some extensions. It should be easy to get started if one is familiar with the C or C++ languages. Developers designed CUDA so that someone with no prior knowledge of OpenGL or DirectX can tap into the GPGPU capabilities of the device directly without much extra effort. To run CUDA programs, it is necessary to have an Nvidia device installed as it is proprietary software[78].

CUDA remains widely used in industry and academia and has access to a wide array of Nvidia-developed and third-party libraries and applications written for it. New toolkits with new software features and hardware support have been added regularly since the first release, with the current version being CUDA SDK Version 12 as of the time of writing.

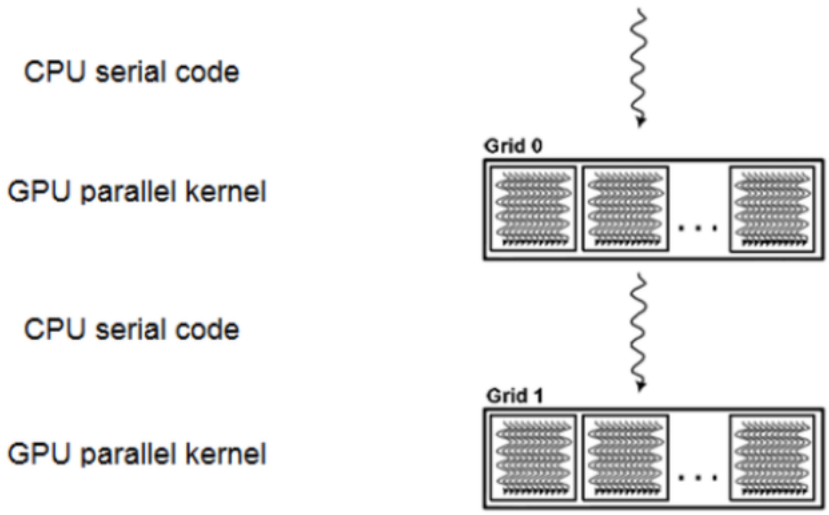


Figure 43. CUDA execution model[78]

The CUDA execution model (Figure 43) comprises a *host*, commonly the CPU, interacting with one or more *devices*, namely the GPUs, which speed up the software's computationally intensive parts. Therefore, a CUDA program's source code contains parts related to the host and the

devices, which lean on the same language (C/C++ and others like Fortran or Python). The device parts extend the primary language with the CUDA functions and instructions that run on the GPU. Nonetheless, the host and the device access separate memories, so an explicit allocation and transfer between the two memories are necessary.

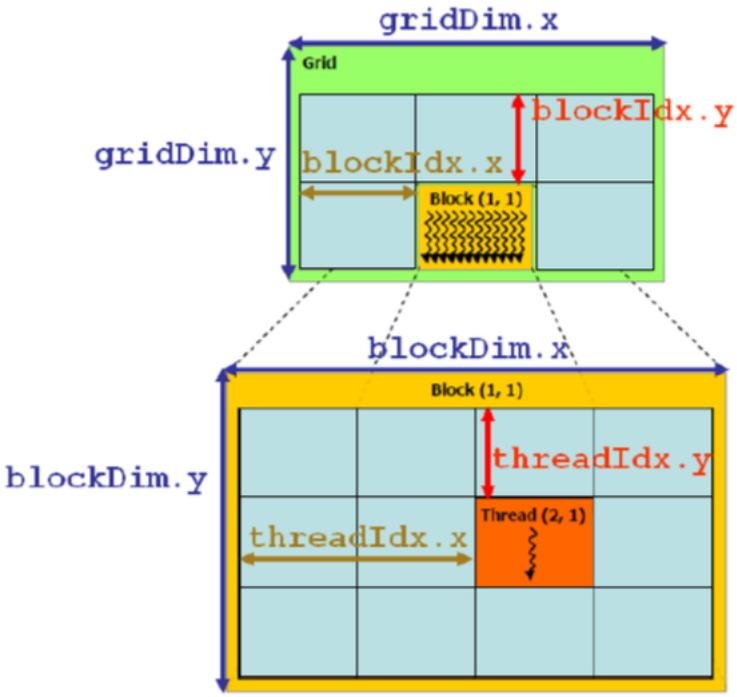


Figure 44. Threads, Grids and Blocks organization[78]

Developers define a kernel as a function working with parallel threads on the device's hardware. The CUDA program execution starts from the host that will invoke a kernel. At this point, the device generates many threads to exploit the parallelism. Threads organise in blocks which, in turn, form a grid (Figure 44). In parallel computing, a task's *granularity* (or grain size) measures the work (i.e., computation) performed by that task. Blocks, sequentially assigned to the various SMs, represent coarse-grained parallelism. Threads, instead, represent fine-grained parallelism. A thread can only belong to one block[78].

When a grid's threads end their execution, the software continues sequentially until the host invokes another kernel. On the other hand, when a kernel ends, it is essential to transfer its results from the device to the host memory and to release the GPU memory. Figure 43 displayed an example of the CUDA code execution model. This section mentioned execution passing from host to device when a kernel starts. In this process, the developer must perform a series of tasks. First, to allocate the memory on the device, the host uses the function cudaMalloc. This function has the following prototype:

- cudaError_t cudaMalloc (void** devPtr, size t size): *devPtr* is the pointer address and *size* the size in bytes to be allocated. *cudaError_t* is the function output that reports the presence of an error

Once allocated the memory, it is possible to transfer elements from host to device using the cudaMemcpy function:

- cudaError_t cudaMemcpy (void* dst, const void* src, size t count, enum CudaMemcpyKind kind): *dst* and *src* are the destination and the source memory addresses, respectively. The *count* is the elements' size in bytes to copy, and the *kind* defines the transfer type, identifying the source and destination. The transfer direction can be host-host (cudaMemcpyHostToHost), host-device (cudaMemcpyHostToDevice), device-host (cudaMemcpyDeviceToHost) and device-device (cudaMemcpyDeviceToDevice)

Once developers allocate memory and perform the data transfer, they invoke functions to execute computations on the data. Upon kernel activation, the scheduler cyclically assigns each grid block to an SM containing a specific number of architecture-dependent CUDA cores[78]. The scheduler will allocate unassigned blocks as soon as an SM terminates a block. As a result, blocks can be executed independently of each other so that no assumptions exist about their execution order. This feature of CUDA is called transparent scalability and is a massive advantage because the same code can be executed differently on hardware with different resources. All grid blocks must contain the same number of threads (maximum 1024 per block). Nvidia introduced this constraint to force developers to use more blocks and use the GPU's potential better. Blocks, however, are only conceptual units. The fundamental execution unit for the SM scheduler is the warp. The threads in each block are grouped into sets of 32 threads, called warps. Each warp is assigned to a core and can be executed independently of the others. When sizing blocks, it is a good practice to choose sizes multiples of 32[78].

Typically, when a kernel starts, data transfer from host to device and at the end of the kernel computation, data come back from the device to host. Once we initialise the device, we must also define the kernel's number of threads and blocks. CUDA provides variables to uniquely identify a block inside a grid using three-dimensional coordinates: blockIdx.x, blockIdx.y and blockIdx.z. Similarly, we can identify a thread in a block using threadIdx.x, threadIdx.y and threadIdx.z variables. It is possible to use only two dimensions in the thread identification, as shown in Figure 44. Furthermore, all blocks must have the same number of threads. Finally, blockDim and gridDim represent the blocks and grid dimensions. In the 2D scenario, developers may identify a single thread using the following Equation 19 and Equation 20:

$$\mathbf{i} = \mathbf{blockIdx.y} \times \mathbf{blockDim.y} + \mathbf{threadIdx.y} \quad \text{Equation 19}$$

$$\mathbf{j} = \mathbf{blockIdx.x} \times \mathbf{blockDim.x} + \mathbf{threadIdx.x} \quad \text{Equation 20}$$

If only one index has to be used to define a thread, it follows Equation 21:

$$\mathbf{index} = \mathbf{i} \times \mathbf{gridDim.x} \times \mathbf{blockDim.x} + \mathbf{j} \quad \text{Equation 21}$$

Furthermore, engineers can synchronise threads activity in a block through the syncthreads function. This way, all the threads reaching the block's barrier wait for the others before continuing the execution. Since synchronisation only affects a block, the system can execute the blocks randomly. For example, Figure 45 demonstrates how the same code might run differently on two boards equipped with a different number of cores. This computational advantage is called automatic scalability[78].

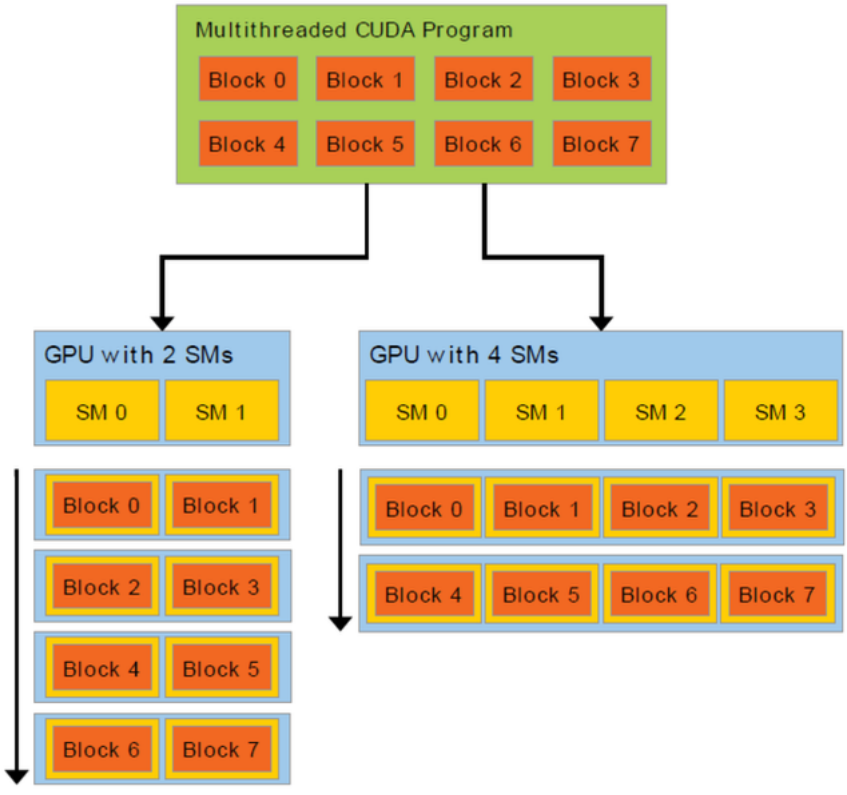


Figure 45. Automatic scalability[78]

At this stage of the program execution model, it is possible to invoke the kernel specifying the grid and the block dimensions:

- kernel <<<dim3 grid, dim3 block>>>(arg1, arg2, ...): (arg1, arg2,..., argN) are the function's parameters

Once the host invokes a kernel, each SM can manage up to 16 blocks depending on its resources. Each block assigned to an SM splits into groups of 32 consecutive threads (i.e., the warps) for execution, which, as we mentioned, must be 1024 at maximum. The most substantial portion of a CUDA code resides in the kernel activation, whose execution takes place on the GPU and can be activated either by the host or by the device itself. The __global__ identifier precedes the first case, whilst the __device__ one foregoes the latter.

Once the kernel execution completes, the result can be transferred from the device to the host by exploiting the cudaMemcpy function presented above, where the kind is cudaMemcpyDeviceToHost.

At the end of the device code execution model, the developer must release GPU's memory by invoking the cudaFree function:

- cudaError_t cudaFree(void* devPtr): devPtr indicates the device memory pointer to deallocate

Once the host invokes a kernel, it has two options: it can wait for the device's results and only then continues the program execution, or it can continue the execution immediately after the kernel call. The former involves synchronising the host and the device activity, whilst the latter implies an asynchronous run and kernel takeoffs can overlap with the host function calls. However, the cudaMemcpy is a synchronous function. Hence, CUDA provides a tool to exploit the concurrency and the ability to perform the CUDA kernel, the memory transfers, and the host operations simultaneously. Hence, developers must use a different CUDA function to transfer data from host to device asynchronously and vice versa: the cudaMemcpyAsync. This function requires the host's memory allocation through the cudaMallocHost.

The CUDA framework also offers streams for concurrent execution, enabling overlap of CUDA operations assigned to different streams. For instance, developers can overlap memory transfers with computation on the device or host to enhance performance. A cudaDeviceSynchronize function is used to block the host until the CUDA kernel completes and the results are returned when host needs the GPU computation result. As shown in Figure 46, streams can be used to overlap transfers and kernels in the sequence of host-to-device memory transfer, kernel execution and device-to-host memory transfer[78].

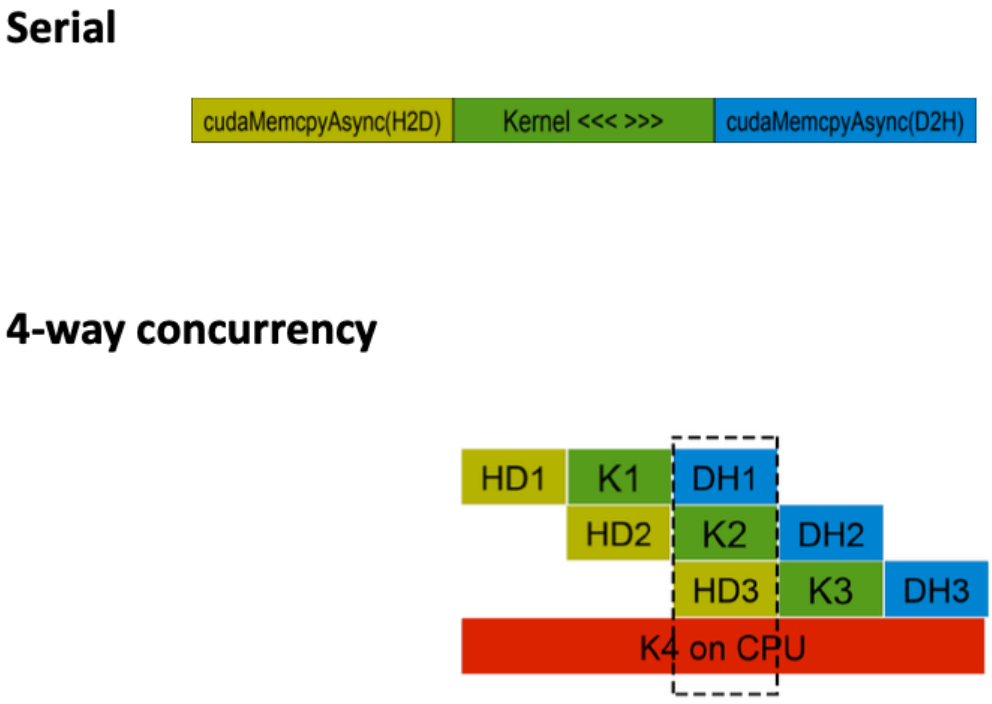


Figure 46. In serial execution, memory transfers and kernel are sequential. Streams allow to overlap tasks[78]

In summary, we can say that a CUDA program consists of a series of specific steps involving the host and one or more devices:

1. Declaration of variables and allocation of memory space on the device
2. Data transfer from the host to the device
3. Activation of the kernel
4. Data transfer from device to host
5. Deallocation of memory space on the device

4.4. CUDA program compilation

CUDA execution model provides the nvcc compiler to assemble a C-CUDA program. The host code compiles through a standard compiler such as gcc, whilst the device code runs through either an assembly form called Parallel Thread eXecution (PTX) or a binary form. Developers set the parameter compute capability to point the GPU architecture to the compiler, which contains two numbers indicating the architecture and the version, respectively[78].

4.5. CUDA example

Algorithm 1 is an example of a simple kernel performing an addition between two-dimensional matrices parallelised with CUDA C. The number of blocks (1D) equals the number of rows in the matrix. On the other hand, the threads (1D) per block correspond to the columns count in the matrix. This choice yields an activation of several blocks equal to the number of columns and, thus, as many CUDA cores as possible. This selection aims to optimise the operation's performance as much as possible.

Algorithm 1. CUDA kernel Matrix Addition

```
1. __global__ void add_matrix(float* mat_dev_val1, float* mat_dev_val2, int rows,
   int columns)
2. {
3.   int index = blockIdx.x * (columns) + threadIdx.x;
4.   if (threadIdx.x < columns && blockIdx.x < rows)
5.   {
6.       mat_dev_val1[index] = mat_dev_val1[index] + mat_dev_val2[index];
7.   }
8. }
```

As we can see, the add_matrix() function must be declared __global__ to run in parallel on the GPU.

The function takes as input as parameters two float pointers to devices previously allocated with a cudaMalloc(), which are the matrices we wish to sum. The matrices were previously stored in memory as monodimensional arrays and thus with a contiguous row-by-row allocation (row-major order). One of the matrices (mat_dev_val1) also stores the result, thus already overwriting the data it possessed but sparing memory usage.

The blocks and threads describe the matrix component's index, which we sum to exploit the GPU's parallelism fully.

The call of the function in the host part takes place in this way:

- add_matrix << <rows, columns >> > (mat1, mat2, rows, columns);

This kernel writing enables blocks as the number of rows and threads for each block as the number of columns. Nonetheless, the number of columns is a multiple of 32 less than 1024. We will then obtain the result in mat1, which we allocated on the device. If we wanted to check the result, a cudaMemcpy() would have transferred it to the host.

4.6. cuDNN library

As we previously mentioned, academics research solutions to complex and computationally heavy problems in healthcare. AI represents one of these, especially related to medical data, which comprise HSIs. Researchers must consider designing applications comprising big deep-learning models and data as complex as HSIs. These projects require hardware that delivers the highest throughput and lowest latency possible[11,12,18].

This doctoral thesis extensively adopted the cuDNN library, especially for embedding the designed models into low-power GPUs. The Nvidia CUDA Deep Neural Network library (cuDNN) is a GPU-accelerated library of primitives for deep neural networks. cuDNN provides highly tuned implementations for standard routines such as forward and backward convolution, pooling, normalisation, and activation layers. Deep learning researchers and framework developers worldwide rely on cuDNN for high-performance GPU acceleration. It allows them to focus on training neural networks and developing software applications rather than spending time on low-level GPU performance tuning. cuDNN accelerates widely used deep learning frameworks, including MATLAB, PyTorch, and TensorFlow[18,78,79].

The library comprises functions optimised for GPU execution, behaving like standard kernels that have already been parallelised and optimised automatically. Their activation assumes developers follow the execution model mentioned in the earlier section, involving all the steps from memory allocation to deallocation.

In addition, the functions of the cuDNN library receive specific types of parameters, the so-called *descriptors*, as input.

The developer specifies a series of information about the data that the cuDNN kernel will process within the descriptor-type variables. This information consists primarily of tensors that, qualitatively speaking, consist of N-dimensional cubes. Therefore, before activating any cuDNN function, we must create and initialise the necessary descriptors that the corresponding kernels need to perform their operations correctly[79].

First of all, we must create a handle to use the library[79]. Accordingly, we declare:

- cudnnHandle_t handle_name;
- cudnnCreate(&handle_name);

When we have finished using the library, we can delete this handle by calling:

- cudnnDestroy(handle_name);

The functions implemented within the cuDNN library require tensors as input, which are the pointers to the data we need to use to perform operations. Developers must allocate data contiguously in memory for

cuDNN to work because it is the only way to achieve maximum efficiency. Moreover, we must store it to respect a specific layout indicated by an enum data type called cudnnTensorFormat_t that we will use in the tensor descriptor itself[79]. CudnnTensorFormat_t, as far as tensors in 4D are concerned, may take the following values:

- CUDNN_TENSOR_NCHW
- CUDNN_TENSOR_NHWC
- CUDNN_TENSOR_NCHW_VECT_C

Assuming we are discussing images:

- N denotes the number of different images in a batch of DL network training (Section 3.11)
- C the number of tensor's channels of the tensor, which in the case of RGB images is equal to 3
- H is the height of the tensor, namely the image height
- W the width of the tensor, namely the image width

To better understand how we should store these tensors in memory, we can refer to these examples (Figure 47 and Figure 48):

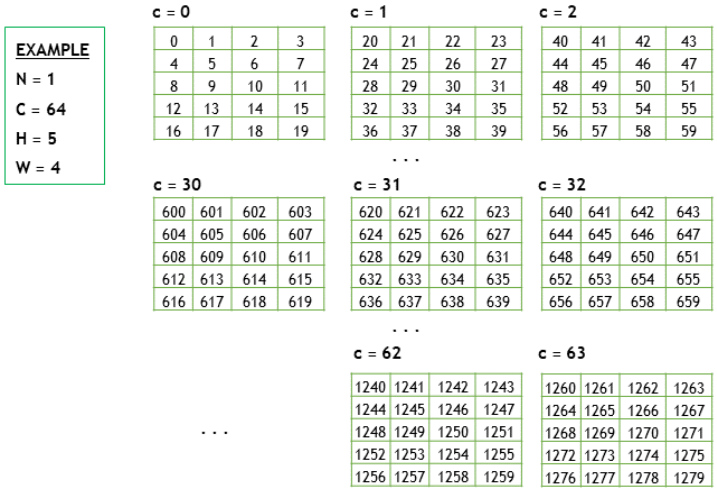


Figure 47. cuDNN tensor representation example[79]

Furthermore, Figure 48 contains the various possible layouts of the tensor in memory:

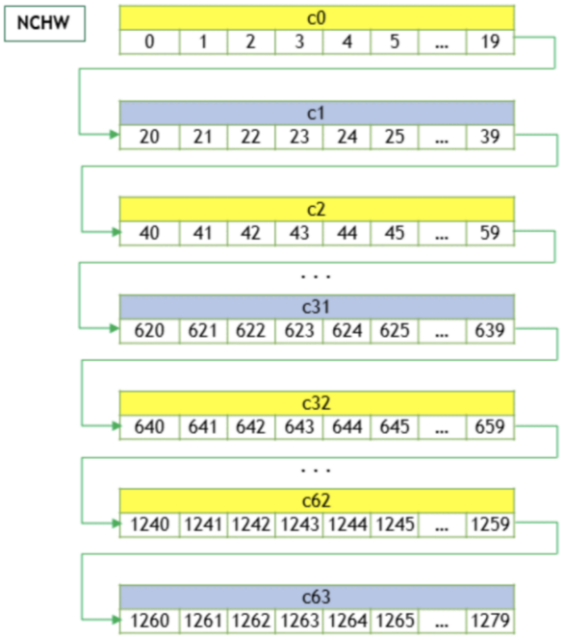


Figure 48. Tensors available memory layouts[79]

4.7. cuDNN example: convolution

In order to give a practical example of how the cuDNN library works, this section describes a step-by-step example of the development of a convolution layer in the inference phase. Therefore, we will encounter all the functionalities of the various tensors and descriptors expected by the input cuDNN kernels.

Algorithm 2. cuDNN forward convolution descriptor

```
1.  cudnnConvolutionForward(
2.  cudnnHandle_t handle,
3.  const void *alpha,
4.  const cudnnTensorDescriptor_t xDesc,
5.  const void *x,
6.  const cudnnFilterDescriptor_t wDesc,
7.  const void *w,
8.  const cudnnConvolutionDescriptor_t convDesc,
9.  cudnnConvolutionFwdAlgo_t algo,
10. void *workSpace,
11. size_t workSpaceSizeInBytes,
12. const void *beta,
13. const cudnnTensorDescriptor_t yDesc,
14. void *y)
```

The kernel function written above receives different parameters as input:

- The handle to the cuDNN library we mentioned in the earlier section
- The xDesc descriptor of the incoming feature map (see Section 3.9) and its values, which we allocated and stored on the device, x
- The wDesc descriptor of the filter and its values, which we allocated and stored on the device, w
- The descriptor of the convolution operation convDesc
- The algorithm (algo) adopted for the convolution calculation (i.e., atrous or standard) is a simple enum-type datum. Nonetheless, if the desired convolution does not exist in the library, a custom algorithm must be written
- A pointer to the GPU memory workSpace of size workSpaceSize, which is necessary for the computation's execution
- The descriptor of the tensor returned by the function. The kernel will store its values in the variable y, which the developer must allocate on the device
- The parameters alpha and beta are scaling factors belonging to the host memory. We usually set them equal to 1 and 0, respectively. The beta is different from zero if we wish to add a bias, and the alpha is different from 1 if we wish to scale the result. They relate the calculated values (result) to the values of the previous layer (priorDstValue), storing the result in the destination tensor (dstValue), as expressed in $dstValue = \alpha \cdot result + \beta \cdot priorDstValue$

Before using any function from the cuDNN library, the software developer must create and initialize all function tensors. First, we create the xDesc descriptor with the cudnnCreateTensorDecriptor() function, where tensorDesc points to the portion of memory in which we allocated the tensor.

```
cudnnCreateTensorDescriptor(cudnnTensorDescriptor_t *tensorDesc)
```

Then, we must tailor the tensor via the cudnnSetTensor4dDescriptor() function where tensorDesc references the previously created tensor and the format indicates the layout with which we organised the data in memory. In this doctoral thesis, we adopted the NCHW format for HS images.

We then specify the data type. In this doctoral thesis, we adopted the float for the HS images and the dimensions of the 4-D tensor: number of batches (n), number of channels (c), height (h) and width (w).

Algorithm 3. cuDNN set-tensor descriptor

```
1. cudnnSetTensor4dDescriptor(
2.   cudnnTensorDescriptor_t tensorDesc,
3.   cudnnTensorFormat_t format,
4.   cudnnDataType_t dataType,
5.   int n,
6.   int c,
7.   int h,
8.   int w)
```

We employ the same functions to create and configure the output tensor yDesc. Similarly, the filter descriptor wDesc is generated with the function cudnnCreateFilterDescriptor() and arranged with the function cudnnSetFilter4dDescriptor().

On the other hand, we create the convolution descriptor convDesc through the function cudnnCreateConvolutionDescriptor(). The prototype of its configuration function cudnnSetConvolutionNdDescriptor() is as follows.

Algorithm 4. cuDNN set-convolution descriptor

```
1. cudnnSetConvolutionNdDescriptor(
2.   cudnnConvolutionDescriptor_t convDesc,
3.   int arrayLength,
4.   const int padA[],
5.   const int filterStrideA[],
6.   const int dilationA[],
7.   cudnnConvolutionMode_t mode,
8.   cudnnDataType_t datatype)
```

The function receives as input:

- The reference to the previously created convolution descriptor convDesc
- The size of the convolution
- The two-dimensional vectors containing the padding, stride and dilatation value along the x and y axes of the image
- The convolution mode may be either CUDNN_CONVOLUTION or CUDNN_CROSS_CORRELATION
- The data type, which, for the HS images this thesis operated, is float

The variable algo, passed to the function cudnnConvolutionForward(), can take the following values, depending on the type of algorithm one decides to adopt:

- **CUDNN_CONVOLUTION_FWD_ALGO_IMPLICIT_GEMM**: expresses the convolution as a matrix product without explicitly forming the matrix containing the input tensor data
- **UDNN_CONVOLUTION_FWD_ALGO_IMPLICIT_PRECOMP_GEMM**: algorithm similar to the previous case but requires memory in the workspace to calculate specific indices in order to facilitate the calculations
- **CUDNN_CONVOLUTION_FWD_ALGO_GEMM**: expresses convolution as an explicit matrix product and requires a significant amount of memory in the workspace
- **CUDNN_CONVOLUTION_FWD_ALGO_DIRECT**: performs a direct convolution without performing the matrix product
- **CUDNN_CONVOLUTION_FWD_ALGO_FFT**: uses the Fast-Fourier Transform (FFT) approach to calculate the convolution and requires a lot of workspace memory
- **CUDNN_CONVOLUTION_FWD_ALGO_FFT_TILING**: algorithm similar to the earlier case, but partitions the input into sub-regions during calculation
- **CUDNN_CONVOLUTION_FWD_ALGO_WINOGRAD**: employs the Winograd Transform approach for convolution calculation
- **CUDNN_CONVOLUTION_FWD_ALGO_WINOGRAD_NONFUSED**: algorithm similar to the previous case but requires more memory in the workspace

On the other hand, the developer must fill the workSpaceSizeInBytes parameter of cudnnConvolutionForward() through the following function in which the input parameters are created and initialised earlier.

Algorithm 5. cuDNN get convolution forward workspace size

```
1. cudnnStatus_t cudnnGetConvolutionForwardWorkspaceSize(
2. cudnnHandle_t   handle,
3. const   cudnnTensorDescriptor_t       xDesc,
4. const   cudnnFilterDescriptor_t       wDesc,
5. const   cudnnConvolutionDescriptor_t   convDesc,
6. const   cudnnTensorDescriptor_t       yDesc,
7. cudnnConvolutionFwdAlgo_t             algo,
8. size_t                    *sizeInBytes)
```

The sizeInBytes pointer stores the output size in bytes of the workspace we will use.

At this point, we allocate the workspace with the sizes we just computed thanks to a cudaMalloc() which will run as follows so that we can also pass the actual workspace as a parameter to cudnnConvolutionForward() as required[79].

```
cudaMalloc((void**)&workSpace, workSpaceSizeInBytes);
```

The convolution outcome is a tensor whose dimensions result from the function cudnnGetConvolution2dForwardOutputDim() for 2D convolutions. It receives as input parameters the previously created convolution descriptor, the input descriptor (x) and the filter descriptor (w), and in turn, stores in the integer pointers n,c,h, and w, respectively, the dimensions of the output tensor[79].

Algorithm 6. cuDNN get convolution2d forward output dimension

```
1. cudnnStatus_t cudnnGetConvolution2dForwardOutputDim(
2.     const cudnnConvolutionDescriptor_t  convDesc,
3.     const cudnnTensorDescriptor_t       inputTensorDesc,
4.     const cudnnFilterDescriptor_t       filterDesc,
5.     int                                 *n,
6.     int                                 *c,
7.     int                                 *h,
8.     int                                 *w)
```

Consequently, we obtained the last missing parameters (n,c,h,w) that we need to allocate with a cudaMalloc() the space for the result tensor of our convolution. Finally, we can invoke cudnnConvolutionForward() using what we have created and initialised as parameters.

4.8. Test systems

As for the applications presented in the following chapters, we conduct tests on two different systems whose characteristics are listed below.

1. System 1 (TS1) comprises Intel-i9-9900X CPU, working at 3.5 GHz, 128 GB of RAM, and two 2944 CUDA-cores Nvidia RTX 2080 GPUs
2. System 2 (TS2) is equipped with an Intel i7-3770 CPU, working at 3.4 GHz, with 8 GB of RAM and connected to an Nvidia Tesla K40 GPU (Kepler architecture). This device has 2880 cores, and 12 GB of RAM and its working frequency is 875 MHz
3. System 3 (TS3) is the Nvidia Jetson Nano™ Developer Kit. It comprises a 128 CUDA-cores Maxwell GPU, a Quad-core ARM A57 running at 1.43 GHz CPU and 4 GB 64-bit LPDDR4 RAM

Chapter 5

The SARS-CoV-2 pandemic

SARS-CoV-2 caused the Covid-19 pandemic, which originated in China and abruptly scattered within Europe since February 2020 and is still challenging the world's health systems. It manifests after incubation and yields a high contagion rate. Hospitals need fast and cheap diagnostic tools to detect infected subjects. Section 2.2 reported that subjects infected by SARS-CoV-2 might present a transforming clinical situation varying from focal to multifocal interstitial pulmonary involvement that LUS may visualise through the artefacts we described in Section 2.1. The high contagion rate accounts for further intricacy because physicians must merge patient care with strict safety protocols[15]. Currently, the main diagnostic tools for detecting and isolating infected people include real-time reverse transcription-polymerase chain reactions (rRT-PCR) in nasopharyngeal swabs (NPS), and IgM-IgG combined antibody tests. However, both these tools have limitations. Both exhibit the same poor sensitivity, with a slight increase only after a specific duration following symptom manifestation. Regardless, IgM-IgG may result in false-negative results in the early phases of the infection. Covid-19 begins with mild or no symptoms and can rapidly transform, subjecting patients to highly critical conditions with possibly fatal consequences resulting from multi-organ failure[15,26,29].

First-line diagnosis of pneumonia might exploit chest X-rays (CXR) for first-aid treatment of patients exhibiting symptoms of pneumonia. Potential alternatives to CXR include computed tomography CT scans and LUS. The main conclusions from studies concerning these methodologies state that LUS and CT scans are significantly better first-line diagnostic tools than CXR, whose main drawback is poor sensitivity. Moreover, LUS is a cost-effective, radiation-free, and promising tool, but a highly skilled radiographer must perform it to achieve accurate results. Furthermore, LUS effectively performed at the bedside in approximately 13 min yields higher sensitivity than CXR[20,21,30].

In respiratory diseases, arterial blood gas (ABG) and LUS quantitative examination play a crucial role[16]. They instruct the diagnosis and disease severity stratification, allowing adequate therapy. Two commonly used indices to assess the pathogenic mechanism of respiratory failure are the PaO2 / FiO2 ratio (P/F) and the alveolar-to-arterial oxygen difference

(AaDO2)[16,80]. While clinical practice operates the P/F as a simple measure of lung dysfunction in critically ill patients to predict disease outcome, the Berlin criteria in ARDS patients reports an elevated AaDO2 accompanied by hypoxemia indicating a ventilation-perfusion mismatch or intra-pulmonary shunting[16]. Covid-19 pneumonia relates to increased shunting and altered oxygen alveolar–arteriolar barrier diffusion, which might be associated with increased AaDO2 and decreased P/F values.

The Covid-19 pandemic has resulted in renewed attention to the studies mentioned above and led medical professionals to evaluate potential answers for the abovementioned challenges and procure fast, cheap, and efficient diagnostic mechanisms. SARS-CoV-2 necessitates specific binding and strict constraints to sidestep cross-contamination, such as infected staff or medical devices, and provide patients with the highest standard of healthcare, such as transferring patients for treatments or examinations and making diagnostic tools readily available to everyone. For example, these crucial necessities made it impossible to use a stethoscope during hospital operations in infectious disease departments owing to the use of personal protective equipment. Therefore, researchers concluded that both CT scans and LUS are promising diagnostic instruments that are capable of early SARS-CoV-2 pneumonia detection and present highly correlated patterns for different disease stages[15,16,26,29]. Although the former initially served a pivotal function during the pandemic, it exhibits some weaknesses in terms of the previously stated constraints, while the latter does not. Consequently, it has been beneficial to rely upon an international standardisation of LUS exploitation, providing a medical procedure and the scoring scale Section 2.3 described.

Researchers have extensively reviewed machine and deep learning biomedical applications in this context, highlighting the challenges of using labelled datasets in medical contexts.

This chapter will focus on the statistical and AI approaches this doctoral thesis researched to counteract SARS-CoV-2. The studies address the operation of specific diagnostic measurements, also used to collect the dataset described in Section 2.11, the classification of LUS clips and assessing patients for SARS-CoV-2 positivity through blood tests. Close collaboration with Fondazione IRCCS Policlinico San Matteo's Emergency Department (ED) of Pavia enabled the investigation of the research just mentioned[15–17].

In the following lines and sections, this chapter describes the state of the art methodologies and results applied to the problems addressed in this doctoral thesis. Then, for each of the works researched in the educational path described in this thesis, the chapter contains an exploratory data analysis, a section describing the materials and methods of the study and the concluding remarks. These address the discussion of the results, conclusions, and implications that advance the field based on current knowledge and our achievements.

5.1. AI-based state-of-the-art for pandemic management

During the last decade, the number of research articles on artificial intelligence (AI) as a resource for all kinds of medical specialities highly increased, demonstrating machine learning (ML) algorithms to be successful[1]. Notably, AI-enabled support systems aid clinicians' decision-making, especially during triage operations at the hospital. Indeed, most studies aim to develop models to schedule patients according to their triage acuity level, assessing the severity of their conditions and deciding upon hospital admission[15,17]. SARS-CoV-2 studies comprise algorithms exploiting computed tomography (CT), lung ultrasound (LUS), and X-ray imaging techniques to diagnose and examine evolving Covid-19 patterns[14,61]. On the other hand, researchers exploited ML and AI-based techniques to tackle contact tracing, predicting and forecasting epidemiological measurements, and SARS-CoV-2 drug development[81,82].

During the first heavy pandemic waves, several countries limited swab testing due to the unfeasible number of them to be analysed. Research has also focused on machine learning algorithms to quickly assess patients for Covid-19 positivity and mortality ever since. The studies confirmed the feasibility of the statistical learning-based approach. Nonetheless, only 8% of the studies observed by literature reviews focused on blood test analyses. Researchers produced statistical models having different goals ranging from Covid-19 to mortality prediction, with classification results having sensitivity levels of approximately 80–89%[17,81–83].

On the other hand, literature documented AI as a good answer for overcoming the formerly stated issues concerning SARS-CoV-2 and introduces advantages, including diagnostic pace, trustworthiness, and support provision to physicians handling the emergency.

Systematic surveys on DL applications for the coronavirus exposed that studies mainly concentrated on CT scans and X-rays. Less than half of the investigations operated on transfer learning, while none considered lung engagement severity. Physicians extensively employed LUS to estimate consequences in patients admitted to the emergency department (ED) and to detect Covid-19 pneumonia in subjects who presented a negative swab. Regardless, only a few studies have researched the application of DL algorithms to LUS data. These studies focused on detecting B-lines, artefacts appearing when patients suffer from pneumonia, or binary classifications of LUS frames into Covid-19 and non-Covid-19[14,15,17].

Accordingly, only a few researchers have exploited data from trustworthy hospital sources, indicating the need for a dedicated dataset. Several authors have described the inconsistent quality of their data and the need to rely on non-validated sources as limitations of their studies.

Likewise, some researchers have worked with LUS from only one particular type of probe, thus needing more heterogeneous data to train the

neural networks, posing another limitation on the soundness of their conclusions and DL algorithm usage[25,84–86].

Only two studies have focused on DL systems to detect Covid-19 pneumonia and assess the severity of lung engagement. The former exploited a spatial transform network developed in 2015, while the latter proposed an original neural network. However, both reported poor performance at frame-level scoring for assessing the severity of lung engagement, and neither used pre-trained or state-of-the-art architectures. Furthermore, the authors of the former study proposed a novel scoring methodology for validated and researched scales evaluating lung health status, which the latter adopted as well[15].

To the best of this doctoral thesis's knowledge, there has been no investigation on assessing and ranking the lung pleural line health conditions through the application of artificially intelligent systems to LUS data obtained from Covid-19 subjects at the time of writing. Moreover, all investigations regard frame classification without addressing the entire LUS clip.

5.2. Alveolar-arterial difference and lung UltraSound to help the Covid-19 clinical decision-making

This first research aims to prove baseline AaDO2 capability, measured at ED admission, in predicting the need for oxygen support and survival expectations in patients affected by Covid-19.

Furthermore, given the recognised role of LUS in assessing Covid-19 presence, the secondary aim consists of evaluating the correlation between AaDO2 and LUS quantitative evaluation. Nonetheless, this research operated the LUS score as the sum of the twelve lung portion assessments (Section 2.2), and this explains why we will also encounter values above 3 reading the following investigation. Proving such correlation is of utmost importance, especially in patients with typical P/F values. Indeed, these patients present higher risks of undertreatment and might subsequently experience fast health conditions worsening due to unexpected clinical transition.

Healthcare practitioners have lived with these working conditions since the first pandemic waves, which elicited the need for simple prognostic indexes to better steer clinical decision-making and safe hospital discharge policies, especially in an overcrowded ED during contingency periods[16].

5.3. Materials and methods

The investigation affects the data which Section 2.11 explained. The data inclusion requirements for collection and the final analysis included:

- RT-PCR test positive result
- Written informed consent
- Lung ultrasound quantitative examination associated with a suspect of Covid-19 presence
- Complete blood count
- Assessment of renal and liver function
- Troponin I
- serum electrolytes
- C-reactive protein
- lactate dehydrogenase
- creatinine kinase
- vital signs
- symptoms
- ABG
- AaDO2, which relied on the mathematical formula in Equation 22

$$\mathbf{AaDO2} = ((\mathbf{FiO2})\,(\mathbf{Atmospheric\ pressure} - \mathbf{H2O\ pressure}) - (\mathbf{PaCO2/R})) - \mathbf{PaO2} \quad \text{Equation 22}$$

We considered the same values for all patients listed for the terms listed in Table 6. Healthcare professionals consider typical AaDO2 values according to the following formula: $2.5 + 0.21 \times age$[80].

Table 6. Shared values in Equation 22 for all patients

Atmospheric pressure	**760 mmHg**
H2O pressure	47 mmHg
Respiratory quotient (R)	0.8

The acquisition protocol described in Section 2.3 reports healthcare professionals performing bedside LUS evaluation to patients waiting for the swab results. They explored the subjects' thorax in the supine or semi-supine position, depending on the level of cooperation.

The Fondazione IRCCS Policlinico Hospital complied with the American College of Emergency Physicians' ultrasonographic guidelines. Indeed, only experienced sonographers with more than ten examinations conducted per week and at least five years of experience performed the LUS tests.

Physicians recorded the LUS clips operating Section 2.3's scores and twelve thorax windows, which this thesis investigated in subsequent research described in this chapter, to allow off-line re-evaluation.

This research evaluated the relationship between LUS score and ABG respiratory parameters on the whole dataset and in a subset of patients whose P/F values were from 300 to 400.

We expressed continuous variables through median values, whilst categorical variables were as percentages. We considered p-values less than 0.05 statistically significant.

This investigation presents the results via scatter plots, regressions, ROC curves, and $\chi 2$ analyses[57].

5.4. Analysis of the results

530 out of 820 patients admitted to ED during the observation period, reported in 2.11, had a SARS-CoV2 positive nasopharyngeal swab. Among those, 223 presented a complete LUS examination and an ABG satisfying the abovementioned requirements. Table 2 summarises the baseline features of the various clusters considered for examination.

61.9% of subjects were males, and their median age was 61, ranging from 22 to 90 years. The most frequent symptom was fever (89.7%), followed by cough (48%) and dyspnoea (46.2%).

Table 2 reports from 136 (61%) patients having at least one comorbidity: 10.3% with at least three pathologies, with hypertension the most observed (45%), followed by diabetes (14.4%), coronary artery disease (12.6%) and asthma (6.3%).

The hospital admitted 7.6% of patients to the intensive care unit (ICU), 45.3% in a general ward, whilst discharged 45.7% and 1.3% died in the ED. 23.3% of the 223 patients received higher-intensity care with continuous positive airway pressure (CPAP) or invasive ventilation (IOT). The Median overall LUS score was equal to 9, and only 16.1% of patients did not have lung involvement: 44.8% presented only vertical artefacts, and 39.1% reported vertical artefacts and consolidations.

Concerning the ABG analysis, whose median values are reported in Table 2, the reduction of P/F and pO2 values was related to the increasing severity of the clinical picture. Conversely, AaDO2 increased with worsening clinical conditions.

Figure 49 displays the relationships between AaDO2 - P/F and AaDO2 - LUS scores. The distribution of AaDO2 values in patients with P/F values ranging from 300 to 400 demonstrated increased AaDO2 values with decreasing P/F[16].

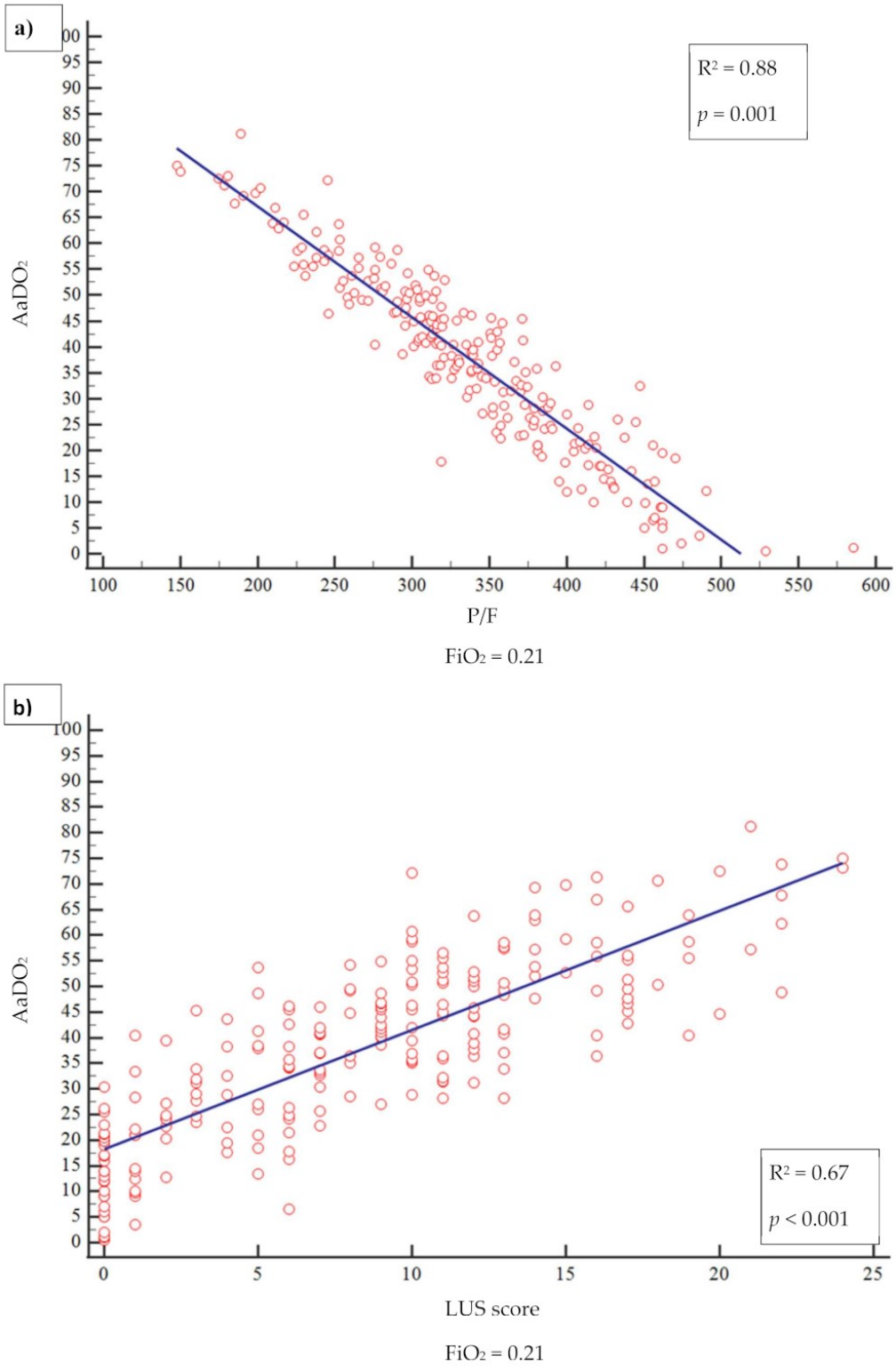


Figure 49. (a) relationship between AaDO2 and P/F and (b) between AaDO2 and LUS[16]

Before producing AI solutions for the ED, academia must focus on providing it with sufficient diagnostic measures. Accordingly, this research

analysed the AaDO2 to predict the need for an assisted high flow of oxygen and survival outcomes.

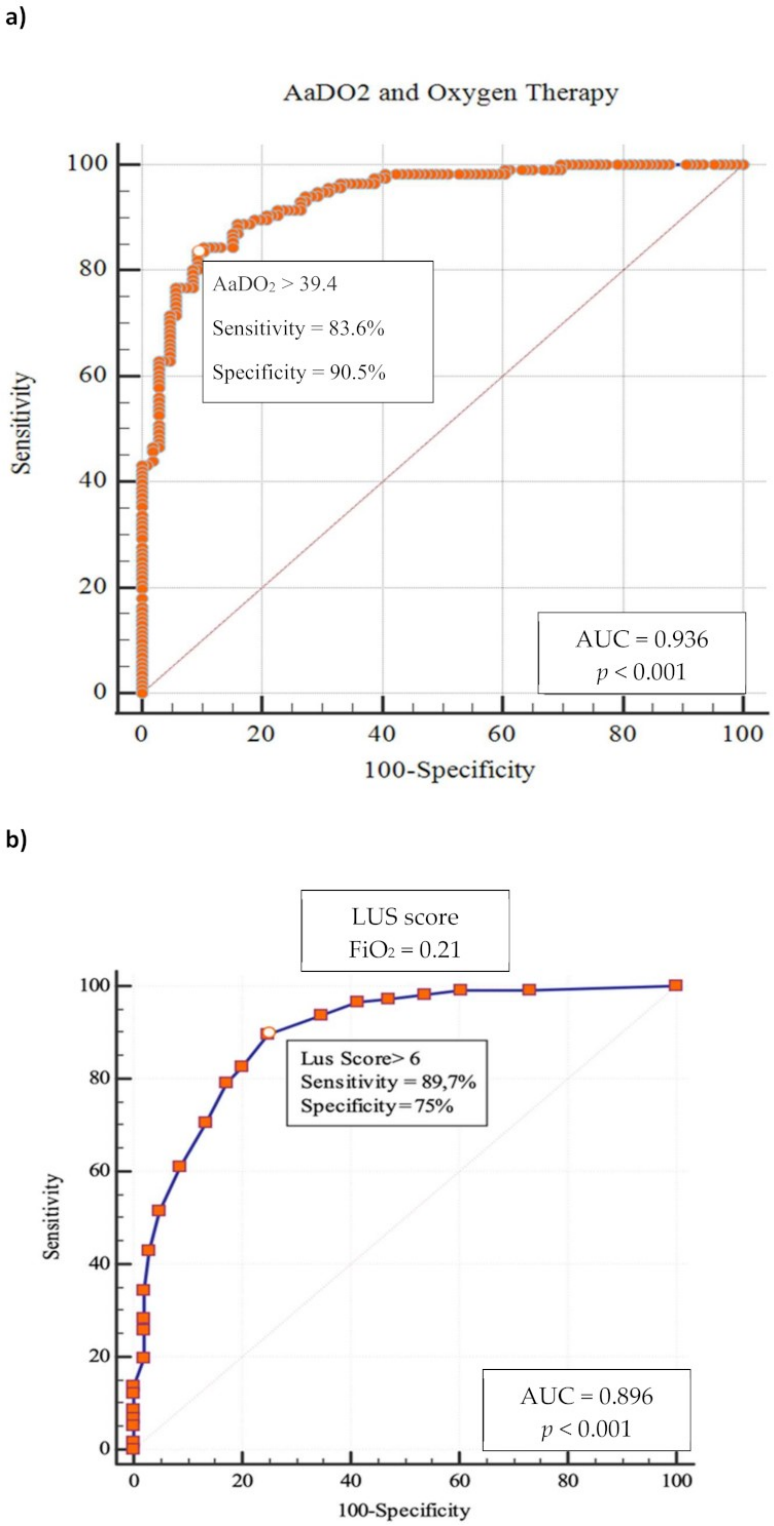


Figure 50. ROC curves in whole cohort. (**a**) $AaDO_2$ and Oxygen Therapy. (**b**) LUS score and Oxygen Therapy[16]

Figure 50 reports 83.6% sensitivity and 90.5% specificity in predicting the need for oxygen with AaDO2 higher than 39.4, whilst 46.9% sensitivity and 90.7% specificity in predicting death at 30 days for AaDO2 higher than 57.2. The AUC values were 0.936 and 0.744, respectively.

The research obtained similar results on the subgroup of patients with P/F values ranging from 300 to 400. In the first scenario, an AaDO2 value higher than 36.4 yields 78.6% sensitivity, 75.4% specificity and 0.831 AUC.

The research also estimated the subsequent need for oxygen support from LUS score overall evaluations higher than 6 with 89.7% sensitivity, 75% specificity, and 0.896 AUC[16].

5.5. Final remarks and study limitations

Based on an observational cohort of Covid-19 patients evaluated at the Fondazione IRCCS San Matteo University Hospital in Pavia (Italy), the present study shows as its main result that AaDO2 can be a valuable parameter to stratify the evolutionary risk of patients with Covid-19[16]. To the best of the research knowledge, this was the first investigation evaluating the function of AaDO2 measured at hospital admission from the ABG analysis to characterise Covid-19 patients better.

ABG testing is readily available in the emergency setting, giving crucial information about pulmonary involvement and respiratory function. AaDO2 enables a more precise evaluation of the pathophysiological basis of hypoxemia than the P/F ratio. However, the latter reached a larger audience to measure pulmonary dysfunction in critically ill patients[16,80].

Furthermore, the combined use of LUS imaging findings and their scoring and AaDO2 allows a better understanding of the underlying pathophysiological mechanism.

Hence, the present study evaluated the role of the alveolar-to-arterial oxygen difference, particularly in Covid-19 patients with P/F values ranging between 300 and 400. According to the literature, this range represents patients without significant acute lung injury. Nonetheless, this study proved the opposite. Indeed, although this subgroup of patients possessed typical P/F values, AaDO2 was higher than regular. Moreover, more than half of these patients subsequently required oxygen therapy support.

Interestingly, patients who subsequently needed oxygen support had a more severe extent of lung involvement, as assessed by LUS, than those who did not. Indeed, literature reported that patients with Covid-19 pneumonia often do not register dyspnoea, despite extreme hypoxemic values. Academia defined this clinical presentation as silent hypoxemia or happy hypoxia, with physical signs that may either overestimate or underestimate patient discomfort.

In conclusion, patients might have presented with few clinical signs and symptoms, a chest X-ray not indicating the significance of lung

involvement, and P/F still within normal limits. Therefore, it is essential to obtain elements that predict the risk of subsequent clinical worsening[16].

This first research described the importance of the data collection, which produced Section 2.11's database. Physicians relied on the analysis this section described to gather data and LUS clips, which set the stage for the research described in this chapter's subsequent sections.

Nevertheless, we should acknowledge some limitations of this study. The retrospective single-centred configuration leads to missing information and unavoidable biases in specifying and recruiting participants. Fondazione IRCCS Policlinico Hospital of Pavia gathered the data in contingency times concerning the SARS-CoV-2 pandemic, and the sample size was relatively small.

Despite these limitations, the study reflects an actual world clinical scenario in the ED during a pandemic outbreak. The promising results open the doors for further validation in future multi-centred extensive prospective studies to consolidate LUS and AaDO2 assessments[16].

5.6. Machine-learning-based Covid-19 and dyspnoea prediction systems for the emergency department

This chapter mentioned the crucial aspects of splitting a hospital's emergency department into clean and dirty areas during the SARS-CoV-2 pandemic to preclude patient-to-patient spreading. The utmost priority is engineering fast and trustworthy tools to assess and manage patients' prognoses to optimise resource distribution. Therefore, AI-enabled support strategies might aid healthcare professionals in decision-making, particularly during ED triage[17].

Given the unfeasibility of continuous swab testing and its limitations imposed by governments, academia has also concentrated on machine learning algorithms to assess patients for Covid-19 positivity and mortality. Fondazione IRCCS Policlinico San Matteo Hospital's Emergency Department (ED) of Pavia allowed operating machine learning algorithms on the clinical dataset gathered from laboratory-confirmed rRT-PCR test patients we described in the earlier and 2.11th Section of this thesis.

The main goal of the investigation we will describe was to quickly stratify patients and employ cross-contamination avoidance strategies, sidestepping comprehensive swab testing and leveraging healthcare professionals' workload[17].

We gathered patients' data according to what we described earlier. Indeed, the dataset, accurately described in Section 2.11 and investigated in the previous one, comprises comprehensive information concerning patients' respiratory failures, routine blood tests, arterial blood gas (ABG) analysis, and quantitative lung ultrasound evaluations.

On the one hand, the research proposed in the previous paragraphs explored the importance of specific clinical parameters to provide prompt and accurate methodologies for physicians engaged with the extreme working conditions settled by the SARS-CoV-2 pandemic. The research involved no AI methodology but straightforward statistical approaches. Nonetheless, it was essential to analyse the data collected and set the motivation for the present and the subsequent studies that this chapter presents.

On the other hand, this section explores machine learning approaches to automate clinical decisions and provide innovative tools that leverage and assist physicians' workload. Indeed, this thesis adopted support vector machines (SVMs) and random forest (RF) algorithms (Sections 3.7 and 3.8) to assess patients' Covid-19 positivity, operating section 2.11's dataset.

Furthermore, this thesis researched estimation procedures concerning whether a subject would need oxygen therapy, such as continuous CPAP or IOT we explored in the earlier section. Indeed, one must organise and wisely engage limited resources during contingency times.

The novelty of the designed approach, summarised in Figure 51, stands in the following passage:

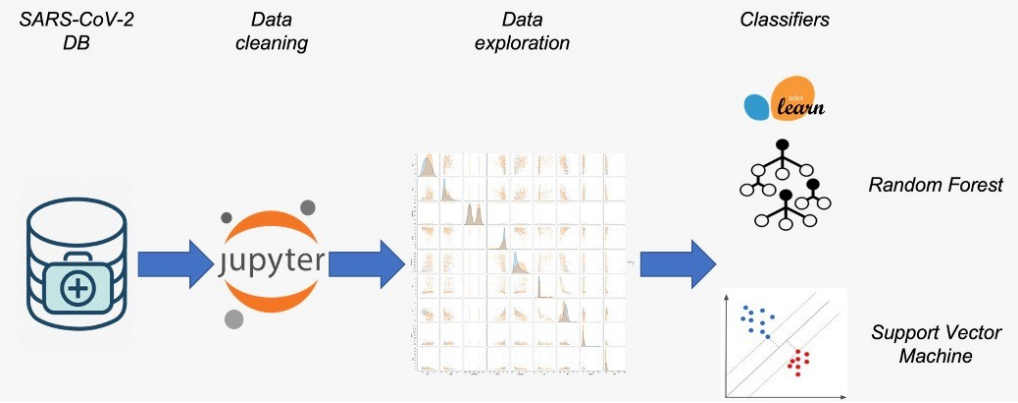


Figure 51. Summary of the machine learning approach: the data analysis workflow[17]

- A careful clinical features collection: the thesis based the classifiers on the features that physicians employed during triaging and daily clinical operations, whose importance was stressed in the earlier section
- Extensive and robust data analysis before ML clustering
- Exploitation blood tests to assess patients rather than imaging data
- Assessment of patients' need for oxygen therapy to carefully engage limited resources in contingency scenarios

- A quantitative lung involvement examination to produce robust results: studies report lung ultrasound examination as a fast, cheap, and agile tool to assess patients' lung involvement

5.7. Methodological analysis

Here, the thesis provides a detailed illustration of the data collection, cleaning processes and exploration. Furthermore, it describes the selection and the design of the machine learning methodologies to diagnose SARS-CoV-2 and predict the need for assisted ventilation.

5.8. Data cleaning and pre-processing

Fondazione IRCCS Policlinico San Matteo's ED of Pavia appointed a strict protocol during triage to analyse patients whom SARS-CoV-2 might have potentially contaminated.

We performed data cleaning and pre-processing operations to apply the ML methodologies described in Chapter 3 of this thesis and aid doctors during the pandemic. The methodology concerns translating categorical features into dummy variables, namely converting textual elements into discrete and numerical values. Moreover, it involved handling missing values, which could have been missing due to several causes.

The first motivation is machinery malfunctioning: the devices performing the tests required might not have stored specific information pleasingly[17].

The second rationale concerns the physicians' workload rate. If a subject was undoubtedly affected by SARS-CoV-2, but there was not enough time to assist other people according to the highest healthcare norms and terminate the data acquisition protocol, the personnel interrupted the procedure to respect the hard time constraints demanded by the pandemic. Unfortunately, there is no straightforward manner to patch missing entries in a database[17].

Consequently, the study modified the dataset to 443 patients, excluding those whose entries were unavailable.

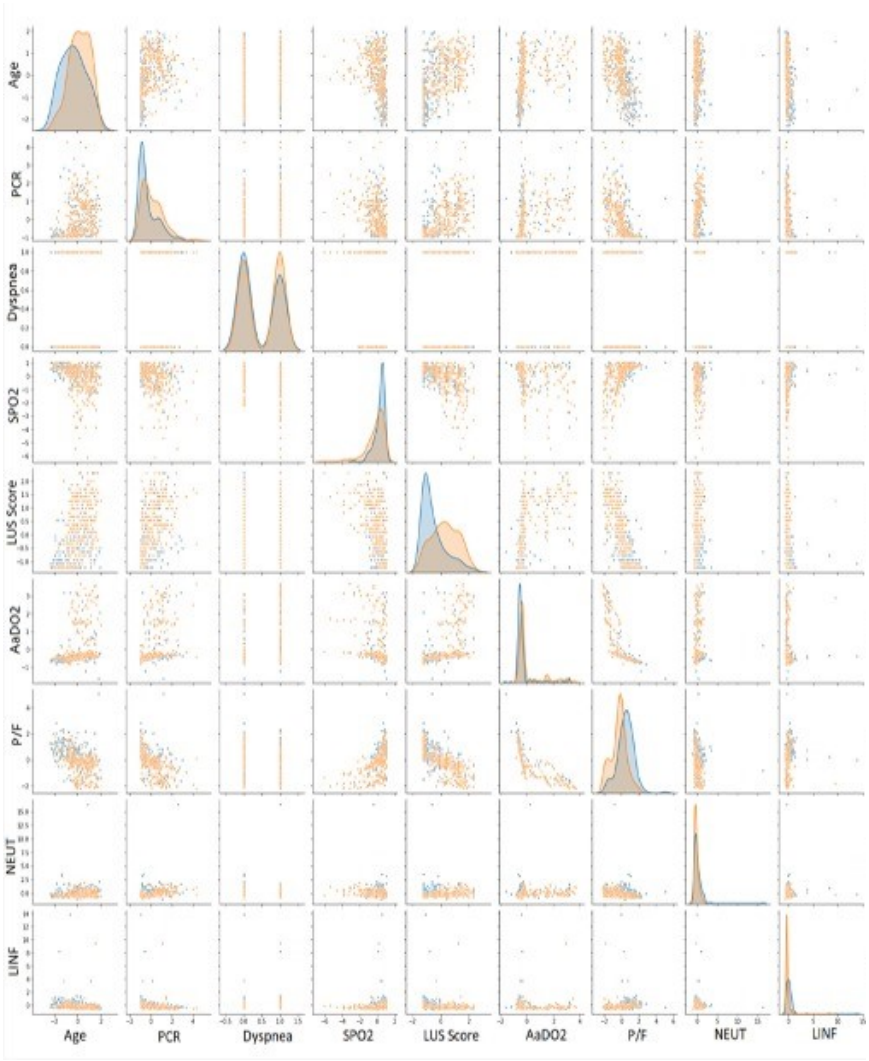
In summary, this analysis divided 90% of the data into the training set and 10% into the test group. Professionals usually recommend a 70-30% split. Nonetheless, the dataset size was insufficient to keep relevant information for training at the end of the data-cleaning process. Accordingly, this research settled for the 90-10% subdivision. Similarly, to meet reliable results free from overfitting, the investigation involved 10-fold cross-validation (Section 3.13). At each learning step, the process randomly split the training set into ten sub-groups, using the 10^{th} to validate the data, whilst the remaining four optimised the models' weights. Chapter 3 mentioned K-fold cross-validation as a standard practice used by data scientists facing dataset size problems.

5.9. Data exploration

The study occurred in the Fondazione IRCCS San Matteo's ED outline. Chapter 2 exhibited the statistical characteristics of the patients in the dataset to explain what features the research stands upon and compare the dataset to others[17]. The percentage of positive patients is 61.2%, slightly higher than the 48.4% registered in another study[17]. Concerning oxygen therapy, only 20.8% of subjects needed CPAP or invasive ventilation, whereas 79.2% of patients required either an oxygen mask, nasal cannula, or no oxygen therapy. Chapter 2 reported the correlation coefficients between each input feature and output targets to explore the characteristics and the machine learning models.

Figure 52 displays the data exploration whose application helped to extract the clinical scenery of people impacted by Covid-19. We can cluster SARS-CoV-2 positive subjects by age and comorbidities, particularly hypertension, diabetes mellitus and cardiovascular disorders. They clinically present fever, dry cough, dyspnea, increased respiratory rate, and reduced haemoglobin and oxygen saturation[17].

Furthermore, Covid-19 relates to increased factors such as lymphopenias, white blood cell count, and C-reactive protein. Similarly, as we mentioned earlier, ABG test results present alterations, such as an elevated oxygen alveolar-arterial gradient and reduced pO2, pCO2 and P/F ratio[16,17].


Age
PCR
Dyspnea
SPO2
LUS Score
AaDO2
P/F
NEUT
LINF

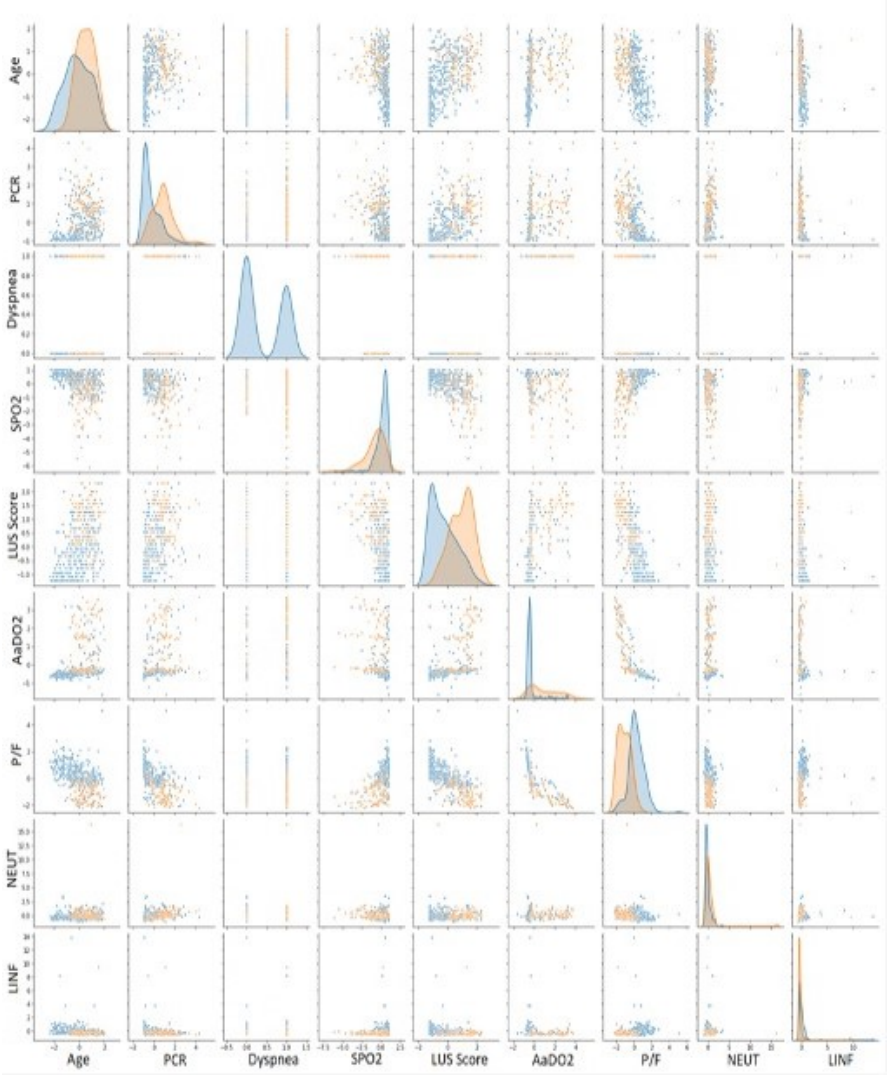


Figure 52. Scatter plot matrices for Covid-19 and oxygen therapy prediction. In the first image, orange points indicate positive patients, whereas blue points indicate negative ones. In the latter, orange points indicate patients who needed CPAP or invasive ventilation, whereas blue points indicate patients who needed either an oxygen mask, nasal cannula, or no oxygen therapy[17]

Such clustering further complies with what the two features scatter plot matrice exhibit. The investigation studied the matrices concerning coronavirus positivity and subjects who needed ventilation support. For simplicity, the plots only expose some selected patterns in Figure 52. Overall, we can notice that no scatter plot establishes a transparent partition between people concerned by Covid-19 and those with other disorders, but only a smooth transition. Nonetheless, there are some recognisable patterns. Covid-19 patients with severe conditions are older than the ones presenting healthier patterns, have higher LUS scores and have lower P/F ratios. We observed the same but a more pronounced pattern comparing

patients who needed CPAP or invasive ventilation with those who did not[17].

To decide whether the study should consider all the features to cluster the patients for both classification scenarios, it comprised principal component analysis (PCA)[1,57]. Namely, the examination involved calculating the number of input values needed to maintain the dataset's 95% statistical variance. We report that to keep 95% of the information while reducing the number of input features, the study should retain 48 principal components instead of 58 (Figure 53). The reduction could have been more significant to explain a different level of intricacy. Accordingly, the ML models do not execute any feature selection process prior to prediction. Besides, physicians demanded a quick response. Hence, the investigation retained all the input features without a further pre-processing step besides the classical ones, which comprised feature rescaling and cleaning[17].

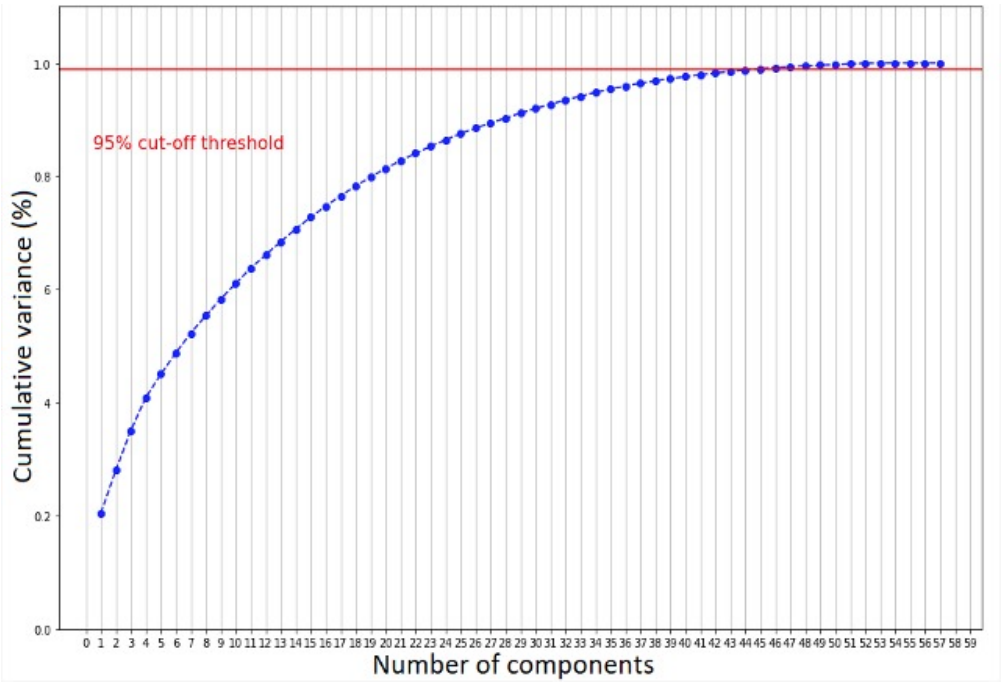


Figure 53. The number of components needed to explain 95% of statistical variance[17]

5.10. Machine learning models

We chose random forest (RF) and support vector machines (SVMs) to advance the state-of-the-art results. The exploration process we reported in the previous sections encouraged this choice.

This section introduces the RF parameters, explained in Section 3.7, in Table 7, listed in the same order as in the Python Scikit-Learn library,

respectively: n_estimators, max-depth, min_samples_split, max_features, min_samples_leaf and bootstrap.

Table 7. RF and SVM hyperparameters for each classification task

Model	Hyperparameter	Covid-19 Prediction	Dyspnea Prediction
RF.	n_estimators	550	500
	max_depth	2	2
	min_samples_split	1	1
	max_features	50	None
	min_samples_leaf	Auto	Auto
	bootstrap	True	True
SVM	C	1	1
	γ	0.01	0.01
	kernel	RBF	Sigmoid

Similarly, this section introduces the best-identified SVM hyperparameters in the same Table.

Both models adopted the hyperparameter tuning procedures explained in Chapter 3. The hyperparameter tuning procedures lean upon pseudo-random number generation. Hence, the investigation set the random seed on 19 to make experiments reproducible. Hence, we could examine the improvements derived from tuning the hyperparameters.

This research operated the first test system described in Chapter 4. The research relied on Python code and the latest version of the Scikit-Learn library to attain the classification goals.

5.11. Analysis of the results and overall discussion

The literature highlighted AI-based medical instruments' significance and function in aiding physicians and to engage limited resources[17,81,83]. Engineers are designing methodologies to determine biomarkers and process signals, innovating how we address clinical tools.

Here, we propose two diagnostic tools: Covid-19 detection and oxygen therapy need estimation due to lung involvement. The ML models selected for the two classification assignments steadily seized convergence throughout optimisation concerning the hyperparameters displayed in Table 7. Results assessment evaluated standard metrics such as accuracy, precision, recall, and the F1-score (Section 3.13). The first metric informs the reader about how good we are at diagnosing the absence of SARS-CoV-2. The latter is the degree of accuracy over an unbalanced dataset, measured by precision and recall. Indeed, the consequences of incorrectly diagnosing a patient as healthy are the inappropriate lack of treatments and cross-contamination among subjects presenting other pathologies.

Considering both clustering scenarios, Figure 54 and Figure 55 describe AUC levels exceeding 93%. At the same time, the investigation reports 96% recall when considering Covid-19 detection. Furthermore, it reports an overall F1-score of 92% for the first task and 83% for the second one, and precision is continuously above 80% (Table 8). These results are particularly worthy of notice when compared to the rRT-PCR test. Indeed, the nasopharyngeal swab attains 73.3% sensitivity (95% CI 68.1–78.0%).

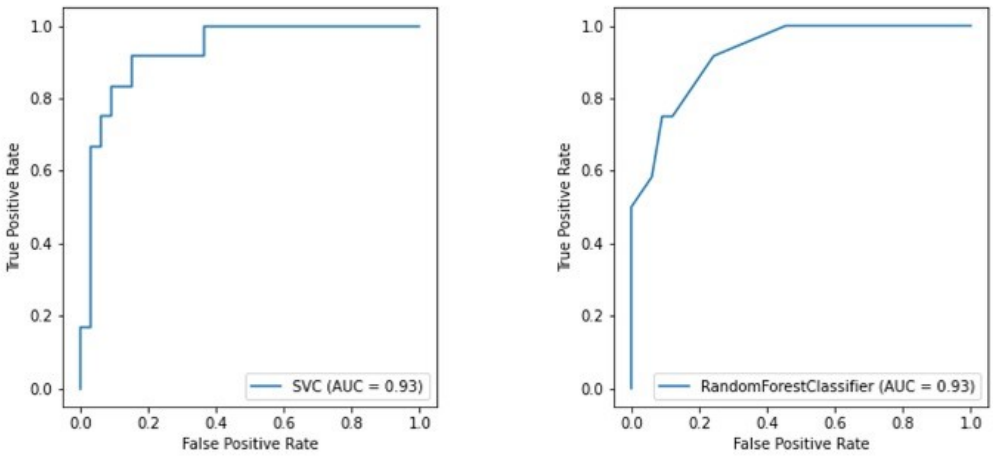


Figure 54. ROC curve of the SVM model for oxygen therapy classification (on the left) and the ROC curve of the RF model for oxygen therapy classification (on the right)[17]

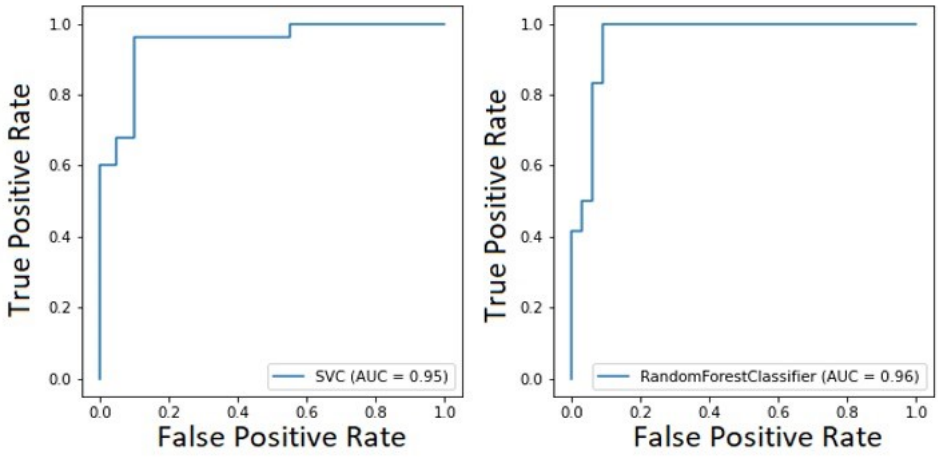


Figure 55. ROC curve of the SVM model for Covid-19 classification (on the left) and ROC curve of the RF model for Covid-19 classification (on the right)[17]

No metric exists to decide if the ED will require additional resources. Consequently, this investigation provided a valuable tool to wisely engage

the hospital's limited tools by predicting whether the considered patient will need high-intensive ventilation (i.e., CPAP or IOT – Chapter 2).

Table 8. Test set classification results[17]

Classification Task	**Model**	**AUC**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Covid-19	RF	93.0%	91.0%	89.0%	96.0%	92.0%
	SVM	95.0%	91.0%	89.0%	96.0%	92.0%
Oxygen Therapy	RF	96.0%	91.0%	83.0%	83.0%	83.0%
	SVM	93.0%	87.0%	80.0%	67.0%	73.0%

Physicians usually determine by looking at the patients' continuous vital signs, whilst this research predicted forthcoming needs concerning a specific time, namely when the patient arrives at the ED. This process implies that the considered scenario could transform abruptly. The research reports 83% F1-score and ROC-AUC values above 90% (Figure 54).

Table 8 yields this thesis' classification performance which we compare with other studies reported in Table 9[87–90]. This thesis improved the state of the art while considering a more significant number of features and also handled a smaller and particularly unbalanced dataset. Concerning Covid-19 detection, it reached 96% of recall, while others could exceed 90% only using a three-way model. Namely, a model abstains from prediction when the confidence score is below 75%[87–90].

Other researchers reached 95.9% sensitivity with 41.7% specificity. Their model represents a valuable screening tool to rule out Covid-19 infection. Nevertheless, low specificity is dangerous in the presence of infectious diseases. Certainly, identifying positive patients and isolating them is more important than ruling out negative ones[87–90].

Table 9. State-of-the-art classification results[87–90]

	Models	Features	Patients	AUC.	Accuracy	Specificity	Recall	F1 Score
Cabitza et al.	Knn, …, SVM	72	1624	76.0%	78.0%	82.0%	74.0%	-
Goodman-Meza et al.	ANN, …, XGBoost	12	1455	91.0%	-	64.0%	93.0%	-
Plante et al.	XGBoost	15	192779	91,0%	-	42.0%	96.0%	-

On the one hand, the dataset we described in Chapter 2 contains features representing the doctors' daily clinical scenario. Namely, the thesis adopted the quantitative LUS examination together with blood and ABG tests, and this process allowed the robust classification performance obtained. On the other hand, the dataset is smaller than the others. Regardless, this research

managed to handle both its size and class imbalances, and the LUS examination requires trained personnel to be performed[17].

Finally, the investigation also comprised the graphical user interface (GUI) shown in Figure 56, targeting assistance to the medical personnel at the emergency department. The GUI presents five sections for each data group: anamnesis, vital signs, blood gas analyses (BGA), blood tests and LUS score. Completing them with the patient's data, we obtain the probability of being Covid-19-positive according to our ML model.

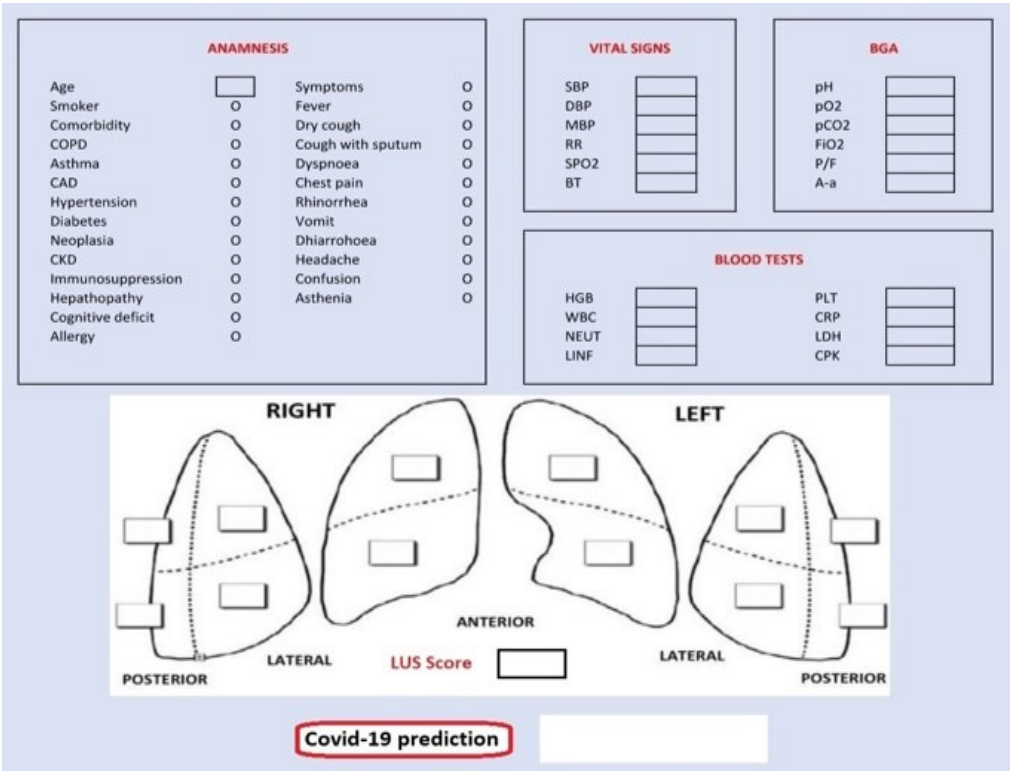


Figure 56. Graphical user interface (GUI)

5.12. Final remarks

Concerning the data collected from the routine hospital operations between 1 March and 30 June 2020, the research proved the feasibility of developing reliable algorithms to diagnose SARS-CoV-2 with high classification performance. The research we examined in Section 5.2 enabled this investigation as well.

Furthermore, in addition to what other studies had already reported, it demonstrated how to estimate dangerous dyspneic scenarios. Namely, whether the subjects at the ED need CPAP or invasive aided ventilation, and this prediction is noteworthy to handle resources in contingency times. The close and stable collaboration with the IRCCS Policlinico San Matteo's ED of Pavia granted highly reliable clinical data for the study. It made it

possible to develop two artificially intelligent systems, one of which the personnel tested as a supporting decision-making device in a real-world clinical scenario after we equipped it with a GUI.

5.13. Deep learning and Lung UltraSound for Covid-19 pneumonia detection and severity classification

This investigation proposes an innovative artificial intelligence (AI) system based on pre-trained and state-of-the-art residual convolutional neural networks (Section 3.16) to detect SARS-CoV-2 pneumonia patterns in LUS frames and classify the severity of lung engagement. It improved on previously presented results by extensively tuning the architecture's hyperparameters. The close collaboration with Pavia's University San Matteo Hospital assessed the work quality. The Hospital's Ethics Committee granted access to LUS data from different probes obtained by several physicians during the pandemic. The personnel evaluated the clips using two assessment scales (Chapter 2). This study modified the one already established in the literature[31] by adding information regarding the lung's pleural line health condition, which helps distinguish cardiogenic from non-cardiogenic causes of B-lines[15,20,23].

The developed AI-enabled assistant can operate both in emergency contexts and in-home monitoring of patients. Additionally, it can help detect patients with apparent Covid-19 symptoms whose RT-PCR or IgM-IgG blood tests were negative. These AI methods can overcome challenges, such as inadequate available RT-PCR tests, high costs, and waiting time for test outcomes.

5.14. LUS score, frames collection, ResNets and overall performance evaluation

This section provides an in-depth description of the research settings concerning the theoretical aspects mentioned in Chapter 2 and 3. In particular, we focus on data augmentation, transfer learning, training options, and the hyperparameters used to train and fine-tune deep networks.

Chapter 2 already illustrated the employed ranking scales to better highlight the results' trustworthiness and the deep architecture's proficiency in detecting Covid-19 pneumonia patterns. Regardless, other studies used a different one. Hence, Table 10 contains their comparison, revealing their differences. Doing so demonstrates the implications of the deep residual networks (Section 3.16). Furthermore, It highlights that manipulating and extending a different scoring method[31] contributed to outperforming the state-of-the-art.

Table 10. Scoring comparison Soldati et al. (2020) and S. Mongodi et al. Modified Score[15]

Severity Score	Soldati et al.[84,91]	Modified Score
Score 0	A-lines	A-lines with at most two B-lines
*Score 0**	*Not defined*	A-lines, and at most two B-lines, with a slightly irregular pleural line
Score 1	An irregular or damaged pleural line along with visible vertical artefacts	Artefacts occupy at most 50% of the pleura
*Score 1**	*Not defined*	Artefacts occupy at most 50% of the pleura and present a damaged pleural line
Score 2	Broken pleural line with either small or broad consolidated areas with wide vertical artefacts below (white lung)	Artefacts occupy more than 50% of the pleura, while consolidated areas may be visible
*Score 2**	*Not defined*	Artefacts occupy more than 50% of the pleura, while consolidated areas may be visible. The pleura is either damaged or irregular
Score 3	Dense and broadly visible white lung with or without larger consolidations	Tissue-like pattern

5.15. Data collection and annotation

This research has also profited from the data collection process that occurred in March 2020 at the Fondazione IRCCS San Matteo Hospital's ED of Pavia, described in detail in Chapter 2 and the two experimentations mentioned earlier in the text. In particular, here we focus on the LUS clips data collection to assess the health conditions of those who contracted Covid-19. The medical personnel operated the ultrasound machine described in Chapter 2's Section 2.3, equipped with both convex and linear probes. They standardised the medical practice and conducted longitudinal and transversal examinations to analyse the pleural length comprehensively, disabling all harmonics and artefact-erasing software.

Despite delivering a negative RT-PCR test, subjects displaying lung involvement have a high chance of being affected by Covid-19. Doctors are used to discriminating dubious from healthy patients observing a triaging approach that comprises LUS examination.

Hereafter, we will use the terms clip, frames, and images as reported in Chapter 2's Section 2.3. The proposed definitions produce continuity regarding observations in similar studies.

The healthcare professionals collected and assigned all 12 clips for each patient with the standardised LUS scores (Table 10). The data comprises 450 patients whose clinical information is presented in Table 2, treated in Pavia, assembling 5400 clips[15].

Physicians at Fondazione IRCCS Policlinico San Matteo ED manually selected all patient's clips, assessed each clip's quality, and assigned a score. They reviewed each clip to assign a score and verify that SARS-CoV-2 pneumonia patterns, described in Section 2.3, were present. Accordingly, this research aims at frame scoring and not at an end-to-end clip classification. Therefore, they manually selected the ones containing such patterns among the many frames belonging to a clip. Other frames might be related either to a healthy lung's portion or noisy and blurred due to incorrect probe movements or respiration-induced dynamic motions. The personnel investigated an extracted clip and appointed frames in which SARS-CoV-2 patterns were visible in a blinded and random process to reject the hypothesis of biased results. The higher the score assigned, the fewer frames are available to classify a clip. For instance, a patient assigned a score of 1 might have only a few frames containing B-lines. Because DL architectures must optimise to detect and classify pneumonia patterns, doctors must identify and collect such patterns. The number of frames selected is different for each clip. The blind selection process avoids retrieving all clips from a patient with the same pattern in most lung portions while discarding clips exhibiting other manifestations[15]. Therefore, the number of patients from whom this research clipped the frames and the number of images used for each subject are unknown. Even though the data collection had been standardized[23], it occurred during contingency times, causing not all subjects to undergo 12 assessments. Some might have received fewer than others because the severe lung engagement was visible in the early phases of the procedure.

The complete annotation and collection procedure lasted longer than one month, resulting in 676 assembled clips based on 5400 starting clips. As physicians performed LUS investigations employing different probes with slightly different settings, clips were of different sizes in terms of pixels. Therefore, we resized all the clips, so each frame sized 224 × 224, which complies with the DL architectures' input[15].

The medical personnel had to meet the demanding and urgent pandemic requirements continuously, and the process mentioned above was demanding and time-consuming. Hence, the research considered the collection and labelling process completed when the DL architectures began yielding satisfying results for the validation and test sets, as described below, and the dataset was said to be well-balanced in terms of per-class appearance.

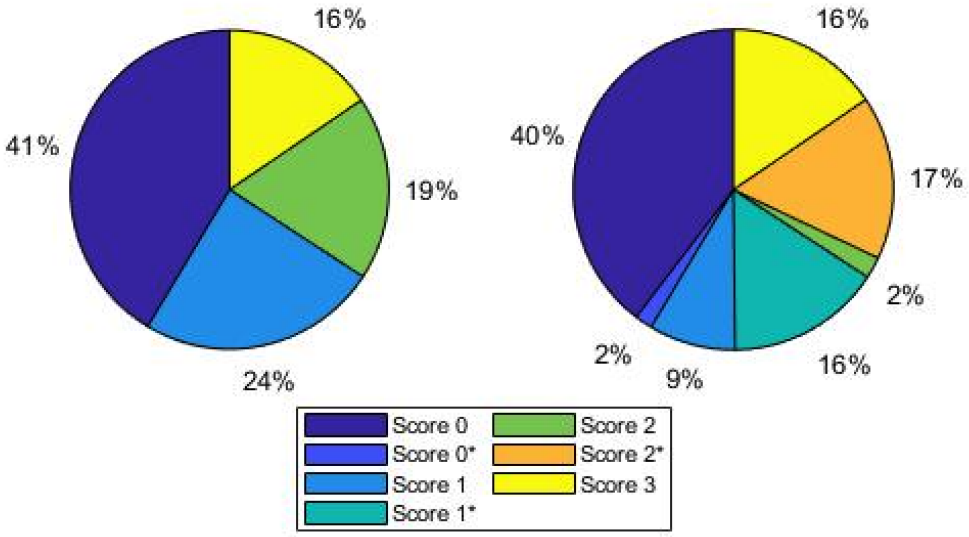


Figure 57. Percentage distribution of frames for each classification task. Left: four class scenario; right: seven class scenario. The percentage of images assigned to each score for both diagnostic tasks is depicted; pleural line involvement is highly likely and more severe when a frame is assigned a high score[15]

Hence, starting with a smaller set of collected frames, the final setting comprised 2908 frames to train the CNNs from among more than 60000 frames. Figure 57 shows the percentage of images assigned to each score for both diagnostic tasks. The pleural line engagement is highly probable in SARS-CoV-2 and more severe when an image is assigned a high score. It explains why most frames that belonged to the group with Score 2 belong to the group with Score 2*, while the same is not valid for lower values[15].

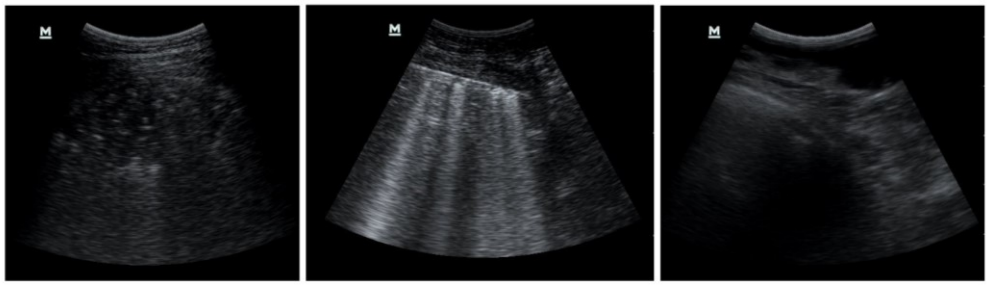


Figure 58. Examples of selected and rejected frames. We retained only the image in the middle of the figure[15]

Figure 58 reports instances of the appointed and dumped frames. The first two represent a score of 3 and 2, but we refused the latter due to noise from probe motions during the bedside investigation. This methodology is

compulsory and time-consuming, as the first and third instances may seem significantly comparable to an inexperienced eye. The same is not valid for a neural network tested on the third and noisy frame. Because we did not assign any label to the discarded frames, the network would try to categorise them as belonging to one of the supposed groups. Nonetheless, it would result in a nearly slipshod scoring and is beyond the scope of this doctoral thesis, which aims to recognise and classify Covid-19 patterns in LUS frames[15].

Finally, we randomly split the data into training (75%), validation (15%), and test (10%) sets, adopting these percentages under standard DL methodologies and maintaining the training set size as small as possible to avoid overfitting problems.

Furthermore, the research employed data augmentation techniques, as explained in Section 3.12. Finally, the collection comprises 17448, 436, and 291 images for the training, validation, and test sets.

5.16. Residual architectures and training settings

In this study, we adopted the deep residual networks described in Section 3.16. The literature reported operating verified architectures as a rational strategy for beginning AI model development from scratch[66]. Notably, this research appointed the two residual networks with 18 and 50 layers each and structured them as declared in the original paper[67]. In addition, it extensively exploited transfer learning (Section 3.15) to enhance the classification outcomes by manipulating features belonging to pre-trained networks. The literature reported this methodology to improve Covid-19 detection[61,92].

Consequently, this investigation chose ResNet-18 and ResNet-50 architectures, which had already experienced optimisation on the ImageNet dataset[58]. Regardless, the models encountered a few modifications in the last fully connected layers because these had as many neurons as the number of classes to detect. The classification problem to be solved involves detecting the lung patterns described in Section 2.3. This research conceived four different architectures, which are the two ResNets solving two queries: the first constitutes four categories, whereas the second seven, obtained by widening the first scale, delivering information regarding the pleural line integrity.

Table 11. Training Options and Hyperparameters[15]

Options and Hyper-parameters	**Four Classes**		**Seven Classes**	
	ResNet-18	ResNet-50	ResNet-18	ResNet-50
Initial Learning Rate	0.0005	0.0001	0.0001	0.0001
Learning Rate's Drop Factor	0.05	0.05	0.05	0.05
Learning Rate's Drop Period (Epochs)	2	3	3	3
Batch Size	128	64	128	64
L2 – Regularisation	0.4	0.75	0.3	0.3
Epochs	15	12	15	12
Environment	Multi-GPU	Multi-GPU	Multi-GPU	Multi-GPU
Optimiser	Adam	Adam	Adam	Adam
Loss Function	Cross-Entropy	Cross-Entropy	Cross-Entropy	Cross-Entropy

Table 11 contains the training options and hyperparameters to address the two different detection problems solved in this analysis. The training process relies on the pseudo-random selection processes; hence, the random seed was set for all experiments. This setting enables reproducible experiments and the detection of improvements derived from the tuning procedure.

Before describing the hyperparameter tuning, it is worth clarifying the rows we encounter in Table 11 whose names may be misleading considering the commonly encountered nomenclature in articles focusing on DL, such as L2-regularisation, number of epochs, and mini-batch. The drop factor implies that we steadily decreased the learning rate for each predetermined number of epochs in a piecewise manner - the learning rate decreases by multiplication with the dropping factor. Second, we selected Adam Gradient Descent[93]. During training, the research employed a validation set, indicating the robustness of the outcomes.

First, the investigation setting heuristically picked the initial learning rate, enabling a desirable classification performance, evaluated over both the training and validation sets. Then, it selected the learning rate's drop factor similarly, encouraging the optimal reaching of the cost function's minimum with elapsing epochs from the training start. Additionally, it sets the number of epochs, after which the learning rate decreases. Reducing it too early may lead to almost no update to the networks' weights after a few iterations. Even with deferring too much, the weights will continuously leap near the cost function's minimum while never reaching it. Upon completing these steps, the investigation concentrated on L2-regularisation, batch size, and the number of training epochs (Section 3.11). Finally, we set the squared gradient and gradient decay factors to 0.999 and 0.98, respectively. Researchers commonly adopt this default decision for Adam optimisation.

Once the tuning process ended, satisfying the classification performances for the test and validation sets, we turned the random seed off and repeated all experiments seven times to display all performance metrics

as a mean and standard deviation and to reject the hypothesis of biased results[15].

The investigation increased the training set's statistical assortment by adopting data augmentation techniques (section), which helped the networks focus on meaningful information. It applied geometric, filtering, random centre cropping, and colour transformations to the training frames. This method, proven to work when applied to Covid-19, produces effective results in DL classification tasks, significantly reducing overfitting[65,94]. Furthermore, we added salt-and-pepper white noise to enlarge the training set. Pre-trained architectures accept images of the size $224 \times 224 \times 3$. Therefore, we treated the grey-scale ultrasound frames as RGB images to avoid modifying the input layers and allow for colour augmentation. Therefore, we applied all augmentations to all training images, independent of the probe employed for the LUS investigation.

The research recursively applied the augmentations in Table 4 to the training set. Hence, it created a new set by unifying the original and transformed images iteratively, which broadened the training set exponentially.

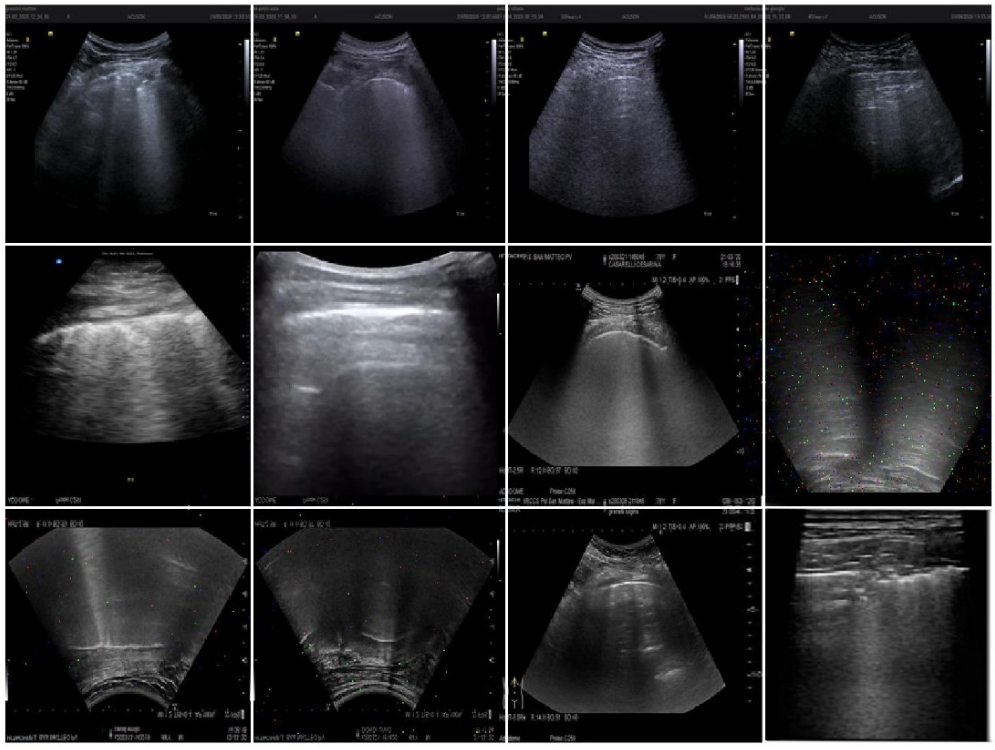


Figure 59. Augmented training set images: augmentations described in this section have been applied to the training images and are shown in this figure[15]

Figure 59 represents a set of 12 augmented examples: introducing such slight alterations into the training set allowed the CNN architectures to

develop invariance to translations, viewpoints, sizes, illumination, and noise, resulting in a more regularised training process[15,65,94]. The validation and test sets did not receive such augmentation processes to reject the hypothesis of biased results.

To further assess the classification reliability, the research operated class activation mapping (CAM) and Grad-CAM techniques[95]. These processes concern AI model explainability. When applied to computer vision applications, they emphasise the assertive parts for assigning a label by the network through a heat map. DL models can focus on points the human eye may not glimpse, thus stressing novel biomarkers in medical contexts (Section 2.12). Consequently, emphasising what networks acknowledge may aid doctors' perceptions. Notably, this research assessed whether the networks correctly highlighted either B-lines or pleural line discontinuities and all other patterns described in Section 2.3. This pioneering idea allows for comparing different prototypes to determine the best one.

Moreover, it is a cost-effective way to avoid increasing dataset preparation times by manually creating segmentations for detecting Covid-19 pneumonia boundaries. Although we intend to highlight the presence of patterns, we focus on something other than exposing their detailed shapes. The literature attempted and validated this method by applying it to Covid-19, achieving excellent results[84,91].

This research operated the first test system we described in Section 4.8.

5.17. Evaluating performance

When handling medical data, it is vital to reduce the number of false negatives to the maximum extent possible, particularly when treating an infectious disease such as Covid-19. Incorrectly diagnosing patients as Covid-19 negative introduces false negatives, causing improper care and lack of necessary treatment (i.e., cross-contamination among subjects who may have additional pathologies) and incorrect medications that may harm an infected person. Hence, this research measured the networks classification performance using the validation and test sets. It investigated accuracy, precision, recall, F1-score, and ROC-AUC (Section 3.13).

Considering the importance of reducing false negatives in medical contexts, professionals particularly consider recall, also known as sensitivity. This parameter indicates the performance of evaluating a frame as not containing Covid-19 pneumonia patterns and belonging to either of the classes considered or not representative of a healthy lung. Regardless, precision describes the classification performance in detecting the considered patterns. Consequently, we regard the F1-score as a function of the two former metrics. In summary, investigations must consider recall and F1-score to minimise the false negatives while maintaining high precision[15].

5.18. Results and discussion

The residual architectures steadily approached optimisation convergence based on the hyperparameters and training options in Table 11. This research presents the metrics discussed in the earlier section regarding the average over the number of classes operated for each classification scenario in Table 10. The training process involved stochastically splitting the data into training, validation, and test sets. At the end of each epoch, the investigation exploited the validation set to assess the models' accuracies and losses. On the training process completion, the research evaluated the metrics mentioned above for the training, test, and validation sets. It considered the network weights at the end of each training, regardless of the number of epochs selected for optimisation. Notably, we should have double-checked a particular epoch exhibiting promising performances with the validation set during optimisation. However, all training series approached convergence steadily when each network's number of epochs elapsed. We repeated the process seven times for each classification scenario to reject the hypothesis of biased results. Consequently, Table 12 contains the evaluation metrics through mean and standard deviation.

Table 12. Classification Performance Results for Test and Validation Sets: Accuracy, Precision, Recall, F1-Score and ROC-AUC

Metric $\boldsymbol{\mu \pm 2\sigma}$ **%**	**Four Classes**		**Seven Classes**	
	ResNet-18	ResNet-50	ResNet-18	ResNet-50
Training Accuracy	96.70 ± 0.01	98.32 ± 0.02	96.76 ± 0.01	98.72 ± 0.01
Training Precision	96.27 ± 0.08	96.65 ± 0.20	96.82 ± 0.07	97.57 ± 0.12
Training Recall	96.09 ± 0.07	97.23 ± 0.15	96.17 ± 0.08	98.62 ± 0.05
Training F1-Score	96.19 ± 0.07	98.27 ± 0.04	95.43 ± 0.06	99.22 ± 0.02
Training ROC-AUC	99.70 ± 0.01	99.95 ± 0.01	99.76 ± 0.01	99.97 ± 0.01
Test Accuracy	97.64 ± 1.79	98.43 ± 1.38	99.33 ± 0.59	99.72 ± 0.26
Test Precision	97.47 ± 1.99	98.59 ± 1.36	99.50 ± 0.43	99.41 ± 0.53
Test Recall	97.36 ± 1.81	98.23 ± 1.44	98.51 ± 1.29	98.93 ± 0.98
Test F1-Score	97.37 ± 1.92	98.45 ± 1.51	98.45 ± 1.49	98.94 ± 0.81
Test ROC-AUC	97.72 ± 0.63	99.91 ± 0.07	99.94 ± 0.02	99.93 ± 0.03
Test Accuracy	97.64 ± 1.79	98.43 ± 1.38	99.33 ± 0.59	99.72 ± 0.26
Validation Accuracy	97.18 ± 1.40	97.93 ± 1.20	99.37 ± 0.60	97.73 ± 1.46
Validation Precision	96.70 ± 1.80	97.82 ± 1.60	98.52 ± 1.40	94.71 ± 3.20
Validation Recall	96.95 ± 1.61	97.52 ± 1.21	98.44 ± 1.41	94.16 ± 0.74
Validation F1-Score	96.76 ± 1.82	97.66 ± 1.41	98.13 ± 1.80	93.73 ± 4.41
Validation ROC-AUC	99.78 ± 0.20	99.81 ± 0.18	99.95 ± 0.03	99.78 ± 0.20

Furthermore, the investigation settings tuned each hyperparameter to convey recall and F1-score levels exceeding 90%, indicating a high and reliable balance over the precision and recall. It resulted in both networks behaving remarkably well in each scenario and with excellent results achieved by ResNet-50. In addition, the investigation setting meets recall levels of over 97% on average, thereby verifying the soundness of the classification performance. In summary, we highlight the reliability and validity of the results in Table 12 regarding the collected measurements[15].

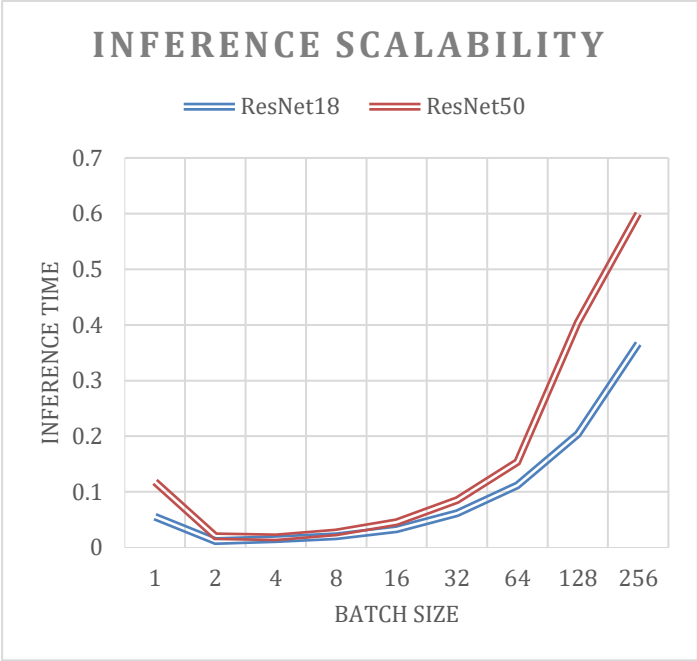


Figure 60. Inference scalability: processing times [s] according to batch size[15]

Furthermore, this study assessed network scalability during inference. Experiments comprised batch sizes ranging from 1 to 256 (i.e., each network classified between 1 and 256 images for the inference process). As expected, the inference times increased with the batch size (Figure 60). The only exception refers to the inference of a single image, which was more significant than others up to a batch size of 64. The reason is memory organisation: it is possible to group multiple images into a single tensor and adopt efficient computational routines to perform the inference. To recap, the inference times of ResNet-50 were greater than those of ResNet-18. As explained previously, ResNet-50 has a deeper and more complex structure than ResNet-18.

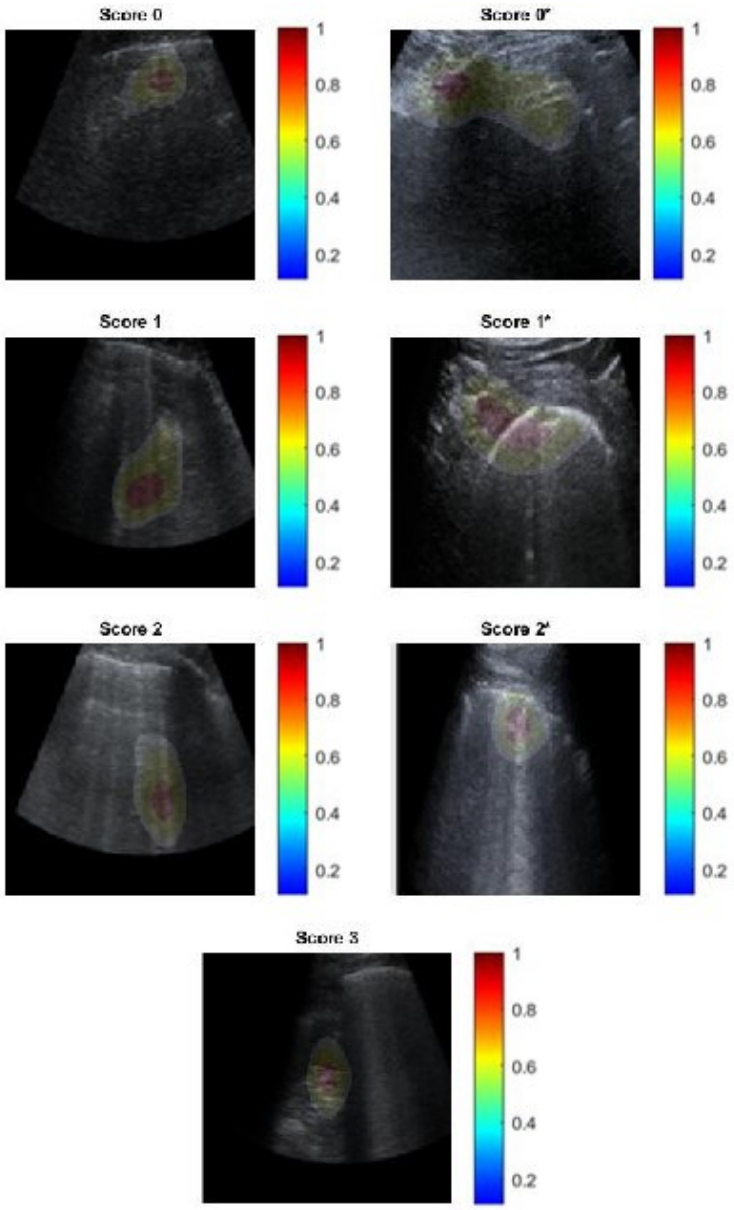


Figure 61. ResNet-50 Class Activation Mapping, seven class scenario: both severity scoring, B-lines and pleural line consolidations and irregularities are correctly highlighted along with tissue-like patterns for Score 3[15]

Furthermore, the research settings operated CAM and Grad-CAM methodologies for each experiment. The physicians evaluated whether the ResNets correctly highlighted B-lines, pleural line irregularities, or other patterns examined in Section 2.3 when ranking a frame, which is the procedure physicians usually operate to assess patients' health conditions. Figure 61 illustrates the behaviour of ResNet-50 in a scenario for which we also evaluated the pleural line. For simplicity and integrity, we present only the CAM and not the Grad-CAM assessments: Figure 61 contains the results starting from the lowest score, indicating that the considered subject

is healthy, and approaching the highest score, suggesting that we should urgently treat the patient. The residual architecture correctly and precisely outlines all patterns, namely A and B lines, small or broad consolidations, and damage to the pleural line. When assuming fewer classes, the statistical models do not evaluate the pleural line but use it to assess a subject's healthiness, specifically when analysing Score 0 instances and the reverb contained in its A-lines[15].

We have described the existing state-of-the-art studies in the introductory section of this chapter, highlighting their strengths and weaknesses. Nevertheless, this research compared its results mainly to three recent studies on applying DL methodologies to LUS data to diagnose Covid-19 pneumonia and evaluate the severity of lung engagement[15]. The first study[84,91] concerns LUS comprising a severity score for frame-level classification that met F1-score ranging from 65.1% to 71.4%. The evaluations considered the test set results or the average value over three different settings: test set, test set with transition frames dropped, and inter-doctor adjustments. Exploiting a spatial transform network developed in 2015, the authors proposed a novel scoring methodology [84,91] concerning already validated and researched scales for evaluating lung health conditions[31], which authors from the third study adopted as well. In all cases, we obtained a 27.15% performance improvement in the best average performance compared to our worst-case outline.

In contrast, the authors of the second study proposed a shallow architecture developed from scratch to address binary Covid-19 detection and severity classification. They exploited the same ranking scale adopted by the authors in the first study. Regardless, the investigation proposed in this thesis exceeded their results concerning the first clustering problem they proposed and attained a performance improvement of more than 40% in the assessment of lung engagement. Finally, the authors from the third and last study compared a wide variety of documented architectures to confirm the accuracy of transfer learning applied to Covid-19 heterogeneous data, specifically considering CT scans, ultrasound, and CXR[15]. Nonetheless, they focused on binary Covid-19 classification. Namely, they assessed whether it was present or not. Despite the favourable outcomes, the authors reported that their images were inconsistent. They based their study on non-validated data from public online repositories, and consequently, they did not assess the trustworthiness of the collected medical analyses. They accomplished excellent classification performances, which this study has been able to meet. Nonetheless, this research resulted in a more complex problem based on reliable data from Fondazione IRCCS Policlinico San Matteo. Accordingly, they obtained F1-score results ranging from 66% to 99% for the different architectures considered, which is analogous to what we researched, as listed in Table 12. Regardless, this thesis' metrics exceed 97% in all cases.

This research optimised the networks for several epochs, specifically between 12 and 15, depending on the specific experimentation. Having

fine-tuned pre-trained networks with CUDA in a multi-GPU environment, owing to the CNN libraries developed at Nvidia (Chapter 4), the process resulted in training times ranging between 17 and 89 min. Therefore, the number of epochs and the overall training time are considerably lower than those documented by different authors[14,15,84,96]. The calculations spread across two Nvidia RTX 2080, with 2994-cores each, resulting in a mini-batch processing, or more, every second.

In conclusion, previous studies on the application of DL in Covid-19 detection have presented some drawbacks. Less than half of the studies still needed to exploit transfer learning. Moreover, although authors prefer already proven architectures, they have relied upon unreliable data sources of poor quality without an appraisal by a competent doctor. In addition, only a few studies have exploited LUS for diagnosing patients with Covid-19. Only the three studies assessed illness severity or exploited transfer learning. The first two analyses concentrated on manipulating a novel scoring methodology without employing transfer learning: they assessed the lung health conditions and attempted to apply computer vision networks with minor tweaks to classify short clip pieces. Furthermore, the second work did not use newly collected LUS data from subjects who contracted Covid-19 but rather gathered clips retained at the Yale-New Haven Hospital since 2012.

Here, this thesis proposed a straightforward yet practical process to address the application of DL to LUS data and Covid-19 assessment, operating already documented and pre-trained architectures in two configurations. In addition, it adopted an existing and validated ranking scale, which we extended to structure the labels to be detected hierarchically. It helps distinguish the cardiogenic from non-cardiogenic motivations of B-lines, and the prompt detection of ARDS pneumonia symptoms facilitates the timely treatment of patients. To the best of this research knowledge, a comprehensive scoring methodology has yet to be proposed for assessing the pleural line together with existing patterns, as this research did. Likewise, this research accepted the challenges faced by our colleagues regarding having several physicians perform the LUS examinations. Consequently, the hospital personnel involved in this investigation further reexamined the clips. We concentrated on data augmentation and hyperparameter tuning to exploit the advantages of transfer learning and obtain the results presented in this chapter.

5.19. LUS frame assessment: ending remarks

This third SARS-CoV-2-related research engineered a highly reliable diagnostic instrument to satisfy exhausted medical personnel's growing request for cheap and trustworthy detection systems. With close collaboration with Fondazione IRCCS Policlinico San Matteo's ED, the investigation leaned on validated LUS data.

The research comprised modern DL methodologies, data augmentation processes, and transfer learning to grade people's lungs operating documented scoring scales[31], which the investigation extended with pleural line information. The investigation relieved the severe drawbacks of data heterogeneity (tolerable sensitivity causing lack of treatment for patients and cross-contamination) and enhanced currently accessible state-of-the-art[15] in Covid-19 detection employing LUS data.

This study provides a strategy for sidestepping the dataset problems debated by other authors[84] about the ranking inconsistencies between ultrasounds due to different doctors examining different lungs at the same disease stage. Notably, the Fondazione IRCCS Policlinico San Matteo ED inspected every test to homogeneously appoint lungs of the same disease stage with the same score.

Ultrasound requires substantial expertise to reach diagnostic reliability – high sensitivity and overall accuracy. This research developed a DL-based system to automatically detect Covid-19 pneumonitis marks in LUS frames and rank them concerning two standardised scales with innovative, reliable, and revolutionary results.

5.20. Main contributions summary

Here, the thesis proposes a list of main contributions deriving from the pieces of study described in the earlier sections.

- *Alveolar-arterial difference and lung UltraSound to help the SARS-CoV-2 clinical decision-making*
    - **Addressed problem**: SARS-COV-2 patients often require prompt diagnosis and risk stratification. Also, predict patients need for aided ventilation.
    - **Proposed solution**: Using A-a gradient and LUS to diagnose and stratify risk for pandemic management.
    - **Advantages**: A-a gradient and LUS can be obtained quickly and safely, provide valuable diagnostic and prognostic information, and combining them can improve diagnostic accuracy.
    - **Disadvantages**: A-a gradient may lack specificity for SARS-COV-2, LUS requires experienced operators, and small sample size limits generalizability.
    - **Main contributions**: A-a gradient and LUS can provide important information for diagnosing and risk stratifying SARS-COV-2 patients, especially in resource-limited settings. Study found the A-a gradient and LUS combination had 83.6% sensitivity and 90.5% specificity, with 90.7% positive predictive value (PPV) and 83.5% negative predictive value (NPV) in predicting the need for high flow of oxygen.

- *Machine-learning-based SARS-CoV-2 and dyspnoea prediction systems for the emergency department*
    - **Addressed problem**: Developing an accurate and reliable system to predict SARS-COV-2 and oxygen therapy requirement in emergency department patients.
    - **Proposed solution**: A machine-learning-based prediction system that uses a combination of clinical and laboratory data to predict SARS-COV-2 and oxygen therapy requirement.
    - **Advantages**: The model has an area under the curve exceeding 93%, recall for SARS-COV-2 detection of 96%, F1-score for SARS-COV-2 detection of 92%, and F1-score for oxygen therapy prediction of 83%. The precision for SARS-COV-2 detection and oxygen therapy prediction is continuously above 80%.
    - **Disadvantages**: The study is limited to a single hospital and further testing is necessary to determine its generalizability to other hospitals or populations. It also requires access to laboratory data, which may not be available in all settings.
    - **Main contributions**: The model has improved results compared to existing models that use a smaller, unbalanced dataset and fewer features. It uses both clinical and laboratory data, which increases accuracy and reliability, and has the potential to aid clinical decision-making in emergency departments. The machine-learning algorithm can also be easily updated as new data becomes available.
- *Deep learning and Lung UltraSound for SARS-CoV-2 pneumonia detection and severity classification*
    - **Addressed problem**: Lack of reliable, accurate and prompt diagnostic tools for SARS-CoV-2 pneumonitis detection and severity classification using traditional methods.
    - **Proposed solution**: A deep learning-based model using Lung Ultrasound (LUS) images for pneumonia detection and severity classification.
    - **Advantages**: LUS is non-invasive and widely available, provides high accuracy and sensitivity, reduces exposure to ionizing radiation, enables comprehensive diagnosis of SARS-COV-2 pneumonia using LUS images, and allows for high accuracy in both pneumonia detection and severity classification, reducing diagnosis time.
    - **Disadvantages**: LUS requires substantial expertise and high-quality data, which may not be widely available in all clinical settings, particularly in resource-poor regions, although LUS is cheaper than other technologies. There is

also a lack of large-scale data for model training and a need for expert annotation of LUS images.

- **Main contributions**: The use of LUS data improves the accuracy and efficiency of SARS-CoV-2 pneumonitis diagnosis and enhances the state-of-the-art SARS-CoV-2 detection. The proposed model provides a comprehensive diagnosis of SARS-COV-2 pneumonia and outperforms traditional methods in accuracy and time efficiency.

Chapter 6

Epidermal lesions assessment through deep learning, high-performance computing and hyperspectral imaging

As mentioned in Chapter 2, skin cancer is one of the most common in the world and comprises non-melanoma (NMSC) and melanoma (MSC) skin cancer. NMSC was the 5th most common form of cancer worldwide in 2018, while melanoma was the 21st [42,43]. Pigmented skin lesions (PSLs) derive from the excessive growth of melanocytes, and academia usually categorises them as benign or malignant[45,46]. Atypical moles, also known as dysplastic nevi, are benign PSLs associated with a high chance of evolving into melanoma.

Hyperspectral imaging is an imaging spectroscopy technique producing three-dimensional images whose pixels illustrate the spectral content of a scene. This cube contains the fraction of incident electromagnetic radiation reflected from a surface. Each material presents a specific variation of reflectance values concerning wavelengths, called spectral signature, unique for each type of material, allowing precise discrimination[33,48,49]. HSI classification approaches seek to identify each pixel's material. The classification comprises several supervised and unsupervised algorithms whose elaboration could be computationally intensive.

Recent technological advances facilitated the use of HS images in fields like medicine for cancer detection[12,33,48,49]. This technique is exploited, especially in tumour diagnosis, because it is non-invasive, non-contact and non-ionising, capable of obtaining spatial and spectral information. Besides, lesions modify the biochemical and morphological tissue structures, causing different optical characteristics concerning healthy tissues, such as absorption, scattering or fluorescence. Consequently, the divergences deliver valuable diagnostic information in the diagnostic and detection stages[18], in which is crucial to meet fast or near real-time responses.

This chapter concentrates on the statistical, AI and high-performance computing approaches this doctoral thesis researched to counteract epidermal tumours from hyperspectral imaging. The studies concern machine and deep learning strategies, handling the dataset described in Section 2.7.

Investigations started exploring standard ML to later dive into novel DL approaches, whose challenges concerning medical and small-sized datasets were described in Chapters 2 and 3.

Nonetheless, investigations went beyond classical statistical testing and delivered the real-time embedded deployment onto the GPUs of some of the algorithms discussed in this chapter.

Close collaboration with Universidad de Las Palmas de Gran Canaria, Hospital Materno Infantil, and Hospital Doctor Negrìn, enabled the research investigations just mentioned[38].

In the following lines and sections, this chapter describes the state of the art methodologies and results applied to the problems mentioned above. Afterwards, it contains a brief exploratory analysis concerning Chapter 2's dataset employed in all the projects contained in this chapter.

Then, for each of the works researched in the educational path described in this thesis, the chapter contains a section describing the materials and methods of the study, the results, and the ending remarks. These address the discussion of the results, conclusions, and implications that advance the field based on current knowledge and our achievements. Accordingly, this chapter will cover investigations concerning all the theoretical aspects listed in Chapters 2, 3 and 4.

6.1. AI and HPC state-of-the-art concerning epidermal tumours

Concerning healthcare applications, academics have designed hyperspectral acquisition systems regarding skin, brain, and plastic samples[11,38,97]. The investigations present disadvantages related to most of the theoretical aspects we analysed in Chapters 2 and 3. Namely, camera type, sensor fusion elaborations or real-world relevance. Present designs differ mainly in the camera employed, their cost and weight, the materials and the existence of customised graphical user interfaces (GUIs). Among the considered investigations, the first employs a snapshot camera to image the region of interest, offering the lowest spatial and spectral resolution among the cameras[38]. The second comprises two pushbroom cameras to offer different wavelength sensitivities, thus delivering higher spatio-spectral resolution but high processing duration, sensor fusion synchronisation and device weight-critical aspects[11,12]. Ultimately, although the latter comprises pushbroom cameras for plastic research discussing laboratory implementation, it suggests high-cost and implementation challenges[97].

Researchers conceived some of the hyperspectral acquisition systems mentioned above to detect skin cancer at its early stages. Accordingly, they aimed to design AI applications to strengthen current diagnostic performance whose significance leans heavily upon dermatologist expertise[18]. Several reviews assessed learning-based investigations about skin cancer diagnosis, embracing several data types, including HS images, emphasising their strengths and shortcomings. Different systematic review articles concentrated on more than fifty investigations affecting different data types and learning methodologies, involving hundreds of dermatologists for direct comparison[45,47,98].

Likewise, the reviews stressed that research should consider the learning strategy and device development to overcome current challenges. These comprehend data availability, interpretability, and computational power, which operating recent DL algorithms and having real-world clinical scenario applicability could solve. Current AI algorithms are still in the very early stages of clinical application. They are only sometimes prepared to assist doctors, but they can be scalable to multiple devices, converting them into contemporary medical instruments[3]. Such novel devices will also accumulate data, overcoming the data availability aspects.

State of the art examinations debate primarily in the architectures engaged, namely artificial neural networks (ANNs) and convolutional neural networks (CNNs), and the data for the training stage. Most investigations used CNNs and dermoscopic images to diagnose epidermal lesions since DL algorithms and high-quality data contribute to better performance. At first, experimenters manipulated ANNs to reproduce the ABCD rule with an accuracy between 70 and 90%. Regardless, small-diameter lesions caused the diagnosis to be more challenging, introducing misclassifications[45,47,98]. Although CNN's introduction improved the solutions, the issues stay since lesions from different etiologies (Section 2.7) have subtle visual divergences. Generally, the experimentations feature diagnostic performances comparable to skilled dermatologists, whose sensitivity, specificity, and accuracy concerning benign and malignant lesions are around 80, 75, and 70%. Board-certified dermatologist accuracy decreases to around 55% when more classes challenge the diagnostic task[46]. Consequently, multi-class classification scenarios worsen the diagnostic evaluation.

Furthermore, studies show that researchers usually trade off low specificity for high sensitivity. Additionally, lesions already marked as suspicious prior to investigation typically biased the outcome metrics[45–47,98]. Undoubtedly, results show that DL algorithm performance improved by over 90% only when researchers conducted experiments with an unconventional binary classification task, namely malignant melanoma (MM) and Basal Cell Carcinoma (BCC) or nevus.

Other studies involved histopathological and clinical images, which exhibited comparable performance concerning dermoscopic data. Regardless, pathologists surgically extracted part of a suspicious mole and

applied labelling to perform diagnosis. Similarly, clinical images introduced worse diagnostic evaluations, even worsened when the researchers considered more than two etiologies[45,47,98].

Early investigations on skin cancer applied machine learning to identify PSL using HS in-vivo skin cancer data[38,45,47,98,99]. The authors used a genetic algorithm to optimise the supervised machine learning algorithms to identify four PSL types: nevus, BCC, MM, and other PSL types. Others proposed an HS classification strategy combining unsupervised and supervised algorithms to discriminate between malignant and benign PSLs. Other retail approaches, such as SIAscope/SIAscopy or MelaFind, use multispectral pictures to notice only melanoma lesions[38,45,47,98,99].

The literature generally concentrates on ML practices for medical HS image classification. In recent years, DL emerged as the ideal solution for end-to-end classification tasks[38,45,47,98,99]. On the other hand, DL algorithms mainly involve HSIs for remote sensing. Thus, at the time of writing, DL architectures should be investigated better for HS medical image classification. Among the different strategies, Vision Transformers (ViT) have recently appeared in the literature. These architectures lean on the self-attention mechanism, at first developed for Natural Language Processing (NLP), which retains a very high number of parameters[100].

Computing conveys a crucial aspect since most contexts require uniform real-time responses. Consequently, academia proposed several experiments manipulating parallel technologies for HSI classification[11,12]. Modern systems adopt a concurrent elaboration of the image pixels to reduce the processing time when feasible. Accordingly, parallel technologies are suitable for pixel-wise classification, where each processing core elaborates on a single pixel or a group of pixels. The literature leans on hybrid systems, including the multicore and the many-core devices we described in Chapter 4.

Hybrid systems suit the processing chains' features that typically include algorithms with diverse tiers of intricacy[1,13]. The key idea is that each device manages the algorithm that best meets the processing requirements.

Concerning skin cancer, the output of these classification systems is typically a thematic map, where researchers assign each pixel with a colour representing a specific tissue type or lesion condition. Nevertheless, systems exist that directly provide a diagnosis without any semantic map of the HS image.

Although AI's first medical adoption occurred in the 1980s[1], researchers have only recently offered solutions for clinical practice. Current sensors gather a broad mixture of knowledge and produce astounding data to train perceptive systems.

AI algorithms deliver robust and reliable classification performance associated with statistically complete and labelled datasets. Indeed, ML performance is directly proportional to the training data available (Chapter 3). Regardless, more than the available amount of labelled information is usually needed in healthcare, mainly when researchers evaluate DL architectures. Consequently, they focus on techniques to generate

statistically relevant synthetic data representative of real situations[55,101]. Different studies proposed architectures operating traditional RGB images, chest X-rays, electrocardiograms, or HS data for diagnostic purposes[1,48,49]. The authors exploited traditional ML algorithms in these works due to the poor dataset size.

Synthetic HS data could originate from a mathematical model considering the interaction between light and matter. However, such a solution development is not feasible due to the physical uncertainties and computational complexity required to model physical light-matter interactions.

The so-called data augmentation process (Section 3.12) refers to geometrical, colour-based, or additive statistical-based noise transformations. Consequently, the procedure transforms the images to yield new instances and increase the statistical variance of the knowledge contained in a dataset. Nevertheless, the size of the original population confines the augmentation usage. Indeed, it is only sometimes feasible to generate a reasonable number of new samples as both the dataset and the number of augmentations are finite.

Researchers overcame such limitations by conceiving the generative adversarial networks (GANs) we described in Section 3.19.

Concerning healthcare applications, authors have already adopted GANs in image denoising, segmentation, classification, and image synthesis[56]. Nonetheless, academia can consider the innovation that comes from applying GANs to hyperspectral imaging (HSI) since only a single study is available[101]. Undoubtedly, it only presents a proof of concept, demonstrating the capability of GANs to generate HS skin cancer images. The authors validated their results only by comparing the typical average spectral reflectance of real and synthetic data. Nonetheless, this research suffers several limitations[101]. Although the authors considered four different lesions (dysplastic nevus, melanoma in situ, malignant melanoma, and benign nevus), they conceived a final validation concerning a typical spectral reflectance without comparing the different lesions. Moreover, the authors do not provide the GAN-generated image with a class.

6.2. In-vivo hyperspectral dermal database

Chapter 2 mentioned the dermatological HSIs acquisition system, which generated a database of in-vivo HS skin lesions. The database was subjected to subsequent analyses to investigate the efficiency of HSIs in discriminating skin tumours.

The data acquisition campaign occurred in the time interval between March 2018 and June 2019. Images of various skin lesions were acquired, located in different body parts, from 116 subjects at the Doctor Negrin University Hospital in Las Palmas de Gran Canaria and the Materno Infantil University Hospital Complex. The Comité Ético de Investigaciòn

Clínica-Comité de Ética en la Investigaciòn (CEIC/CEI) approved protocol and procedures. A preliminary analysis of the acquired data removed 55 images from the database because these derived from critical areas (e.g., shoulders, nose, chin and other face parts) that made it challenging to acquire them under optimal conditions, preventing the complete lense contact with the skin surface. Hence, the final database comprises 76 images referring to 61 subjects. Some images refer to the same patient but to skin lesions of different types positioned in various body parts[38,99].

If the dermatologist doubted an epidermal lesion, it was histopathologically removed for examination to obtain a definitive diagnosis.

Professionals clustered the images in the database and assigned each pixel a specific label from one of the aetiologies described in Section 2.7's Figure 11. The procedure yielded 15961 pixels for classification experiments using ML algorithms. Initially, lesions comprised benign, malignant, and atypical classes.

The present thesis work carries out the analysis and classification of the HSIs from this database. Experts manually segmented the images, generating the ground truth, and labelled them to distinguish the categories we mentioned in Section 2.7.

6.3. HS images pre-processing

The data needs a pre-processing stage before classification to have homogeneous spectral signatures among the patients. In unbalanced databases, such as the one under study, this process helps limit the statistical variance of the spectral signature of the classes present with lower frequency.

The pre-processing consists of four steps:

1. Calibration
2. Removal of bands at the extremes of the spectrum
3. Noise filtering
4. Normalisation

Calibration allows the correction of any distortions, for example generated by non-uniform illumination, present in the raw HSI acquired (Y). Calibration comprises two reference images: the white image (W), derived from a reference (ref) capable of reflecting 99% of the incident light, and the black image ($Dref$) delivered with the camera shutter closed and the light off. The calibrated image (Y) originates from Equation 23:

$$\boldsymbol{reflectance} = \frac{\boldsymbol{raw\ image} - \boldsymbol{W_{ref}}}{\boldsymbol{W_{ref}} - \boldsymbol{D_{ref}}} \qquad \text{Equation 23}$$

Since the sensor has a low response to the bands at the spectrum's ends, the noisy bands removal occurs. Specifically, the first four and the last five wavelengths disappear, moving from an initial spectral resolution of 125 to a final spectral resolution of 116.

Subsequently, we apply a moving average smooth filter to remove noise further. Consequently, the value of each pixel leaves a place for the average of the values of neighbouring pixels. Filtering leans on Equation 24 where $y\ (i)$ is the new value assigned to the pixel, N is the number of neighbouring pixels considered, and $2N + 1$ is the span.

$$\mathbf{y\ (i)\ =\ 1\ (y(i\ +\ N)\ +\ y(i\ +\ N\ -\ 1)\ +\ \cdots\ +\ y(i\ -\ N))} \qquad \text{Equation 24}$$

The following rules realise the filter:

- The span must be odd
- The filtered value must be in the centre of the span
- The span must account for values that cannot satisfy the correct number of neighbours
- The start and end values are not filtered, as a span cannot be defined

Finally, we apply normalisation of the data between 0 and 1 to homogenise the reflectance values of the entire dataset[38,99].

6.4. In-vivo HS data exploration

The Universidad de Las Palmas de Gran Canària gathered a database of in-vivo HSIs, containing 76 images in the form of 50x50x125 hyperspectral cubes. Each hypercube has a 50x50 ground truth, wherein physicians clustered each pixel through a specific skin or skin lesion label. Figure 62 shows the ground truths of the entire database, with each type of label associated with a colour: black for skin, shades from yellow to red for malignant skin lesions and shades from green to blue for benign skin lesions.

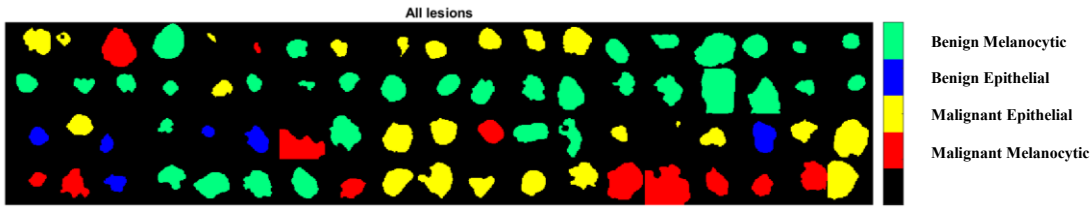


Figure 62. Ground truths of the entire epidermal database

We can distinguish four different labels, according to the convention shown in Figure 11.

As none documented this database before, it was necessary to check its contents to get a general idea of the data distribution.

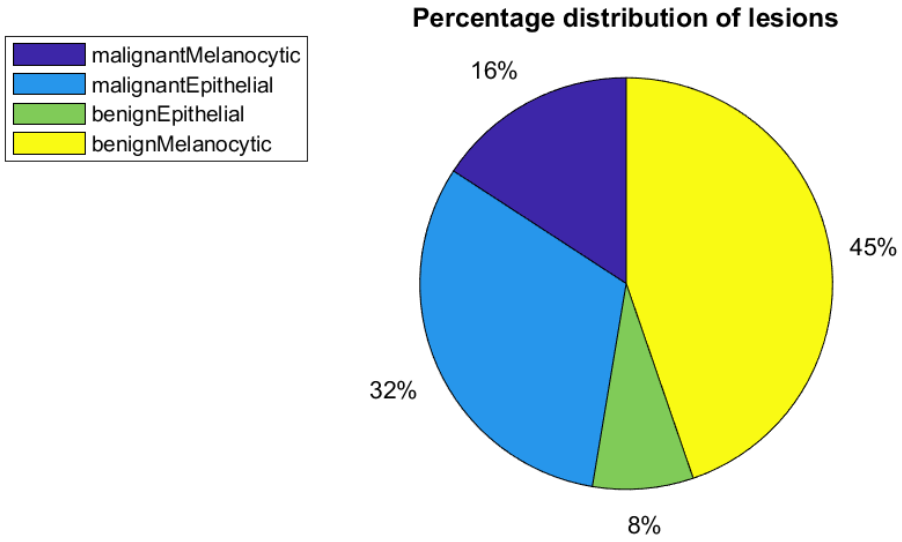


Figure 63. Percentage distributions of skin lesions: benign-malignant distinction (left); multi-class distinction (right)

First, this thesis investigated the skin lesions percentage distribution, as shown in Figure 63. This analysis showed that the database is reasonably balanced in the categorisation into benign and malignant: 53% of the lesions are benign, whilst 47% are malignant. Regarding the differentiation of tumour subclasses, the most prevalent classes are benign melanocytic (45%) and malignant epithelial (32%). However, the other two tumour classes have reduced frequency within the database. The distribution of lesions within the database broadly reflects the actual distribution of tumour forms in the population.

Accordingly, the investigation measured the lesions' average areas, depending on the class considered. It yielded boxplots, graphic representations of the data highlighting their median value and the ranges of variation of the surfaces in percentage value. The aim was to verify to what extent the lesion size could give clues about its classification, reflecting the ABCD rule and confirming the validity of the database. Figure 64 shows, as an example, the percentage distribution of malignant tumour lesions' area. Melanoma differs from other types of malignant skin tumours by its size.

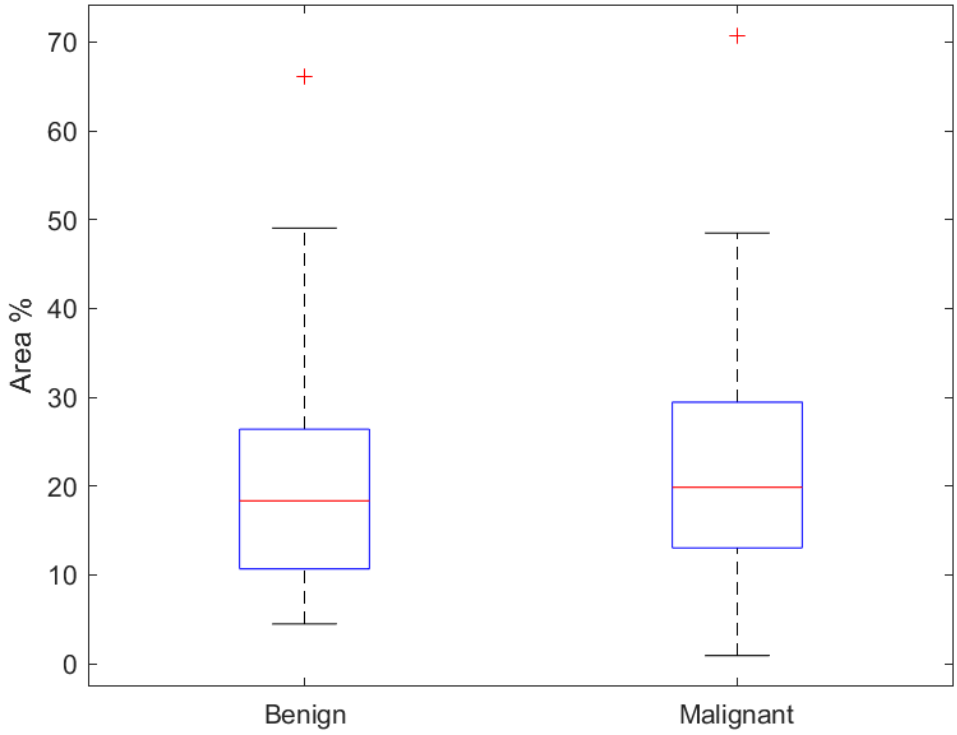


Figure 64. Benign-Malignant areas comparison

Finally, this section analyses the average spectra of the skin and the various skin tumour types, including standard deviation. It was thus possible to compare the spectral signatures associated with the various classes. Figure 65 shows the mean spectra of malignant and benign skin lesions with associated ranges of variability. Some signatures are pretty distinguishable, while others show non-negligible variability. Therefore, the average spectral information is insufficient for the classification of lesions. Accordingly, this doctoral thesis investigated CNNs and more sophisticated methodologies capable of autonomously extracting the relevant features during the discrimination process.

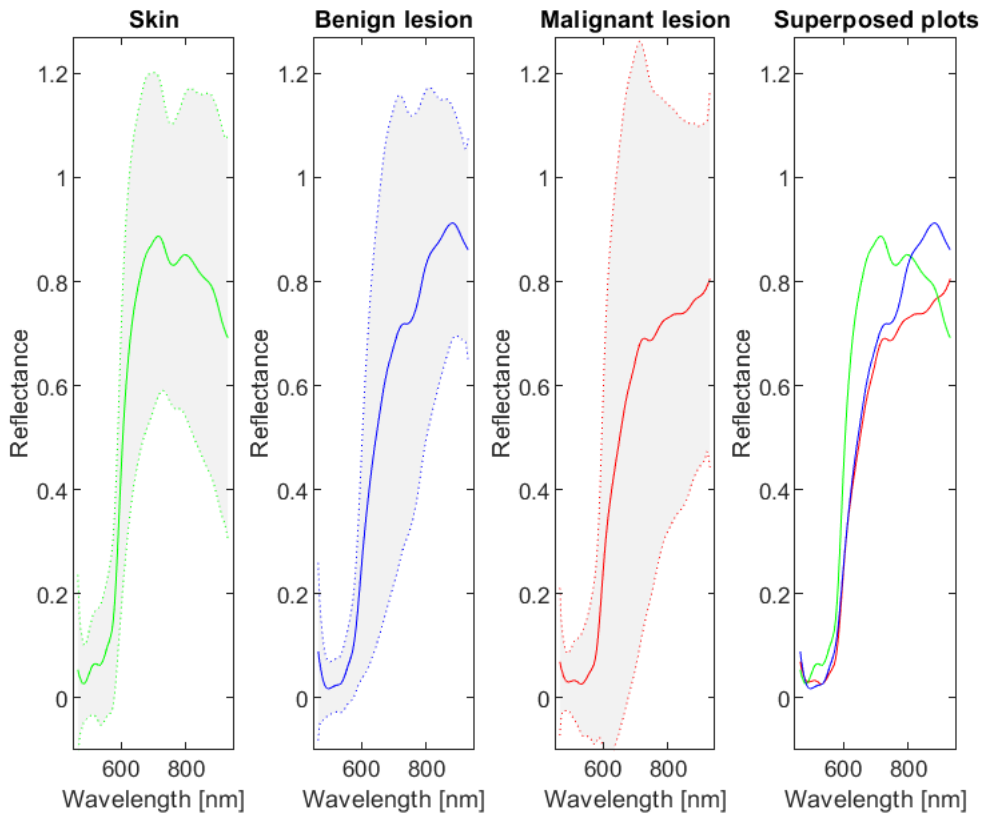


Figure 65. Average spectral signatures comparison for the epidermal HS dataset

6.5. Hyperspectral imaging acquisition set-up for medical applications

Here, the thesis introduces a breakdown to overcome the challenges mentioned in the literature review section, instructing a hyperspectral imaging blueprint designed to work with pushbroom sensors, one of the highest-quality detectors, to photograph a region of interest. It works in various contexts, and the investigation conceived it for dermatological or surgical procedures which interest a motionless subject. The system comprises inexpensive, open-source and consistent components. Consequently, it includes Python libraries, an Arduino UNO board, a Nema-17 stepper motor, its driver controller, and a recirculating ball screw for precise movement. Similarly, it offers a diode-based targeting procedure, hooked to a 3D-printed circular crown, to measure the right focusing span. The blueprint comprises a GUI to let healthcare professionals interact with the imaging system, move it with high accuracy, and gather diagnostic data.

In summary, the investigation's main contribution is the proposal of an affordable hyperspectral imaging system and its detailed implementation report. Furthermore, not only does this chapter discuss its development

challenges and strategies, delivering a GUI to automatically stir the calibration and acquisition protocols and subsequent data storage, but also its validation to allow reproducibility.

6.6. Acquisition set-up and building blocks

One of the design's main goals is automatically imaging skin areas with pushbroom sensors with few settings. The only step needed is acquiring black and white reference images for successive calibration pre-processing. This section provides all the information concerning the system's design and building blocks in terms of hardware and software modules. Figure 66 displays the hyperspectral imaging platform system.

Figure 66. Hyperspectral imaging system presented blueprint

6.7. Specim FX-10e hyperspectral camera

The blueprint addressed in this doctoral thesis operates with any Generic Interface for Cameras (GenICam) compliant pushbroom camera that can acquire only one strip of pixels at a time[97]. They require the movement of the target object or the camera to scan the entire scene. The camera used in this study is a Specim FX10e, and it is a VNIR camera, therefore sensitive to visible and near-infrared wavelengths between 400 nm and 1000 nm.

Table 13 shows some relevant technical characteristics of the FX10e model. The camera's lens has a 1.7 F-number, which is the ratio between the focal distance f and the diameter of the lens. The Field Of View (FOV), the detector's sensitivity angle to electromagnetic radiation, is 38°, and we operated a 15 cm focusing distance.

Table 13. Specim FX10e hyperspectral camera specifications

Technical specifications	**FX10e**
Spectral range	400-1000 nm
Detector type	CMOS
Slit width	Physical width 42 µm. Projection on sensor 32 µm.
Pixel pitch	16x8 µm
# Spatial pixels	1024
Binning (spectral x spatial)	2 x 1
Spectral binning options	2x | 4x | 8x
# Spectral bands covering the specified range	224 | 112 | 56
Spectral sampling/pixel	2.7 nm | 5.4 nm | 10.8 nm
Spectral resolution FWHM	5.5 nm (mean)
SNR	600:1
Frame rate (fps) full range (220bands) max	330 fps
Frame rate (fps) MROI examples	20 bands = 2800 fps 5 bands = 6500 fps
Shutter	Electromechanical shutter for dark background registration

The pixel pitch is the pixel size at the sensor and is 16x8 µm. This 2x1 ratio size means the camera gathers two spectral pixels for each spatial pixel. We can change the ratio between spectral and spatial pixels (i.e., 4x1 or 8x1), but the spatial dimension of a detector's pixel does not vary and is 8 µm. Regardless, the pixel size at the scene plane, called pixel size, is not 8 µm. It depends on the FOV, the number of effective strip pixels, and the focusing distance from the lens. The third may vary if the first two parameters are determined (38 ° and 1024 pixels).

In this thesis, we decided to operate continuously at a focusing distance of 15 cm. Accordingly, it was possible to calculate the size of a pixel at the scene plane with Equation 25:

$$\textbf{Pixel size} = \tan\left(\frac{\textbf{FOV}}{2}\right) \times \frac{2h}{\textbf{Np}} \qquad \text{Equation 25}$$

In the equation above, h is the object's distance from the target, FOV is the angle described earlier, and Np is the number of strip pixels. Hence, employing our datasheet's values, the outcome is approximately 100 µm. The total imaged scene's width, also known as field dimension, depends on the following trigonometric formula in Equation 26:

$$\textbf{Field dimension} = \textbf{2h} \cdot \textbf{tan}\left(\frac{\textbf{FOV}}{\textbf{2}}\right) \qquad \text{Equation 26}$$

It corresponds to the pixel size multiplied by Np (i.e., 1024), which is approximately 10.3 cm. Figure 67 exhibits the computations just mentioned above.

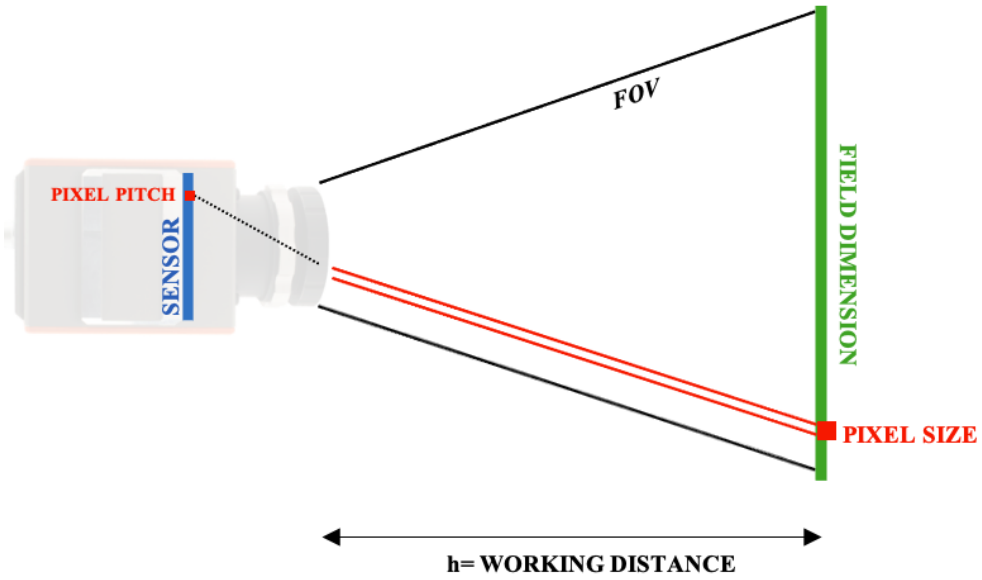


Figure 67. Hyperspectral camera optical schema for trigonometric equations

The evaluations concerning the FX10e's optical characteristics are meaningful for design and functionalities, as we will see in the following paragraphs, concerning the camera's frame rate and motor's movement synchronisation.

6.8. The motion system

The motion structure comprises a recirculating linear ball screw guide, a stepper motor and a driver driven by an Arduino UNO board. This machine can collect the reflectance spectrum of any region of interest line by line[40]. Pushbroom cameras need a linear movement between the camera and the sample to entitle complete scanning by moving the camera or the sample at a steady and constant speed. The investigation designed a structure targeting skin cancer assessment where the camera moves while the target

is stationary. Accordingly, motion-acquisition synchronisation is crucial. Proper calculations concerning the previous section and the software interface design controlling the motor's motion facilitate such synchronisation.

Figure 68. Recirculating ball screw drive

The linear ball screw guide is made of aluminium and has a length of 200 mm, a diameter of 12 mm and a pitch of 4 mm (Figure 68). The screw guides the Schneider Electric NEMA-17 stepper motor's shaft. If N is the number of motor expansions at each step, the motor moves by $\theta = \frac{360°}{4N}$, controlling the angular position and speed by varying the steps' frequency. NEMA17 performs 1.8° of angular displacement at each step, taking 200 steps to complete a revolution. It is driven in current by a TB6600 Driver (Figure 69), which controls its speed and direction.

Figure 69. Microstep-based motor driver

The driver allows choosing between eight micro-steps (Figure 70), promoting the motor's angular step division into n substeps. Thirty-two micro-steps (Figure 70) correspond to a 0.05625° step angle, resulting in 6400 steps for an entire revolution. The pitch of the screw resembles its linear motion and equals 4 mm. Consequently, the motor moves by 0.625 µm at each micro-step. Correspondingly, the driver supplies 1.5 A to the motor, as the datasheet suggests.

Micro Step	Pulse/Rev	S1	S2	S3
NC	NC	ON	ON	ON
1	200	ON	ON	OFF
2/A	400	ON	OFF	ON
2/B	400	OFF	ON	ON
4	800	ON	OFF	OFF
8	1600	OFF	ON	OFF
16	3200	OFF	OFF	ON
32	6400	OFF	OFF	OFF

Figure 70. Motor driver's datasheet

An Arduino UNO board controls then the driver. We programmed the board via the Arduino IDE (Integrated Development Environment) during the investigation. The sketches are written in Wiring, similar to the C language, allowing flashing to the board.

Figure 71 displays the wiring diagram of the motor, driver and board connections. We suitably positioned the power supply, driver, and board and fixed them inside a single container for electrical components measuring 190x140x70 mm, placed near the linear guide.

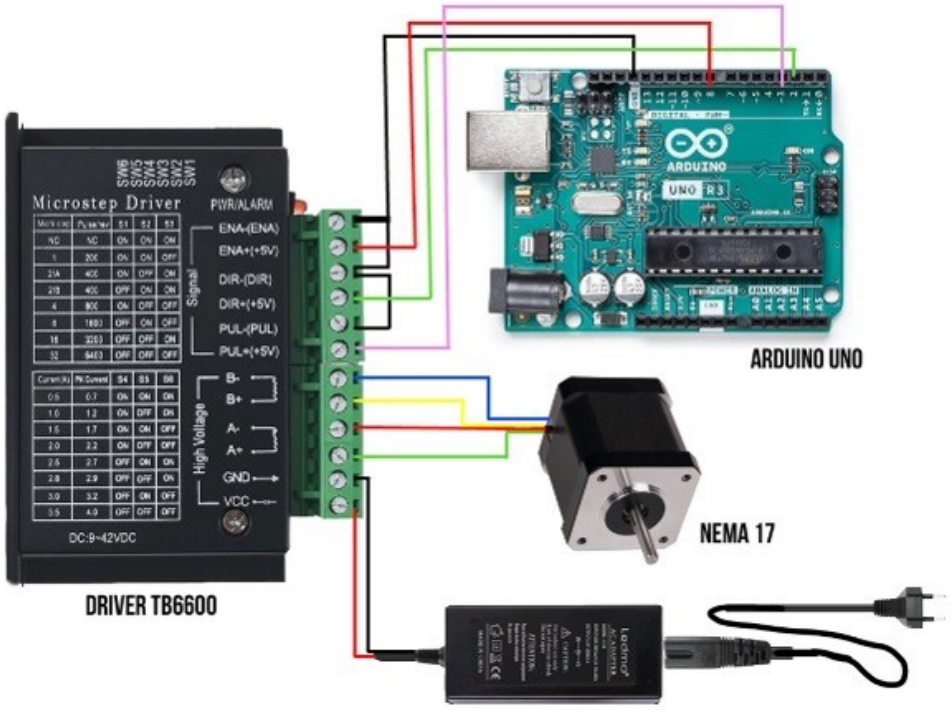


Figure 71. Electrical configuration of the hyperspectral blueprint

6.9. The illumination system

In hyperspectral imaging, the scene's illumination is essential. Clearly, any camera measures the light beam reflected by the object of interest. Researchers must carefully choose the light source according to its sensitive wavelengths in hyperspectral applications[40]. Proper illumination features a continuous intensity spectrum without peaks and with a good intensity contribution in amplitude. Sunlight, is an excellent option for continuity and intensity spectrum at all wavelengths. Nonetheless, it is not easy to handle as it varies rapidly in direction, intensity and colour.

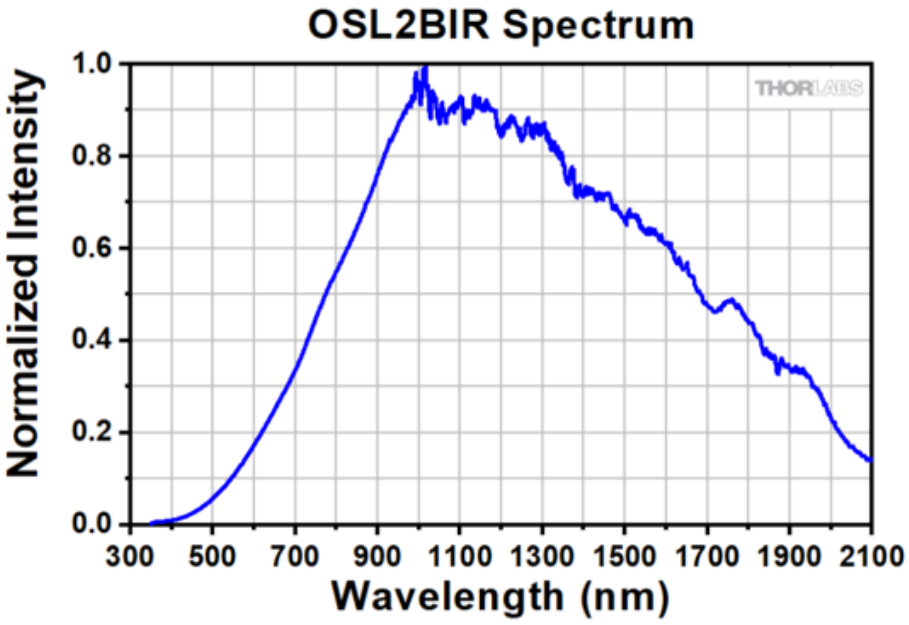


Figure 72. THORLABS OSL2BIR 150 W 3200 K intensity spectrum

On the other hand, an artificial light source allows complete control regarding direction, intensity, and scattering and delivers repeatability in acquisition circumstances. This thesis sought illumination with a continuous spectrum and good intensity between 400 and 1000 nm, in which the Specim FX10e camera operates. To allow light sources differentiation among the available options, these beamed the white reference, and we visualised the reflectance using Specim LUMO software. Consequently, the illumination system operated in this work is the THORLABS OSL2BIR 150 W 3200 K, which has an aluminium-coated reflector for improved infrared performance. The specifications include the intensity spectrum shown in Figure 72. The illumination system comprises two bulbs mounted on two supports which the investigation arranged to be directed to the region of interest and granting distance regulation.

6.10. Image calibration

Several steps must happen to obtain a hyperspectral datacube with a pushbroom camera correctly. This chapter reported how optical variables, motion structure, synchronisation requirements, and correct illumination influence the data quality. Another crucial step is the image calibration process[97]. Hence, this investigation performed calibration following Equation 23 reported in Section 6.3. In the equation, the white reference is the image of a zenith polymer white calibration panel that is certified to reflect more than 95 per cent of the incident radiation in the range of 400 to

2500 nm. We placed the white reference panel at 15 cm. The dark reference, on the other hand, represents the minimum value that the sensor measures when no radiation hits it. We ultimately closed the camera length to obtain the black reference.

6.11. Target centring and distancing

The investigation faced two practical problems in obtaining images: correct focusing length measuring and target centring. Accordingly, the investigation devised a structure equipped with laser-emitting diodes. We mounted on a custom-made 3D-printed crown (Figure 73) two 5 mW - 5 V small red laser emitting diodes, both driven by the Arduino board. The emitters assemble at a precise distance of 15 cm, the focusing distance, at the centre of the lens. The diodes are only switched on in the moments immediately prior to the acquisition, we check the distancing and centring, and then the scanning process can start.

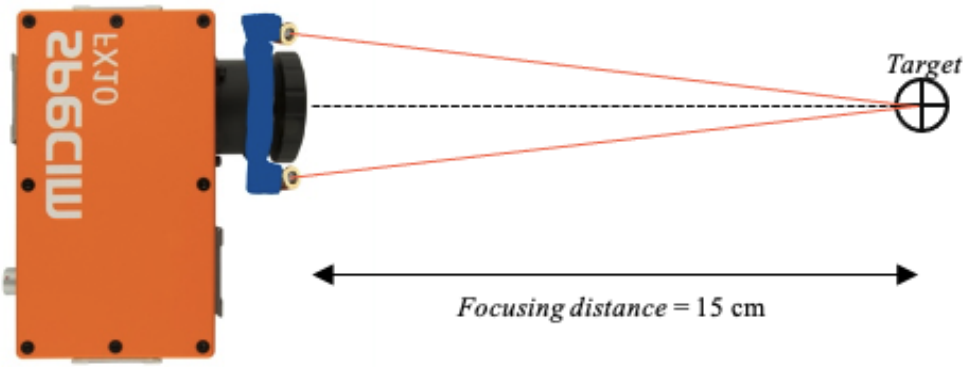


Figure 73. Hyperspectral set-up targeting and distancing system

6.12. Camera control, system synchronisation and image scanning

The Specim FX10e camera's rear panel has two connectors: the 12 V DC power cable and a GigE connector. The GigE Vision protocol is an international interface standard for video transfer developed for high-performance retail cameras and managing devices over Ethernet. Therefore, we developed software to acquire images in the laboratory using any GenICam-compliant hyperspectral camera using the gigabit ethernet interface[97]. The camera gathers and sends information via Ethernet when a trigger arrives, allowing the camera's image sensor exposure to start. It can be generated internally by the camera (free running) or by an external device (external trigger). The research adopted the external triggering mode

in this blueprint to synchronise the motor's motion and camera frame rate. It operated the python pyserial library to control the motor, facilitating software transmission with the Arduino's serial port.

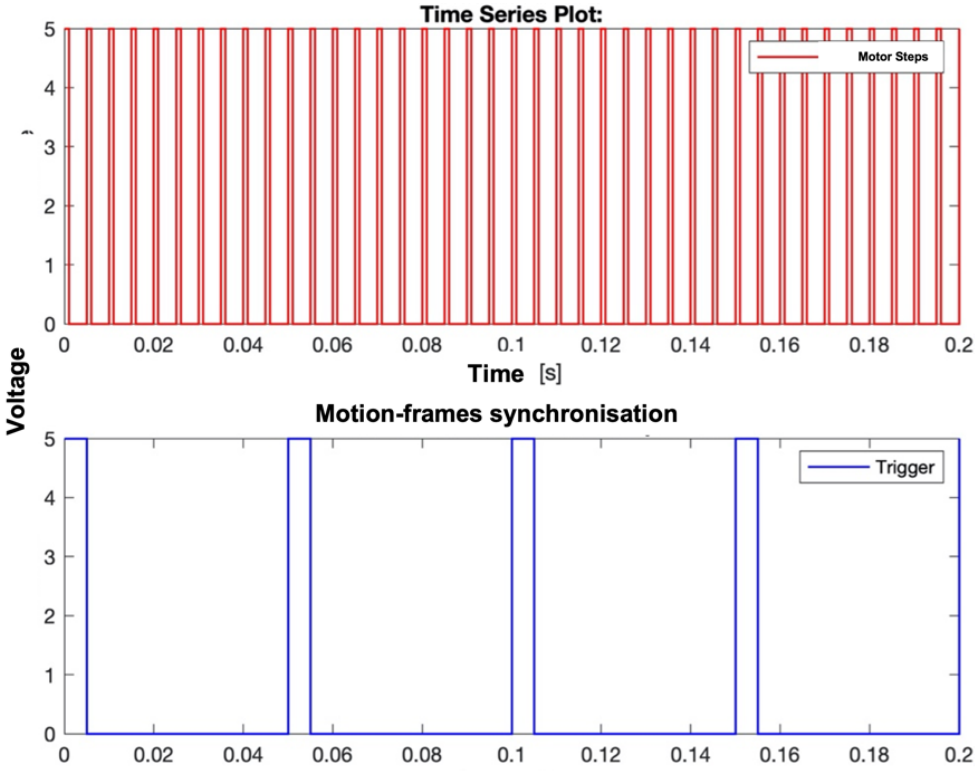


Figure 74. Motion-frames synchronisation

Accurate motor speed and position control are essential to obtain a correct hypercube. Since the scanning result is closely related to the concepts of trigger and exposure time, the accuracy and smoothness of the camera movement directly affect it[97]. Along with pyserial, this thesis operated the harvester library, which guarantees image acquisitions under the GenICam standard[97]. Through the Harvester routines, we can perform the main camera control actions, such as starting and stopping the data flow, gathering the captured frames, storing them to disk, and configuring any acquisition-related parameter. The mixture of the functionalities provided by the harvester and pyserial libraries makes it feasible to control the camera and motor concurrently, enabling synchronisation.

The investigation fulfilled motion-acquisition synchronisation by revising the control signals accordingly. The PWM signal generated by the Arduino board's pin regulates the movement, controlling the motor's steps. With each signal's positive edge, the motor takes one step. We can control the motor's direction by operating another board's pin. Hence the need to capture frames via external triggering. The camera performs a frame capture at each positive edge of the trigger signal, driven by the harvester

library. Concerning the optical evaluations carried out in Section 6.7, we should notice that the pixel size at the image level is approximately 100 µm. Therefore, the laboratory blueprint must sample at least once every 100 µm linear displacement to convey a complete image. The motor performs 2.5 µm of linear motion at each step in this manuscript's configuration. Accordingly, 40 motor steps are required to perform 100 µm. The investigation captured four frames every 100 µm and averaged them to avoid aliasing and noise (Figure 74).

Consequently, the Arduino UNO board sends a pulse to the motor every ten steps and a pulse to the trigger pin simultaneously as the tenth motor pulse. This way, a scene frame is captured every ten motor steps. This configuration synchronises the signals and guarantees precise and fully controlled capturing. The choice of motor step optimised acquisition quality and time. In this mode, to acquire an area of 10x2 cm, the acquisition time is 40 seconds.

6.13. The Graphical User Interface (GUI)

This thesis designed the GUI using the PyQt5 python library (Figure 75). The development employs a drag-and-drop strategy which encourages the interface creation process. It is necessary to determine the main window, divide it into frames, assign types and names to the various elements and position the buttons as wished. At the end of this step, the library generates the corresponding code in python, which we can modify. The various generated buttons can link to any custom routine. The interface considerably improves the system's usability, makes the button-action relationship performed immediately and facilitates the acquisition process[97].

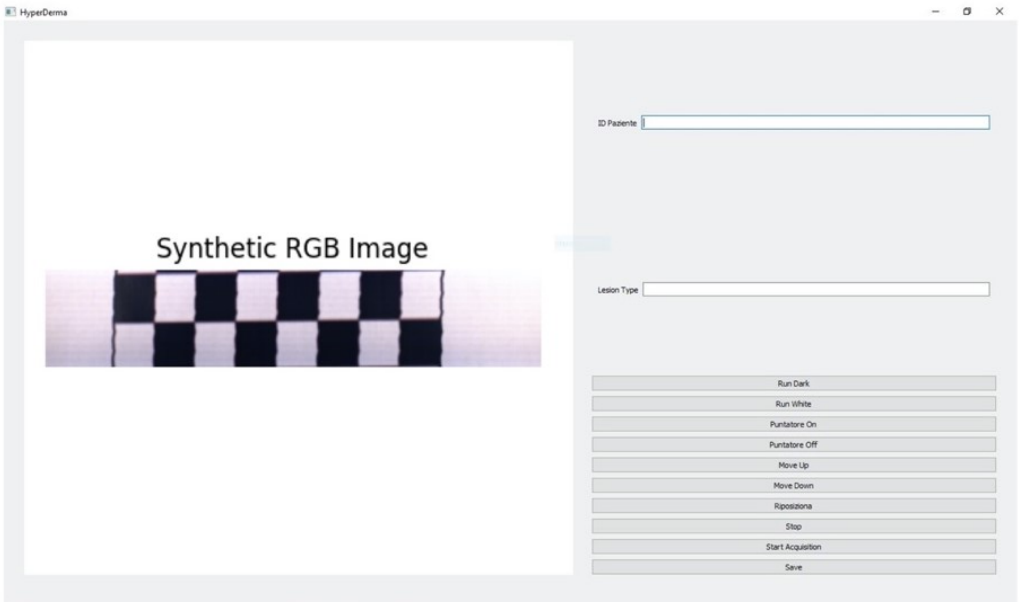


Figure 75. Hyperspectral set-up's GUI

The GUI (Graphical User Interface) implemented, as shown in Figure 75, includes the following command buttons and text fields:

- **Patient ID**: filling this text field stores the files referring to the acquisition performed with this identifier
- **Lesion Type**: the acquired file is also assigned a preliminary classification based on the doctor's belief
- **Run Dark**: the calibration of the black reference performs an image acquisition with the lens closed and stores a numpy file with which the system can perform the calibration later
- **Run White**: the white reference calibration performs an image acquisition upon the white reference panel placed in front of the camera at 15 cm. It stores a numpy file with which the system can perform the calibration later.
- **Pointer On**: turns on the laser diodes, to be used just before acquisition to centre the target
- **Pointer Off**: turns off the laser pointer diodes
- **Move Up**: moves the camera up and continues to move up until the button Stop is pressed
- **Move Down**: Moves the camera downwards and continues to move down until the button Stop is pressed
- **Stop**: Stops the motor
- **Start Acquisition**: it starts the scanning process. The camera moves up 1 cm and then down 2 cm, capturing the scene's frames
- **Save**: stores a series of files referring to the datacube *npy*, *img*, *hdr* and the RGB image in *png* format

At the end of an acquisition process, namely approximately 40 seconds after clicking on the Start Acquisition command, a synthetic RGB image (synthesised from three bands: Red = 700.47 nm, green = 546.09 nm and Blue = 435.79 nm, calculated within the 224 bands acquired between 400 nm and 1000 nm) of the captured scene appears on the left panel in Figure 75 to verify the success of the operation immediately. If the Patient ID and Lesion Type text fields are empty, a popup will remind it and ensure correct data storage.

6.14. Acquisition set-up validation

The thesis considered some crucial metrics to guarantee the system's repeatability. Consequently, the main objective of this analysis is to assess the system's ability to acquire the same scene under comparable conditions with similar results. This procedure guarantees that the tool is not heavily dependent on uncontrolled variables and that the information faithfully represents the scene's attributes at the capture time.

The experimentations gathered images by repeating the capture procedure under the same lighting conditions. To be sure of acquiring the same spatial window, the procedure employed the reposition command, which is present within the GUI (Figure 76). When performing a traditional acquisition, once centred on the target, the system moves up 1 cm and then down by capturing 2 cm frames. In this way, ten images of the same target object were collected through repeated consecutive acquisitions, and from these were extrapolated specific indices and comparison graphs.

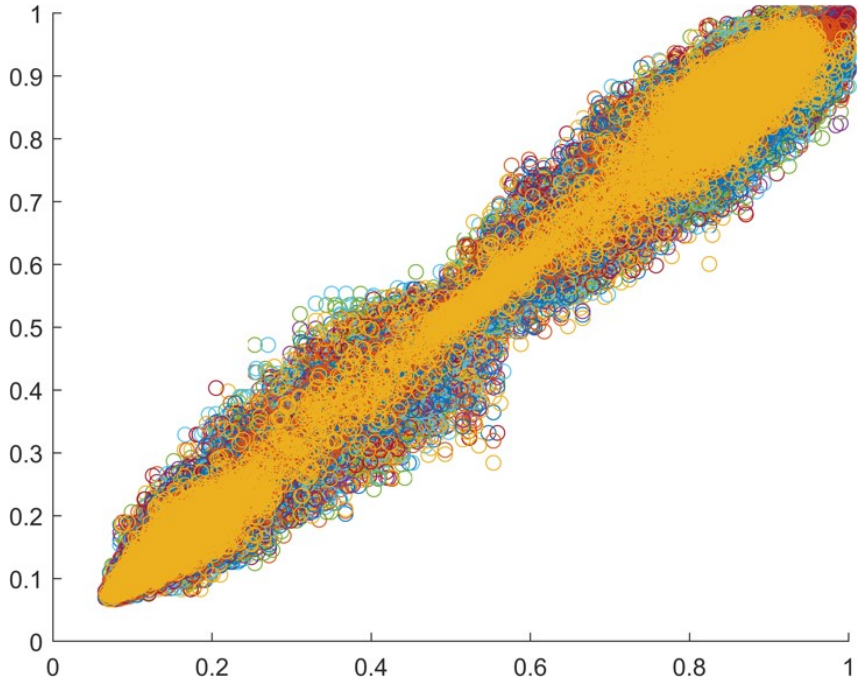


Figure 76. Repeatability analysis through voxel plot

The graph displays the differences between these images in a scatter plot, in which all the volumetric picture element (i.e., voxels) values of each of the two hyperspectral images compared appear on the two axes. The voxel value represents the reflectance of light in each pixel of the hyperspectral image at a given wavelength[38,99]. A scatter plot can be an effective tool to visualise the degree of correlation between the two variables placed on the axes. Ideally, the scatter plot should be a bisector line between the positive half-axis of the abscissas and the positive half-axis of the ordinates, which would indicate that each corresponding voxel pair between the two images contains the same information. The comparisons for all possible image combinations resulted in Figure 76's plot. The tendency of the points of the scatter plot, which identify the values assumed by the corresponding voxels of the two different images, along the bisector testifies to the high degree of correlation between two successive acquisitions and is, therefore, an index of repeatability 16. Another index is the Relative Percentage Difference (RPD), calculated as shown in the Equation 27 below:

$$\boldsymbol{RPD}\ (\%) = \mathbf{200} \times \frac{|\boldsymbol{R1} - \boldsymbol{R2}|}{\boldsymbol{R1} + \boldsymbol{R2}} \qquad \text{Equation 27}$$

R1 and R2 correspond to the compared HSIs and measure the percentage of how much one differs from the other. Lower values of RPD represent significant similarity. This manuscript compared all possible combinations and derived the average value. The calculated average RPD is 12.45%, again giving us a remarkable degree of repeatability[38,97,99]. The last measure considered is the Structural Similarity Index Method (SSIM). It is a well-known quality metric used to measure the resemblance between two images and is related to the quality perception of the human visual system (HVS). Instead of traditional error summation methods, SSIM models any image distortion as a combination of correlation loss, luminance distortion and contrast distortion[97]. The similarity index has a decimal value between 0 and 1; value 1 indicates two identical images, and value 0 indicates no similarity. This investigation obtained an average SSIM of 0.8725.

6.15. Ending remarks

Here, we presented a hyperspectral acquisition system engineered to gather diagnostic clinical data concerning skin cancer. It is enhanced by a linear synchronous motion, an appropriate illumination system, a 3D-printed circular crown containing targeting and distancing emitting diodes, and software modules supported by open-source packages. The hyperspectral system enables image collection with any GigE-compliant hyperspectral pushbroom camera. Furthermore, the investigation validated the architecture to check synchronisation between motor and camera frame

rate, calibration, and capturing repeatability. In the future, the research aims to collect an online database of clinical hyperspectral images.

In conclusion, the main contribution of this work is to serve as a guide for any research group working on hyperspectral technologies. All the sections report details to accurately capture spectral information and techniques to validate the correct operation of the system. First, the whole system works with any GenICam protocol-compliant camera. Secondly, the thesis operated cheap and promptly available hardware and open-source software to enable research groups to work with hyperspectral systems most efficiently. Indeed, all software modules used in this development are open source, allowing high flexibility and representing a lower-cost approach compared to market solutions.

6.16. Parallel classification pipelines for skin cancer detection exploiting hyperspectral imaging on hybrid systems

The second investigation in this chapter concerns a parallel-computing implementation of an HS dermatologic classification framework based on K-means and SVM algorithms and snapshot HS cameras to achieve the first automatic and real-time in-situ PSL identification of this doctoral thesis. This research represented the starting point for evaluating DL architectures applied to HS images.

Previous studies of groups who participated in the investigations of this doctoral thesis described the HS in-situ acquisition system[38,99]. The introductory analysis validated the hypothesis of HSI exploitation to differentiate between PSLs through pixel-wise supervised classification. Regardless, it consisted of a classification framework to differentiate between malignant and benign PSLs, but without considering the computational aspects of the proposed algorithm.

This research presents a variation of the classification framework aiming to differentiate between malignant, benign, and atypical PSLs. Furthermore, it comprehensively reports the implementation and parallelisation to obtain real-time performance. This real-time diagnostic tool could assist dermatologists in differentiating PSLs during routine clinical practice, delivering more diagnostic information regarding NMSC than other retail systems that only discriminate between melanoma and non-melanoma.

6.17. Hyperspectral dermatologic classification framework

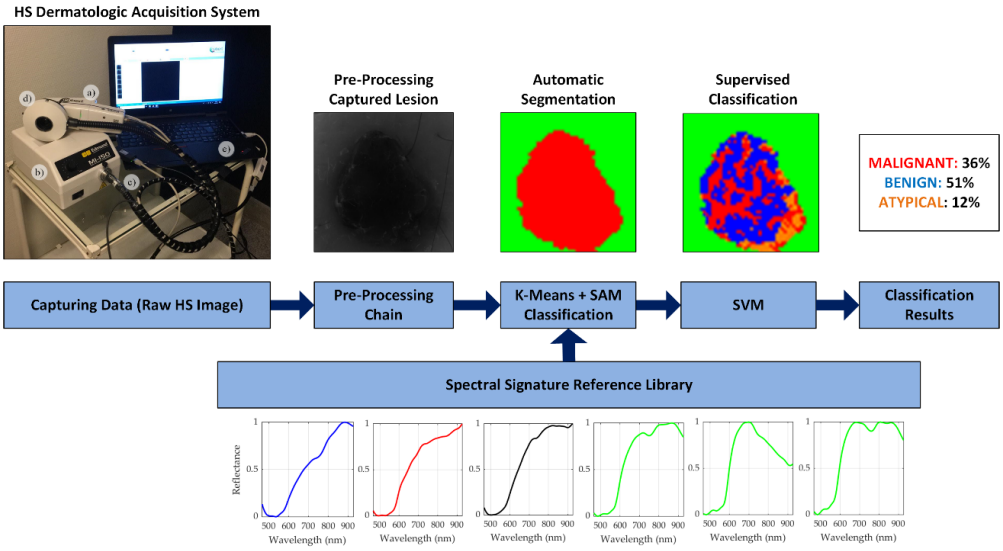


Figure 77. Block diagram of the HS dermatologic classification framework (pre-processing, automatic segmentation, and supervised classification) and HS dermatologic acquisition system. HS dermatologic acquisition system is composed by: (a) HS snapshot camera; (b) QTH (Quartz-Tungsten Halogen) source light; (c) Fiber optic ring light guide; (d) Skin contact part attached to the ring light; (e) Laptop with the acquisition software installed. Spectral signature reference library is composed of six spectral signatures: benign, malignant, and atypical pigmented skin lesion spectral signatures in blue, red, and black colors respectively, and three different skin spectral signatures in green color [38,99]

The HS dermatologic classification framework constitutes three main steps: pre-processing, automatic PSL segmentation, and supervised classification. Figure 77 shows a block diagram of this framework. The pre-processing consists of what we reported in Section 6.3. After pre-processing, the other steps in the chain automatically segment the resulting image, providing healthy and PSL pixel labelling. This discrimination operates a spectral signature reference library, which contains the following[38,99]:

- Three spectral signatures of benign, malignant and atypical PSL This section highlighted those in blue, red and black colours, respectively, in Figure 77
- Three skin spectral signatures, in green colour in Figure 77

The spectral reference for each PSL class is the average of all the labelled spectral signatures. Furthermore, the K-means employed the

results using the Silhouette, Calinski Harabasz, and Davies Bouldin methods to compute the number of clusters[38,99]. The research split the normal class into three groups due to various skin types, particularly emphasised in the NIR (Section 2.5).

Eventually, a supervised classifier categorises into benign, malignant, and atypical the pixels previously identified as a lesion.

6.18. Pre-processing chain

The pre-processing chain described in this chapter consists of four stages: calibration, wavelengths removal, noise filtering and normalisation. Algorithm 7 contains the pseudo-code of the pre-processing chain, where Y indicates an HS image with n pixels and b bands (i.e., wavelengths).

Algorithm 7. Pre-processing chain[19]

Input: Y → Hyperspectral image with n pixels and b bands
D_{ref} →Dark reference
W_{ref} →White reference
N →number of neighbours

1. Hyperspectral image Y acquisition
2. *Stage 1.1: Image calibration*
3. **for** *i*=0 to *n-1* **do**:
4. **for** *j*=0 to *b-1* **do**:
5. $Y_{calibrated}(i,j) = \frac{Y(i,j)-D_{ref}}{W_{ref}-D_{ref}}$;
6. **end**
7. **end**
8. *Stage 1.2: Extreme bands removal*
9. Remove the first 4 and last 5 noisy bands
10. *Stage 1.3: Smooth filtering*
11. **for** *i*=0 to *n-1* **do**:
12. **for** *j*=1 to *b-N-1* **do**:
13. $sum = 0$;
14. **for** *x*=0 to *b-1* **do**:
15. $sum += Y_{calibrated}(i, j + x)$;
16. **end**
17. $Y_{calibrated}(i,j) = sum/b$;
18. **end**
19. **end**
20. *Stage 1.4: Normalization*
21. **for** *i*=0 to *n-1* **do**:
22. Find the *max* and *min* values over the bands
23. **for** *j*=0 to *b-1* **do**:
24. $Y_{calibrated}(i,j) = \frac{Y_{calibrated}(i,j)-min}{max-min}$;
25. **end**
26. **end**

Output: $Y_{calibrated}(i,j)$

In Algorithm 7, lines from 3 to 7 perform the calibration stage. As described in Section 6.3, W_{ref} and D_{ref} derive from moments before data acquisition operating the same illumination conditions. The resulting calibrated image ($Y_{calibrated}$) results from the line 5's equation.

After the calibration stage, we remove the first four and last five bands due to the HS detector's inadequate response. Line 9 performs this procedure, and the final spectral signature consists of 116 wavelengths. Furthermore, the procedure reduced noise through smooth filtering. Line 11 loop performs the filtering for each HS image pixel. Line 12 contains the loop declaration where N is the value of the neighbours previously chosen. For this experiment, N equals five.

Finally, a normalisation process between 0 and 1 to each pixel homogenises spectral amplitudes. Lines 21 to 26 perform the normalisation process.

6.19. Unsupervised PSL segmentation

K-means and SAM algorithms automatically segment the pre-processed HS image into normal and PSL pixels. The K-means algorithm divides the hypercube into k different clusters, and the optimal k equals three. Nonetheless, after performing the three-way clustering, we generated a two-class segmentation map to identify the skin and the lesion. The SAM algorithm produces the two-classes segmentation map, which compares the centroid from each cluster with a spectral signature from the reference library. The library contains the six different spectral signatures earlier described.

Algorithm 8 reports the unsupervised segmentation comprising K-means and SAM. Line 2 initialises the actual_centroids variable to select the centroids used by the K-means algorithm. This variable starts with k stochastic different HS pixels from the input image Y. The error variable in line 3 is the average of the absolute values of the difference between centroids. The error represents a constraint for algorithmic convergence. The algorithm's main loop from lines 6 to 13 computes the distances between a specific pixel and the centroids with an iterative procedure. The distance operates the Euclidean metric, and each pixel will belong to a particular cluster when it reaches the minimum distance. Using actual_centroids and previous_centroids variables allows for analysing the variation from the previous iteration. This loop finishes when the error becomes lower than the established threshold or after a maximum number of iterations.

Eventually, the SAM algorithm compares each cluster with the reference library and returns the binary segmentation map. Lines from 14 to 20 correspond to the similarity evaluation.

Algorithm 8. Automatic segmentation [19]

Input: *Y, k,* $threshold$, MAX_ITER, *HUGE_VAL*

Stage 2.1: K-means initialization
Randomly choose *k* pixels as *actual_centroids*
$error = HUGE\_VAL;$
$iter = 0;$
Stage 2.2: K-means clustering
while $error < threshold$ && $iter < MAX\_ITER$ **do**:
  Compute the distance between pixels and centroids
  Clusters update
  $previous\_centroids = actual\_centroids;$
  Update *actual_centroids*
  Compute error between *actual_centroids* and *previous_centroids*
  $iter++;$
end
for *i*=0 to *k-1* **do**:
  for *j*=0 to $n_{ref}-1$ **do**:
    dist(j) = compute SAM between *actual_centroids(i)* and *ref(j)*
  end
  h = find the index of the minimum value of *dist*
  Assign to the i-th cluster the same class as *ref(h)*
end

Output: PSL pixels

6.20. Supervised classification

This research operated an SVM to classify the pixels identified as PSL by the previous step. The SVM algorithm aims to find the best hyperplane to separate different data and compute the probability of belonging to each class of study[57].

This study selected the sigmoid kernel after comparing the performance results with others after hyperparameters optimisation. Algorithm 9 illustrates the pseudo-code of the supervised classification where pix_no_skin contains the lesion pixels obtained from the previous stage.

Algorithm 9. Supervised classification [19]

Input: *pix_no_skin*, $n_{pix\_no\_skin}$, *class*, n_{sv}, *sv*, *epsilon*

Stage 3.1: SVM data preparation
for *i*=0 to $n_{pix\_no\_skin}-1$ **do**:
  Stage 3.2: SVM distance computation
  for *j*=0 to $n_{sv}-1$ **do**:
    $prod = sv(j) * pix\_no\_skin(i);$
    $dist(j) = \tanh(slope * prod + intercept)$
  end
  Stage 3.3: SVM binary classification
  for *j*=0 to *class-2* **do**:
    for *z=j* to *class-1* **do**:
      Solve binary classification problem between class *j* and class *z*

12. **end**
13. **end**
14. *Stage 3.4: SVM multiclass probability*
15. $Pc1 = \cdots = Pcclass = \frac{1}{class}$;
16. Computing the matrix *Qp* using the binary probabilities
17. **for** *z=0* to *class-1* **do**:
18. **for** *iter*=0 to 99 **do**:
19. **if** $Pc_z - Pc_z$_prev < epsilon **do**:
20. **break**;
21. **end**
22. Update multiclass probability of the i-th pixel to belong to class z
23. **end**
24. **end**
25. Compute class with maximum probability for the i-th pixel
26. **end**
27. Update similarity evaluation labels with SVM results

Output: Probabilities class.

The pseudo-code comprises four main steps:

1. Data preparation
2. Distance computation
3. Binary classification
4. Multi-class probability

The iterative procedure from lines 2 to 26 computes the probability of the i-th pixel belonging to a specific class. The kernel evaluates this probability, computing the distance. Line 5 multiplies the pixel by a support vector, and line 6 returns the distance using the kernel's parameters: slope and intercept.

The next stage performs the binary classification based on the probability mentioned earlier. Ultimately, the multi-class probability comes from lines 15 to 24, utilising the probabilities obtained in the previous stage. This process ends when the value of the previous iteration is under a certain threshold or if the number of iterations reaches 100. When one of these two conditions takes place, we obtain the multi-class probabilities of the pixel.

In summary, all the abovementioned steps feature high computational complexity, thus preventing real-time processing. Consequently, the exploration of parallel architectures is mandatory to provide an efficient instrument for clinical practice.

6.21. Parallel classification pipelines

In the following sections, this research explores various parallel strategies targeting multicore and many-core hardware to decrease the processing time of the serial pipeline.

The first step comprised writing the classification framework (pre-processing, K-means, and SVM) in C language. This serial code represents a basis for the parallel versions that integrate the OpenMP and the CUDA frameworks for the multicore and many-core philosophy, respectively. Hereafter, the thesis evaluates miscellaneous parallel classification pipelines partially written in OpenMP or CUDA[78].

6.22. OpenMP overview

This section will briefly review the OpenMP framework which did not receive a dedicated chapter of the thesis since all investigations mainly concerned CUDA programming from the hardware acceleration perspective.

OpenMP is a programming model and API for parallel programming in C, C++, and Fortran. It allows developers to specify parallel regions of code that can be executed concurrently on shared memory systems, such as multi-core processors or symmetric multiprocessor systems.

One of the key features of OpenMP is the use of pragma directives to specify parallelism in the code. The *omp for* directive is used to specify a loop that should be executed in parallel.

Here, are two examples of the *omp for* directive. The following code demonstrates how to parallelize a loop using the *omp for* directive:

Algorithm 10. OpenMP first example

```
1. #pragma omp parallel for
2.  for (int i = 0; i<N; i++)
3.  {
4.     #loop body
5.  }
```

In Algorithm 10, the loop will be executed in parallel by multiple threads, with each thread responsible for iterating over a subset of the loop iterations.

The following Algorithm 11 demonstrates how to specify a loop reduction using the omp for directive:

Algorithm 11. OpenMP first example

```
1. int sum = 0;
2. #pragma omp parallel for reduction (+:sum)
3.  for (int i = 0; i<N; i++)
```

```
4.  {
5.      sum += data[i];
6.  }
```

In this example, the *reduction* clause specifies that the loop variable *sum* should be reduced using the + operator. This means that the value of sum will be updated by each thread in a thread-safe manner, resulting in the final value being the sum of all iterations.

6.23. Parallel pre-processing versions

Serial code profiling reported filtering and normalisation as the most time-consuming phases of the pre-processing. Accordinlgy, this research accelerated only these two code portions. Each thread performs the filtering and normalisation of a single HS pixel. A for loop that iterates over the number of pixels, and the pragma omp parallel directive elaborates the iterations concurrently. We declared the loop variables private while we shared the HS image among all threads.

The same parallelisation occurred in CUDA. After image calibration and band removal, data are allocated and transferred to the device's global memory. The transferred data also consists of the reduced image and the array storing groups of five contiguous wavelengths for each pixel to avoid data overwrite during the moving average algorithm. A CUDA kernel filters through a grid containing as many threads as the number of pixels. The grid includes blocks of 32 threads, which is the warp definition given by Nvidia (Section 4.3). If the number of pixels is not an integer multiple of 32, the last block will contain some threads that do not relate to a pixel. In this case, these threads do not perform any computation. These inactive threads do not slow down the computation because their number is negligible compared to the total number of pixels. Another kernel, with the same grid and block parameters, computes each pixel's maximum and minimum values across the bands. These values then take part in the normalisation step, performed by a different kernel. The normalised image overwrites the original, initially transferred to the device's global memory.

The result of the pre-processing is transferred back to the host memory only if the K-means uses serial or OpenMP-accelerated processing. Otherwise, the result remains in the device's memory. The flowchart of the CUDA pre-processing is in Figure 78.

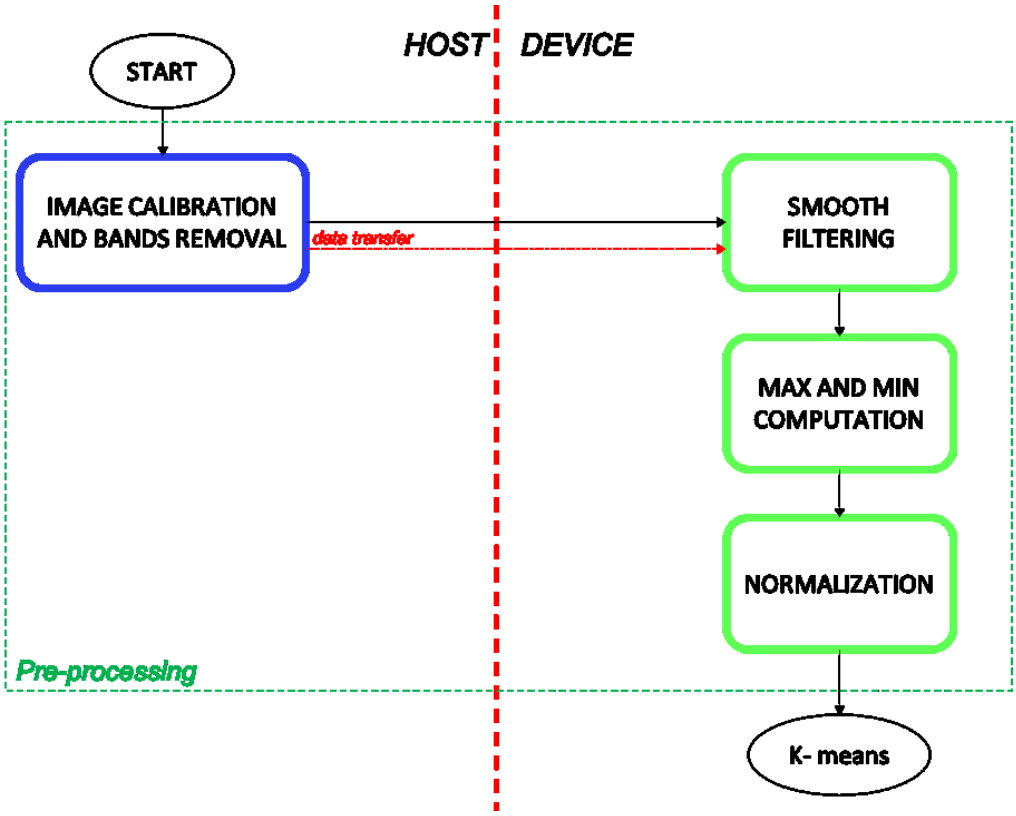


Figure 78. Parallel HS image pre-processing flowchart [19]

6.24. Parallel K-Means versions

The K-means most time-consuming part is the distance computation between each pixel and each centroid. The number of iterations equals the number of pixels times the number of clusters.

The other operations have a negligible computational cost when performed on a serial processor. Hence, the research parallelised only the distance computation using OpenMP and CUDA.

The pragma omp parallel for directive exists before the for loop iterating over the pixels. Again, the loop variables are declared private, and the HS image is shared. Moreover, also the centroids are shared among the pixels.

The CUDA version adopts a different strategy. Indeed, all the steps happen on the device to minimise data transfers between host and device memories. Figure 79 displays the flowchart of this parallel version.

The first task the GPU performs is centroid initialisation, which consists of copying the values of the selected pixels into the centroids. A kernel whose threads number equals the number of clusters operates in this step.

The error computation happens between the device and the host. At first, a kernel computes the difference between the actual and the previous centroids. Then, the cublasSasum function sums the absolute values of these differences. This function directly transfers the output to the host, where the division by the number of clusters operates a serial thread.

At this point, the iterative K-means process starts on the host. The following steps repeat until the error converges to a fixed threshold or they attain a maximum number of iterations.

The first step concerns the distance computation performed on the device by a kernel, having as many threads as the number of pixels. In particular, each thread simultaneously computes the distance between the pixels and the centroids.

Then, we update the clusters and the centroids with two different kernels. The former provides a pixel-wise parallelisation since each thread finds the nearest centroids for each pixel.

The latter includes as many threads as the number of clusters to perform the update. At this point, we evaluate the error.

Once the condition of the while loop is false, the flow continues with the similarity evaluation step, which assigns biological meaning to each cluster.

The overall computation involves a restricted number of data, allowing efficient elaboration on the host device.

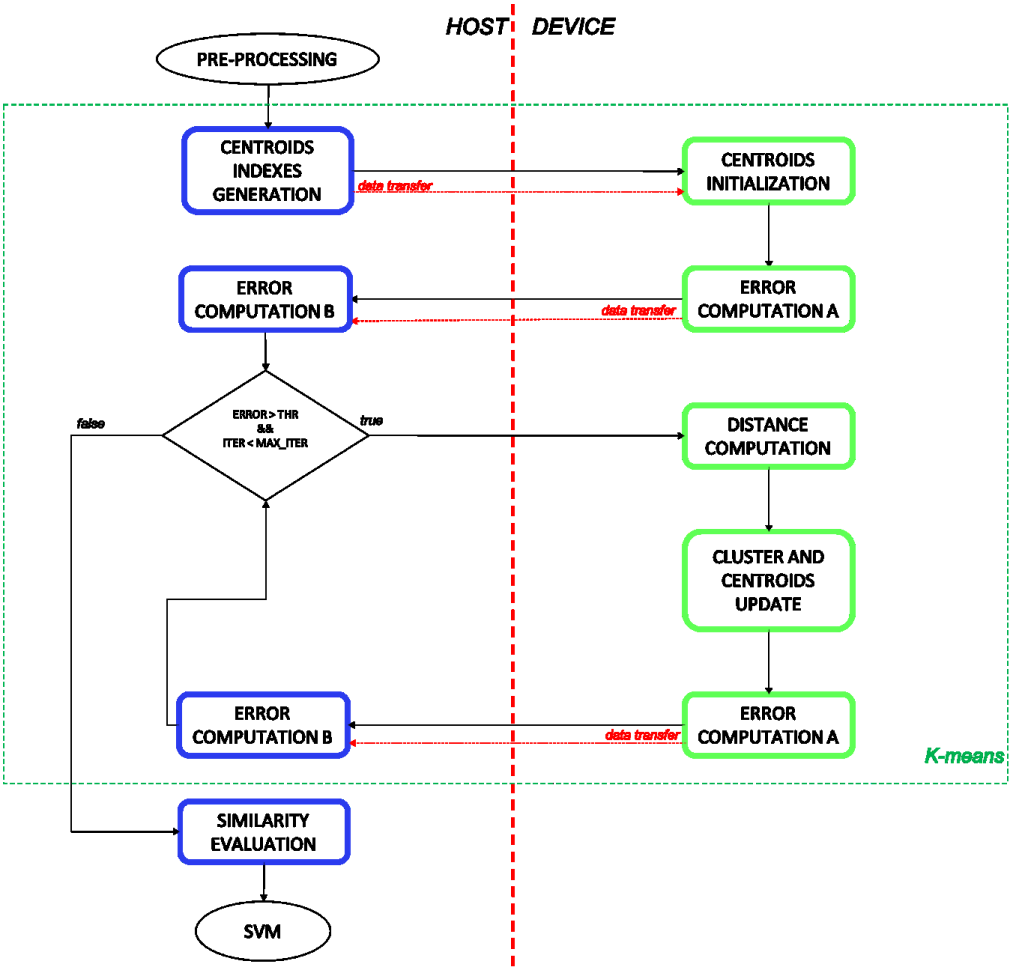


Figure 79. K-means parallel execution flowchart [19]

6.25. Parallel SVM versions

The SVM algorithm has three main steps: distance and non-linear function evaluation, binary classification, and multi-class probabilities computation.

The first phase is the most time-consuming part. We must stress that only a subset of the original HS image arrives at the SVM algorithm. The SVM training generated 9242 support vectors, which is higher than the number of pixels of each image. For this reason, the OpenMP version parallelised the for loop which iterates the support vectors. In particular, each thread performs the dot product between the assigned support vector and the pixel. Then, it applies the hyperbolic tangent to the product result after considering the slope and intercept values. In this case, the shared variables are the pixels to be classified and the support vectors, and the private variables are the loop indexes.

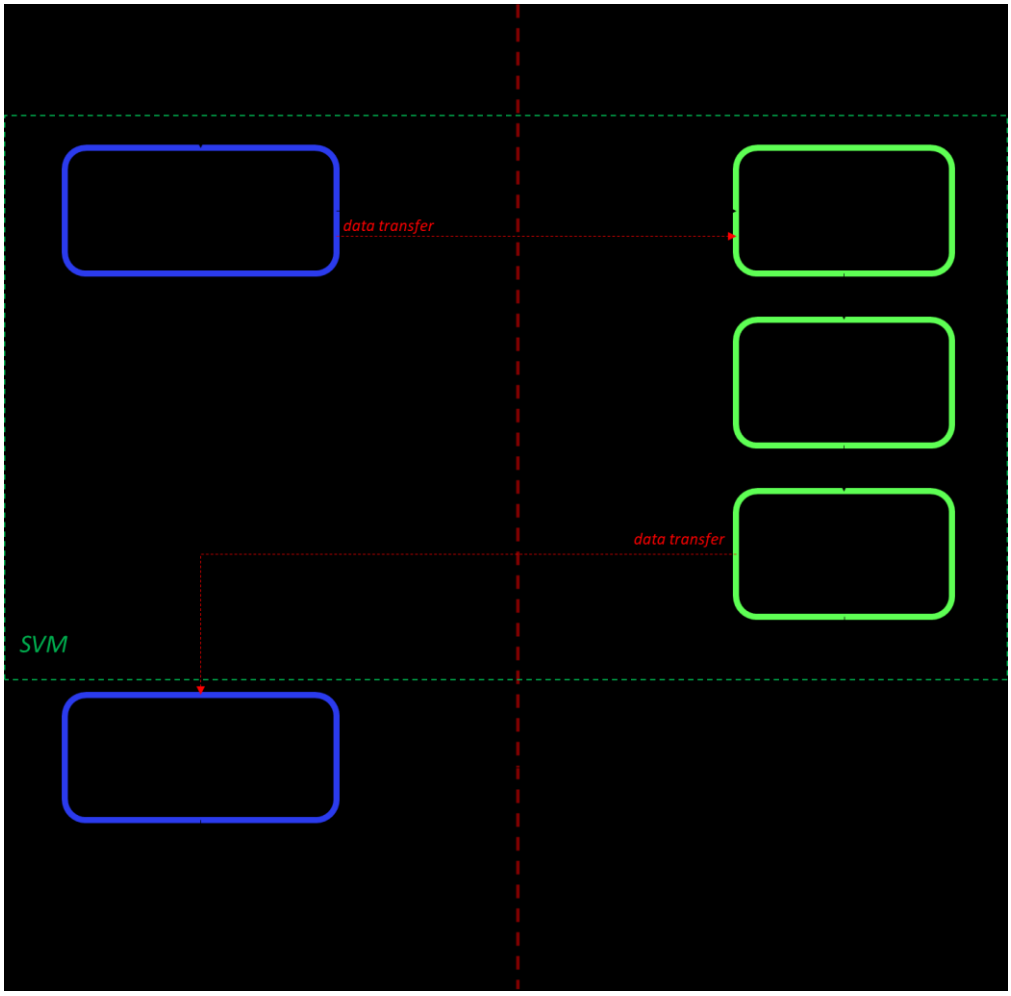


Figure 80. SVM parallel execution first and second versions' flowchart [19]

The research developed three different CUDA versions to find the most efficient one. Figure 80 shows the first and second arrangements' flowcharts.

The flow starts on the host, where we transfer the SVM model parameters and the pixels. The distance computation kernel computes the dot product between support vectors and pixels and evaluates the hyperbolic tangent. In this kernel, the number of threads is equal to the number of support vectors for the same reason explained in the OpenMP version. Again, each block contains 32 threads.

A different kernel evaluates the binary probability. The kernel's grid dimension represents the difference between the first and second SVM CUDA versions. In the first case, the number of threads equals the number of support vectors, whilst there is a single thread in the latter. The main reason for this choice is that binary probability computation is a very efficient task to be processed sequentially. The idea is to reproduce serial processing on the device, avoiding further memory transfers, even if the GPU working frequency is lower than the CPU one.

The last kernel computes the multi-class probabilities. In this case, the number of threads equals the number of classes: each thread evaluates the probability of each pixel belonging to a class. Then, the cublasIsamax function determines the class with the highest probability for each pixel. This function also transfers the output (i.e., the pixel labels) to the host.

The third CUDA version relates to the fact that binary classification performs very efficiently on serial processors. Therefore, this computation has been moved to the host side to evaluate whether transferring back data, performing the elaboration on the host or if a serial kernel is the best solution. Figure 81 illustrates this version. The kernels related to the distance computation and the evaluation of multi-class probabilities did not change compared to the previous CUDA versions.

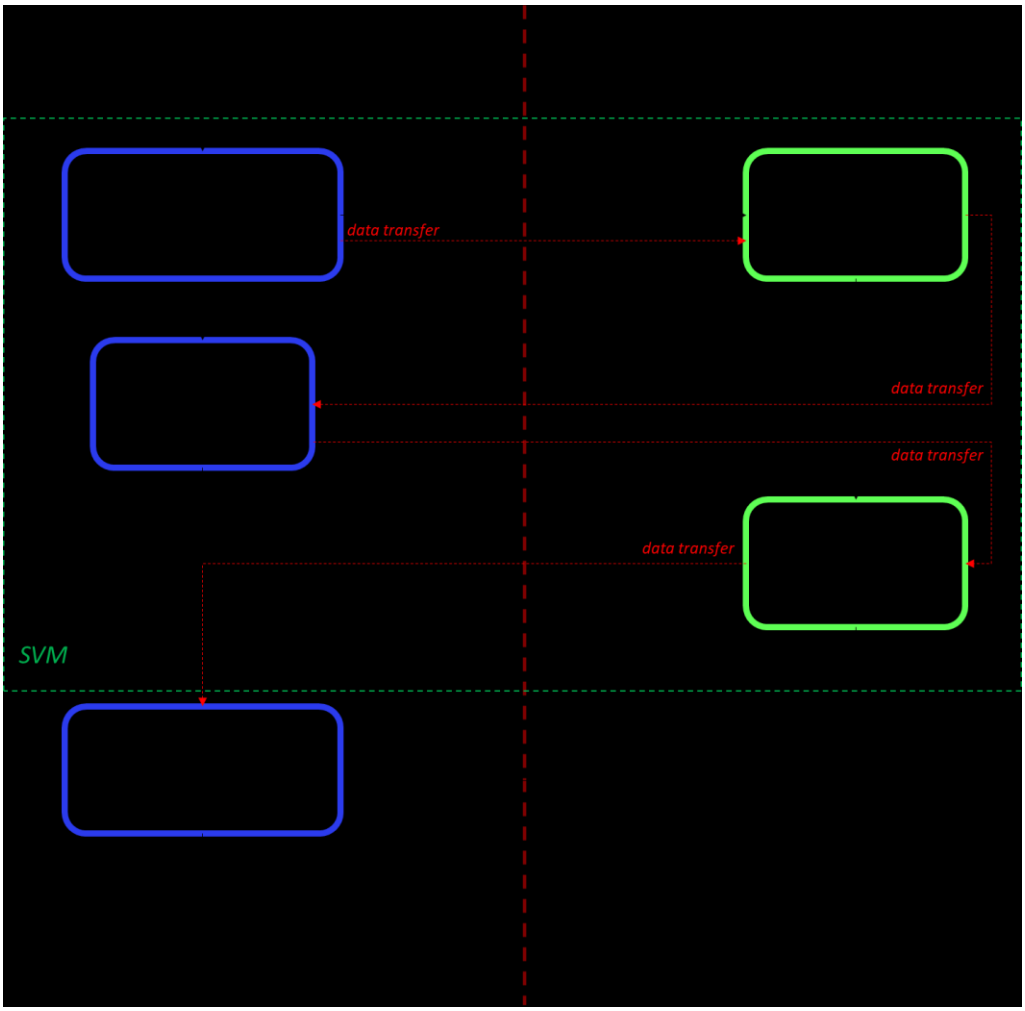


Figure 81. SVM parallel execution third version's flowchart [19]

6.26. Complete classification system

Table 14 presents the fifteen parallel versions developed in this work. The basic idea is to find the best configuration regarding processing time. The visual profiling reports that all the pre-processing versions provide equivalent performance, and we included all the versions for the final configuration assessment. In particular, even if the CUDA performance is similar to the serial one, it was decided to quantify if an initial data transfer can benefit the subsequent steps of the processing chain.

Table 14. Different versions of the classification framework, integrating the serial (S), OpenMP (O), and CUDA (C) codes of the single algorithms. C1, C2, and C3 refer to the three SVM CUDA versions [19]

	Pre-processing			**K-means**			**SVM**		
	S	**O**	**C**	**C**	**S**	**O**	**C1**	**C2**	**C3**
V1	×			×	×				
V2	×			×		×			
V3	×			×			×		

V4	×			×				×	
V5	×			×					×
V6		×		×	×				
V7		×		×		×			
V8		×		×			×		
V9		×		×				×	
V10		×		×					×
V11			×	×	×				
V12			×	×		×			
V13			×	×			×		
V14			×	×				×	
V15			×	×					×

On the other hand, concerning the K-means clustering, only the CUDA version has been included in the different complete systems since it vastly outperforms the serial and OpenMP processing. Accordingly, the speedup of the multicore and many-core K-means versions, compared to the serial processing, are about 1.5× and 6×, respectively.

Ultimately, all the SVM versions participate in the final system's integration. The research also developed a configuration (not included in Table 14) considering all the serial versions to compute the final speedup.

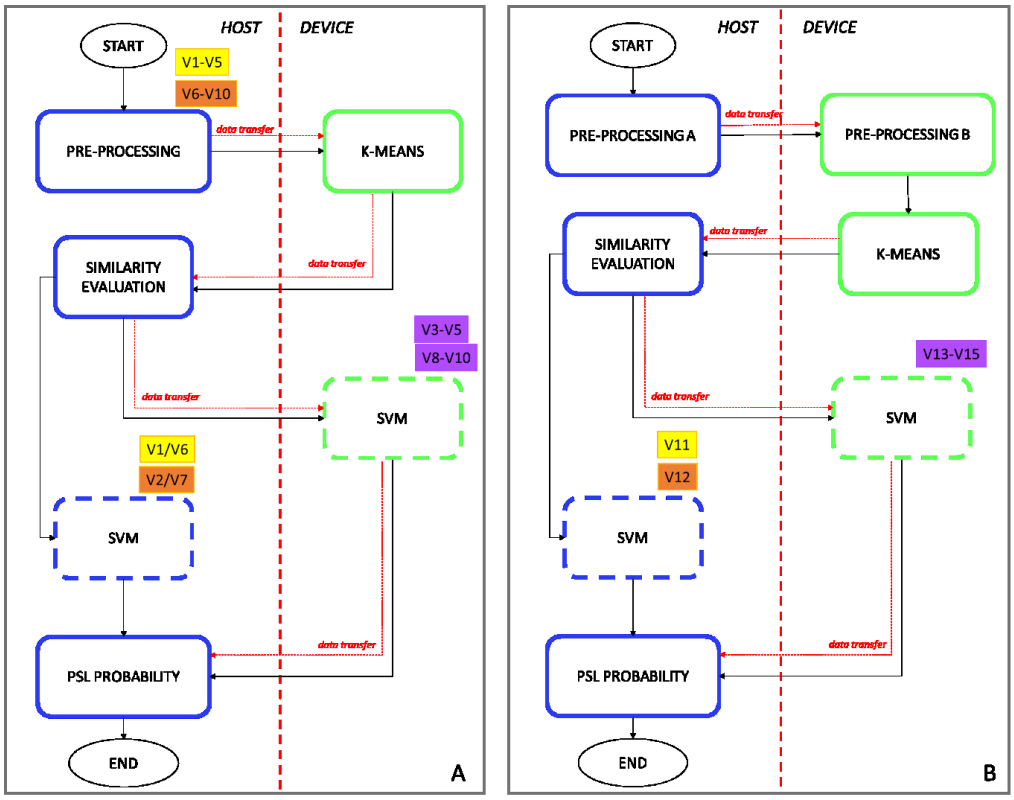


Figure 82. Comprehensive parallelisation flowchart [19]

Figure 82 illustrates the parallelisation flowcharts. The pre-processing happens on the CPU in the first ten versions. In all these cases, the pre-processed image goes to the device before the K-means execution.

6.27. Skin cancer classification performance

The genetic algorithm (GA)[57] operated the SVM's hyperparameters tuning. The methodology proposed a stratified patient assignment where the labelled data comprised three independent sets: training, validation, and test. The custom figure of merit (FoM) in Equation 28 evaluated the GA performance.

$$FoM = \frac{1}{2} \cdot \left(\sum_{\substack{i,j \\ i<j}}^{n} \frac{ACC_i + ACC_j}{|ACC_i - ACC_j| + 1} \right) \cdot \binom{n}{2}^{-1} \quad \text{Equation 28}$$

$$FNRc = \frac{FN_i}{P} \quad \text{Equation 29}$$

Eventually, the false negative rate per class (FNRc) assessed the results obtained for the optimised classifier. Equation 29 shows the mathematical expression of the FNRc, where FNi is the number of false negatives in the i-th class and P is the total number of positive samples.

Figure 83 shows the FNRc results of each HS test image. These results indicate the necessity of increasing the HS skin database to include high inter-patient data variability.

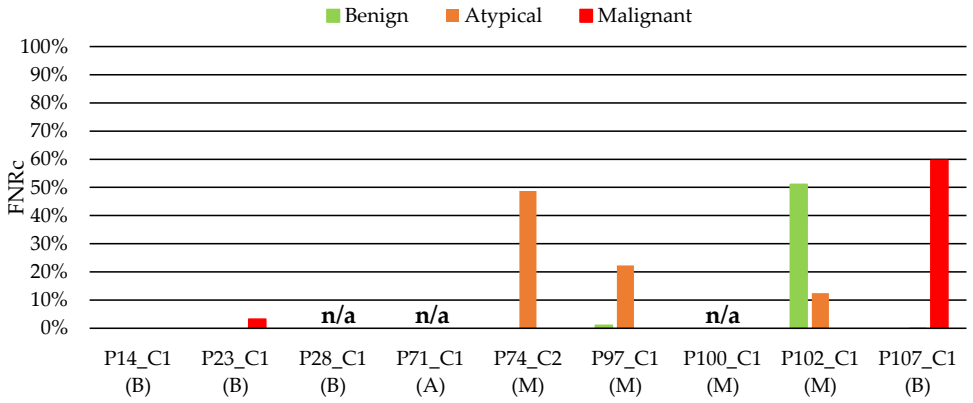


Figure 83. Classification FNRc results per each HS image obtained with the SVM Sigmoid classifier. (A) Validation classification results. (B) Test classification results. Below each patient ID, the correct diagnosis of the PSL is presented. B: Benign; A: Atypical; M: Malignant [19]

Figure 84 shows the processing time of the complete HS dermatologic classification framework using the test set implemented in MATLAB. These outcomes come from an Intel i7-4790K with a working frequency of 4.00 GHz and 8GB of RAM.

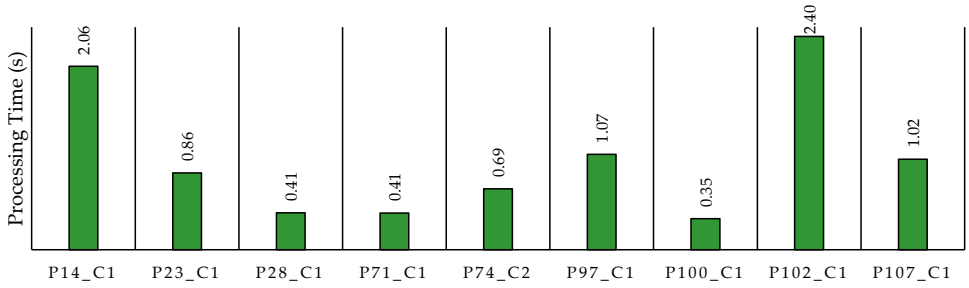


Figure 84. Processing time (in seconds) of the MATLAB execution for each HS image [19]

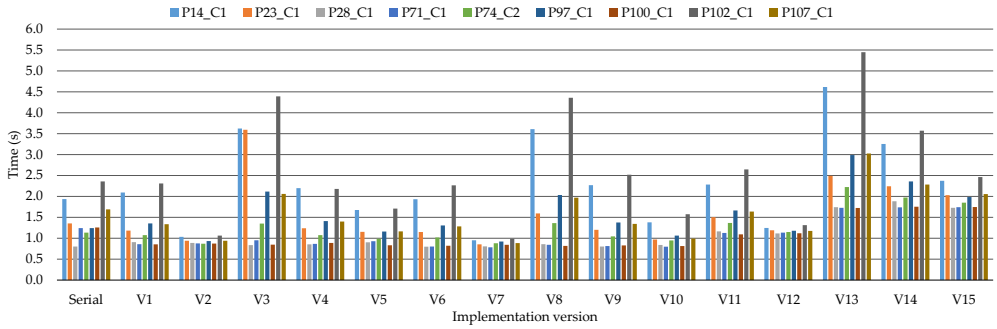
6.28. Real-Time elaboration

The acquisition system takes approximately one second to capture an image with 50 × 50 pixels and 125 bands. As we can appreciate from the results mentioned in the previous section, MATLAB implementation cannot always guarantee real-time processing.

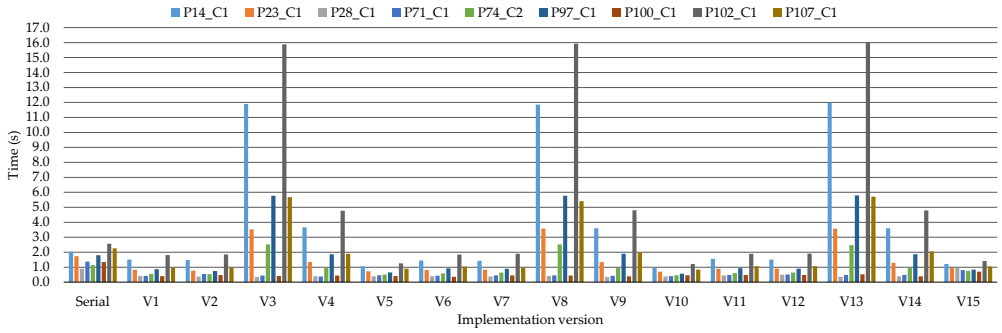
This research operated the first two test systems from Section 4.8 to assess the parallel code performance[19].

All the code versions have used Microsoft Visual Studio 2019 under Microsoft Windows 10. For all the versions, suitable compiler options generated an executable code optimised for processing speed.

The processing times report the mean of five different executions. For the GPU versions, we also include the data transfer time. Figure 85 shows the processing times for the described test systems using each HS image of the test set. The figure reports that only some of the versions are real-time compliant.



(A)



(B)

Figure 85. Processing times (in seconds) of the serial and parallel versions using the (A) TS1 and (B) TS2 [19]

6.29. Comparison and discussion

Concerning the parallel processing times, the time quantification results vary on two main factors: the number of K-means iterations and the number of pixels to be classified by the SVM. Remarkably, this last factor significantly changes among the images, and it is nothing but the lesion's area.

Concerning the pre-processing step of all the images, the OpenMP elaboration consistently outperforms the CUDA on TS1. The same trend exists in most of the cases on TS2.

Efficient parallelisation of the filtering and normalisation steps adopting a multicore approach is preferable to transferring the image to perform the pre-processing on a GPU device. Nonetheless, efficient pre-processing parallelisation only marginally impacts the final classification time.

Since all the parallel versions include the K-means algorithm developed in CUDA, this section only discusses the impact of the different SVM versions on the total processing time.

If we consider the TS1, whether the pre-processing happens by exploiting OpenMP (V6-V10) or CUDA (V11-V15), the best SVM version is the multicore one (V7 and V12). Nonetheless, the Intel i9 CPU classification, with twenty cores working at a high frequency, provides better performance than the elaboration on the device, which also requires a data transfer. Moreover, in this last case, the computational load is insufficient to exploit the GPU cores efficiently.

On the other hand, if the pre-processing happens in serial, the V2 version is the best solution. Finally, V2 and V7 are the two best solutions. However, only the V7 is always real-time compliant.

Comparing the performance of the two test systems and considering the images where the SVM is not performed (P28_C1, P71_C1, P100_C1), TS2 is always faster than TS1. The Tesla K40 GPU features a lower processing time on the K-means clustering than the RTX 2080. The former

board does not manage the graphical context of the operating system and can use all the resources to perform the computation. The latter is a standard GPU that shares resources between graphical context management and computation[19].

6.30. Conclusions

This research presented a parallel classification framework based on HSI exploiting the K-means and the SVM algorithms to perform an automatic in-situ PSL identification. The framework used an in-vivo dataset, and the algorithms' parameters tuning happened in MATLAB for later implementation of the processing framework on HPC platforms.

Several parallel versions, exploiting multicore and many-core technologies, have been developed to ensure a real-time classification.

This preliminary study demonstrated the potential use of HSI technology to assist dermatologists in the discrimination of different types of PSLs. However, additional research must occur to validate and improve the results obtained before being used during routine clinical practice using a real-time and non-invasive handheld device. Notably, a multicenter clinical trial with more patients and samples in the database will be necessary to validate the proposed approach further.

6.31. Deep convolutional Generative Adversarial Networks to enhance Artificial Intelligence for skin cancer applications

The third breakdown in this thesis proposes a deep convolutional GAN (DCGAN) to generate synthetic HS epidermal lesion images employing a small dataset. This investigation is crucial concerning the challenges highlighted in Chapters 2 and 3, and this chapter's literature review.

The investigation assessed the GAN by operating the synthetic data to train a ResNet-18, which classifies the original training data. Furthermore, the research evaluated the performances regarding the Frechèt inception distance (FID)[102], and the metrics mentioned in Section 3.13.

In particular, the novel contributions proposed by this essay are as follows:

1. A DCGAN architecture extended to generate synthetic hyperspectral medical images
2. The adoption of state-of-the-art techniques such as transfer learning and label smoothing
3. The modification of the proposed DCGAN into a conditional network

4. The use of a ResNet-18 network to evaluate the similarity between *synthetic* and *real* datasets

This research employed the dataset, the pre-processing, and the taxonomy this thesis described in Chapter 2 concerning the hyperspectral skin cancer assessment.

6.32. Deep convolutional Generative Adversarial Networks

Goodfellow et al. proposed the original GAN in 2014[74], and it leans on two subnetworks: a generator (G) and a discriminator (D). Figure 38 from Chapter 3 depicted the fundamental concept behind a GAN.

The generator G inputs a latent space vector z from a standard gaussian distribution and produces a sample G(z). This sample represents the mapping from the latent space z to the actual data space.

On the one hand, G optimises to estimate the training data distribution and generate synthetic samples with the same real data distribution.

On the other hand, discriminator D receives the synthetic data produced by G or a sample (x) from the real dataset as input. Accordingly, D estimates whether the sample came from the training data or G.

G and D play a *minimax game*. G tries to minimise the probability that D will predict its outputs as fake, while D tries to maximise its probability of correctly discriminating between real and fake samples.

Researchers proposed several network topologies to implement G and D and deep convolutional GANs[55,56,74,103,104]. Accordingly, deep CNNs emerged as stable and affordable architecture for synthetic image generation with promising results especially for medicine. Notably, G consists of transposed convolutional layers, whilst D addresses common convolutional layers.

Considering HS images, the conversion from z to the data space performed by G consists of creating synthetic HS images with the training images' exact spatial and spectral dimensions. Since this thesis employed the skin cancer dataset described in Section 2.7 for training, G should generate an image whose sizes are $50 \times 50 \times 116$.

Figure 39.a from Chapter 3 displayed the G architecture and the sizes adopted in this work for G. The deconvolutional layers from 1 to 6 anticipate a batch normalisation and the ReLU activation function. Finally, the last deconvolutional layer adopts the tanh as the activation function.

On the other hand, D receives as input an HS image with the exact size, $50 \times 50 \times 116$, and performs a binary classification to determine if the input image is real or fake. For this reason, this network addresses convolutional layers, and Figure 39.b illustrated its architecture. All the leaky ReLU functions adopt a negative slope equal to 0.2, and the sigmoid function characterises the final convolutional layer.

6.33. Transfer learning in GANs

Authors who proposed GANs architectures typically trained the framework adopting large datasets, such as ImageNet[58], which include thousands or even millions of images. Those datasets' dimensionality is enormous compared to the 76 HS images considered in this doctoral thesis. This study addressed the curse of dimensionality to ensure the correct approximation of the original data distribution (Section 2.12).

Researchers usually adopt transfer learning as the possible solution to overcome the issue. As described in Chapter 3 of this thesis, it consists of using a model previously optimised for a task whose dataset size was more significant. It becomes the starting point for tackling a new problem with smaller training sets. In this research, transfer learning consists of pretraining the GAN using RGB skin cancer images and using the obtained parameters as initialisation for the final model, which operates the HS dataset. Accordingly, we trained the initial model using the HAM10000 dataset[105], randomly selecting 5000 RGB images from the database. We resized the images to 50×50 pixels to have the same HS dataset spatial dimension. Likewise, we modified the output layer of G and the input layer of D to address 3 channels instead of 116.

We adopted the Adam[93] optimisation method for the backpropagation algorithm, with a learning rate of 0.0002 for both networks and a batch size of 128. The training elapsed after 100 epochs. Finally, we exploited a label-swapping technique to avoid discriminator overfitting, which would imply no learning for the generator network. Figure 86 displays some images taken for the original dataset and different images generated by the network.

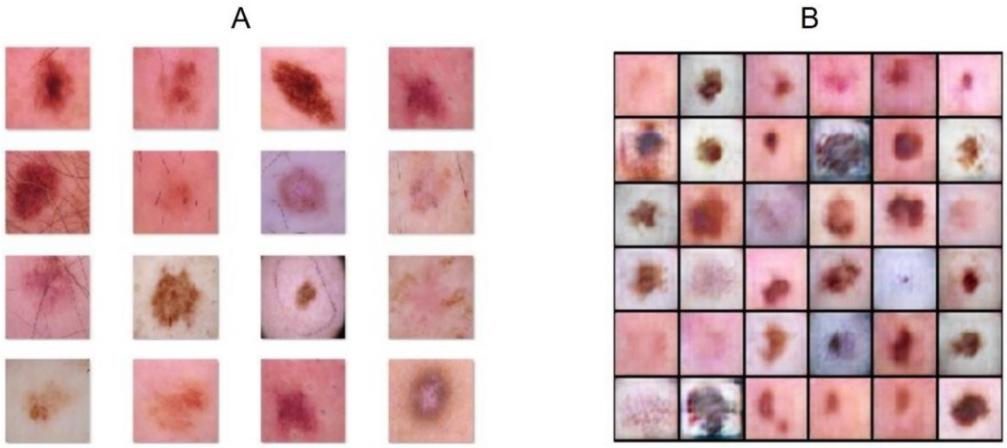


Figure 86. (**A**) Images taken from the training set. (**B**) Images generated by the architecture[56]

We transferred the network weights retrieved at the end of this training process to the architectures described in Figure 39. Notably, the

investigation only modified G's output layer and D's input layer. Consequently, the values obtained by the training with the RGB dataset initialised the weights related to the channels associated with the green, red, and blue wavelengths among the 116 channels of an HS image. The investigation initialised the remaining values stochastically and reduced the batch size to 2. Moreover, we changed the G's output layer size from 116 to 117. The additional channel generates the segmentation mask related to the synthetic image. The mask generation is of critical importance since it includes information that can be used in the training process of a generic deep segmentation network, highlighting the lesion contours.

Eventually, the proposed architecture transformed into a conditional GAN (cGAN)[106]. It means that G receives as input, together with the random noise vector, the class label-smoothed value to which the synthetic image should belong. Namely, the G can alternatively generate fake data related to the benign or malignant classes. The architecture of the proposed cGAN is in Figure 87.

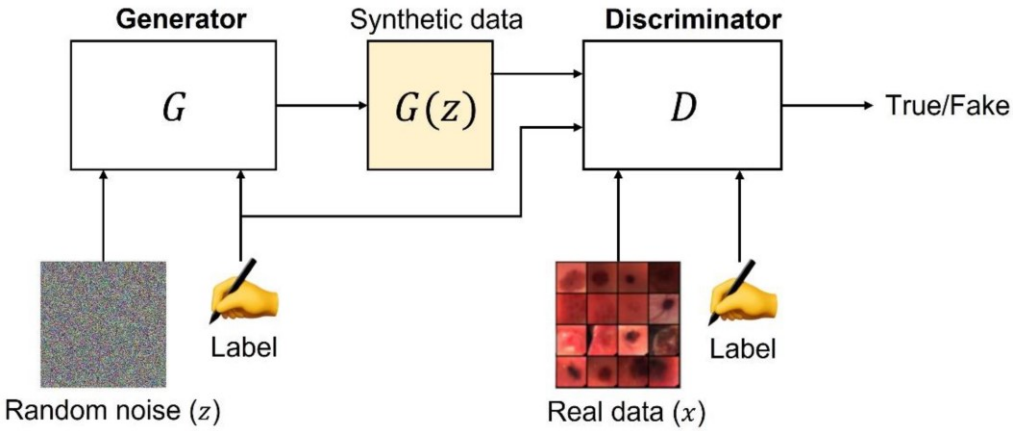


Figure 87. The cGAN architecture[56]

We trained the cGAN for 200 epochs. During training, different methods improved the quality of the synthetic images.

First, the weights of each layer scaled by a factor c according to the equalised learning rate rule[56] in Equation 30, where *InputChannels* represents the number of input channels to the considered layer.

$$c = \frac{\sqrt{2}}{\sqrt{InputChannels}} \qquad \text{Equation 30}$$

Consequently, the investigation implemented the two-time-scale update rule (TTUR)[75]. Particularly, it assigned the two networks different learning rate values. The learning rate of G was lower than that assigned to D. Thus, the analysis updated the weights related to G with more steps than the ones assigned to D to enhance the quality of the synthetic images.

To avoid D learning to discriminate real from fake images in a few training iterations, we swapped the labels for a random 5 % of the training data. Indeed, we treated some fake images as real and vice versa. Finally, we adopted L2 regularisation at 10^{-5} to reduce overfitting.

6.34. ResNet-18 classification

The investigation operated a ResNet-18, which we described in Chapter 3 of this manuscript, to measure real and synthetic HS data closeness. Namely, we trained the architecture only with synthetic HS images to classify the real epidermal lesions dataset. Therefore, we exploited overfitting as a measure to understand how well the synthetic data reproduces the real statistical distribution. This approach is innovative and not present in the literature and reveals if the synthetic dataset represents a significative description of the real dataset. In this case, overfitting should not be considered a negative effect. Indeed, overfitting on the synthetic dataset and obtaining good performance in the classification of the real dataset means that the GAN generalised the considered problem. Results reported in the subsequent sections highlight the trustworthiness of our generated HSIs.

The proposed approach is in Figure 88, where the blue arrows indicate that the set trains the model, while the green arrow denotes that the dataset tests the classification.

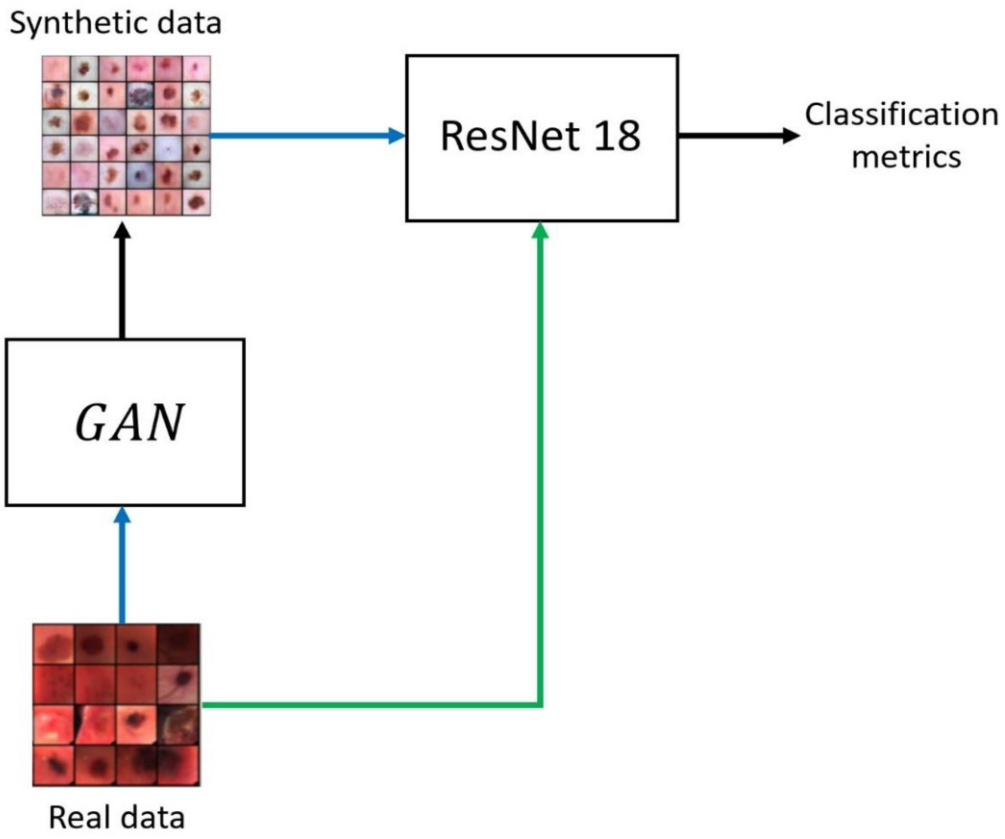


Figure 88. The proposed methodology to evaluate the similarity of the datasets. The blue arrows indicate that a set was used to train a model. The green arrow indicates that the set is classified by the network[56]

The ResNet-18 pre-trained on the ImageNet dataset, hence the input layer changed to consider an image of size $50 \times 50 \times 116$. Consequently, we optimised the network with the Adam gradient descent method in 50 epochs. The ResNet-18 was trained with 1000 synthetic images, while the test set included only authentic images.

6.35. Evaluation metrics

We employed several evaluation metrics to measure the performance of the developed generative framework. Frechèt inception distance (FID) is the state-of-the-art metric to assess the performance of a GAN in terms of the quality of the synthetic images[74,75,102]. The FID metric calculates the distance between the calculated feature vector for the authentic image and the generated image. Thus, a low value ensures that the two sets are similar. The FID is defined in Equation 31, where μ represents the mean value, Σ is the covariance matrix and Tr indicates the trace of a matrix. The subscripts 1 and 2 indicate the *real* and the *synthetic* images sets, respectively.

$$\mathbf{FID} = |\boldsymbol{\mu_1} - \boldsymbol{\mu_2}|^2{}_2 + \boldsymbol{Tr}(\boldsymbol{\Sigma_1} + \boldsymbol{\Sigma_2} + \mathbf{2} \times \sqrt{\boldsymbol{\Sigma_1} \times \boldsymbol{\Sigma_2}}$$ Equation 31

Concerning the ResNet-18 classification performance, we employed accuracy, precision, recall, and F1 score as described in Section 3.11 of this thesis.

6.36. Experimental results

The investigation assessed the synthetic dataset quality in two ways. On the one hand, we employed a gold standard metric in GANs, the FID[102]. On the other hand, we evaluated the accuracy, precision, recall, and F1 score of a ResNet-18, trained only with synthetic images, and then validated on the original dataset. Namely, we exploited overfitting to assess synthetic and authentic data distribution closeness. The generator produced a total of 1000 synthetic HSIs of skin lesions for these tests, equally balanced between benign and malignant classes.

6.37. Frechèt Inception Distance (FID)

The synthetic HS dataset generated by G obtained an FID value of 17.37. We computed the FID between the original data distribution and its augmented version to evaluate and compare different FID results. In particular, we simply horizontally flipped every HS image in the dataset. In this case, the investigation measured an 8.96 FID value. The two FIDs are close, thus indicating that the synthetic and the real data are similar.

6.38. ResNet-18 classification performance

We exploited the synthetic dataset to train a ResNet-18 tested on authentic HSIs. The ResNet-18 trained for 50 epochs with 1000 generated synthetic images. The network achieved 100% accuracy on the training set, thus overfitting it. Consequently, we used the architecture network to classify all the images in the real dataset.

Table 15. ResNet18 real HS dataset classification performance[56]

Metric	Value [%]
accuracy	84.21
precision	81.57
recall	86.11
F1 score	83.77

We report the performance of the ResNet-18 in the classification of the real images in Table 15, which clearly shows that ResNet-18 can correctly classify most real images. Accordingly, these results indicate that the synthetic and the original dataset are comparable. Accuracy, precision, recall, and F1 score are 85.52%, 83.50%, 85.65%, and 92.77%, respectively.

Nonetheless, it is worth noticing that the values should be kept distinct. The first results allow data leakage on purpose to assess the presence of overlap between the real and synthetic data distributions. On the other hand, the training on real data foresaw a train–test split to avoid the aforementioned data leakage and accurately assess generalisation capabilities of the model on new data. In conclusion, the difference between the metrics in the two training scenarios highlights that the synthetic data quality might be further increased before its usage to enlarge the training set.

6.39. Spectral signature analysis

The investigation also compared synthetic and original datasets through spectral signatures. Figure 89 compares the original and the synthetic spectral signatures of the skin, malignant and benign lesions. From a visual inspection of the average spectral signatures and their ranges of variation, we can observe that the synthetic data outlines the same distribution as the original dataset.

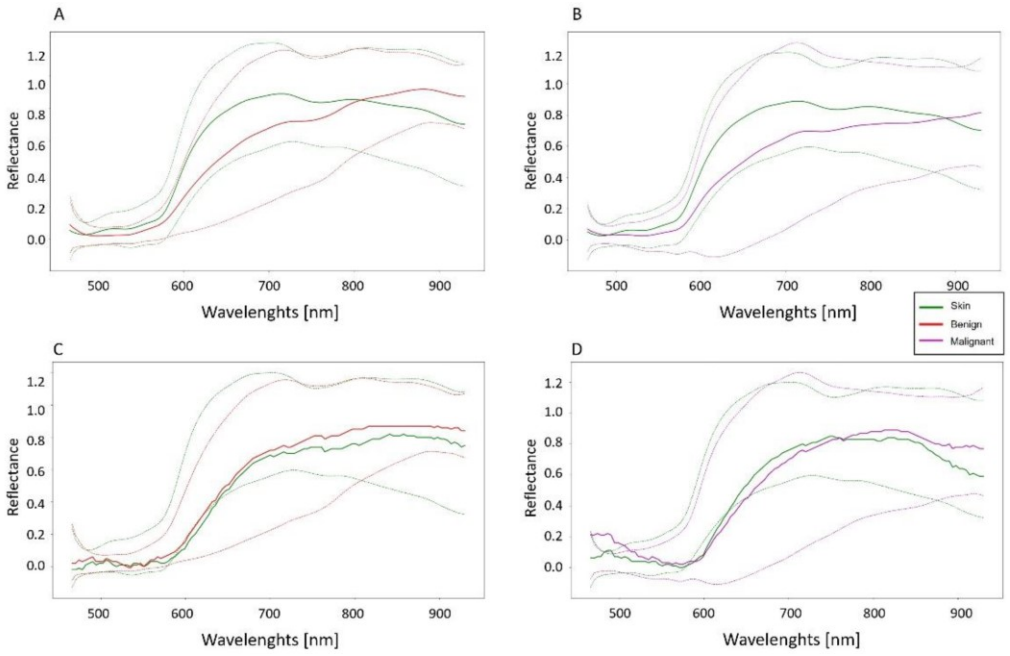


Figure 89. Synthetic and authentic spectral signatures comparison[56]

Nonetheless, the investigation quantitatively compared the spectral signatures via the Jensen–Shannon divergence[56], given by Equation 32, where v and w are the spectral signature to compare, and i represents the i-th band.

$$\mathbf{JS}(\mathbf{v},\mathbf{w}) = \frac{\mathbf{1}}{\mathbf{2}}\,\mathbf{\Sigma_i}(\mathbf{v_i log}(\mathbf{v_i}) + \mathbf{w_i log}(\mathbf{w_i}) - (\mathbf{v}+\mathbf{w_i})\mathbf{log}(\mathbf{12}(\mathbf{v}+\mathbf{w_i}))) \qquad \text{Equation 32}$$

The Jensen–Shannon divergence equals 0.6, 0.10, and 0.04 for the benign, malignant, and skin synthetic and real signatures, respectively. It is worth noticing that this metric is bounded by 1 for two distributions. Thus, the obtained values highlight the similarity between the real and synthetic signatures.

6.40. Comparisons with the state-of-the-art

Researchers widely explored GANs to generate synthetic images. However, the literature concentrated on generating synthetic data that typically is not HS images. Thus, a fair comparison can only happen with one work[56,101,103], which considered HS images related to skin cancer. The work reported the results only in terms of the mean spectral signature of the whole synthetic dataset, and no metric such the FID exists between the real and the synthetic dataset.

These considerations highlight that the proposed research describes and analyses, more broadly and comprehensively, a GAN architecture capable of generating hyperspectral synthetic data even if the training set contains a low number of examples.

6.41. Limits of the investigation and future developments

Data-centric applications strongly rely on the dataset size, influenced by subjects participating in clinical research and data acquisition campaigns. The data availability challenge appears in scenarios similar to the ones described in this thesis, where physicians employ a novel, non-standardised, and unique technology in routine clinical practice. Notably, data security policies currently obstruct research data sharing. Accordingly, this research proposed synthetic data assembling to overcome these limitations, providing researchers with increased and anonymous data, and accelerating deep learning methodologies into general clinical practice[3]. Recently, synthetic data generation has attracted considerable attention in the medical field, enhancing existing AI with novel data augmentation methodologies. Nonetheless, experimenters must provide knowledge concerning synthetic and original data distributions[1,3]. Not only can the

synthetic data be evaluated through quantitative appraisal, but it could also be with qualitative assessment processes provided by medical experts [1,56].

This investigation engineered a proof-of-concept to produce synthetic data to enhance and accelerate the development of AI algorithms for a specific context, especially when scientists engage a limited HS dataset to engineer a decision support system to aid skin cancer diagnosis. It aims to pave the course for deep learning techniques in medicine when the number of labelled samples is limited. Nonetheless, investigators should carry out large data acquisition campaigns to include data from several subjects, including different skin lesion types and many clinical centres. Additionally, physicians should perform a rigorous clinical study to validate the usefulness of the offered solution. Dermatologists should evaluate whether the HS spatial information correlates with the morphological features belonging to the different skin lesions. Therefore, qualitative evaluations could assess the similarity between the original and synthetic epidermal tumour distributions through a heuristic blind evaluation test. Finally, scientists should evaluate several HS camera models to develop a generative instance capable of producing distinct data distributions.

6.42. GANs for epidermal HS image generation final remarks

Here, we proposed a convolutional DCGAN architecture to generate HS medical data, particularly for skin lesion analysis, by operating a small-sized dataset to train the framework.

We adopted the FID metric to evaluate the similarity between the real and the synthetic data. We measured a 17.37 FID, which indicates sound synthesis and similarity between the distributions of the two datasets.

Additionally, a ResNet-18 trained only on synthetic data and tested on authentic images. The accuracy, precision, recall, and F1 score were all above 80%, demonstrating that the synthetic data and the authentic images are comparable. Finally, the investigators compared the spectral signatures qualitatively and quantitatively.

The literature reports only one work considering GANs for medical HS data[107]. Nonetheless, this work validated the results only in terms of visual similarity between the mean spectral signature of original and generated images.

Future research lines will investigate novel GAN architectures for medical HS images. Finally, the conditional GAN could produce different tumour etiologies besides benign and malignant ones.

6.43. Neural networks-based on-site dermatologic diagnosis through hyperspectral epidermal images

The fourth research in this chapter regards a DL pipeline comprising eight different architectures for classifying and segmenting HS in-vivo skin lesion images (Figure 90). Enhanced by data augmentation, transfer learning, and extensive hyperparameter tuning, the analysis optimised the networks with the database we illustrated in Chapter 3. The study worked the database, pre-processing and taxonomy the manuscript illustrated in Section 6.2.

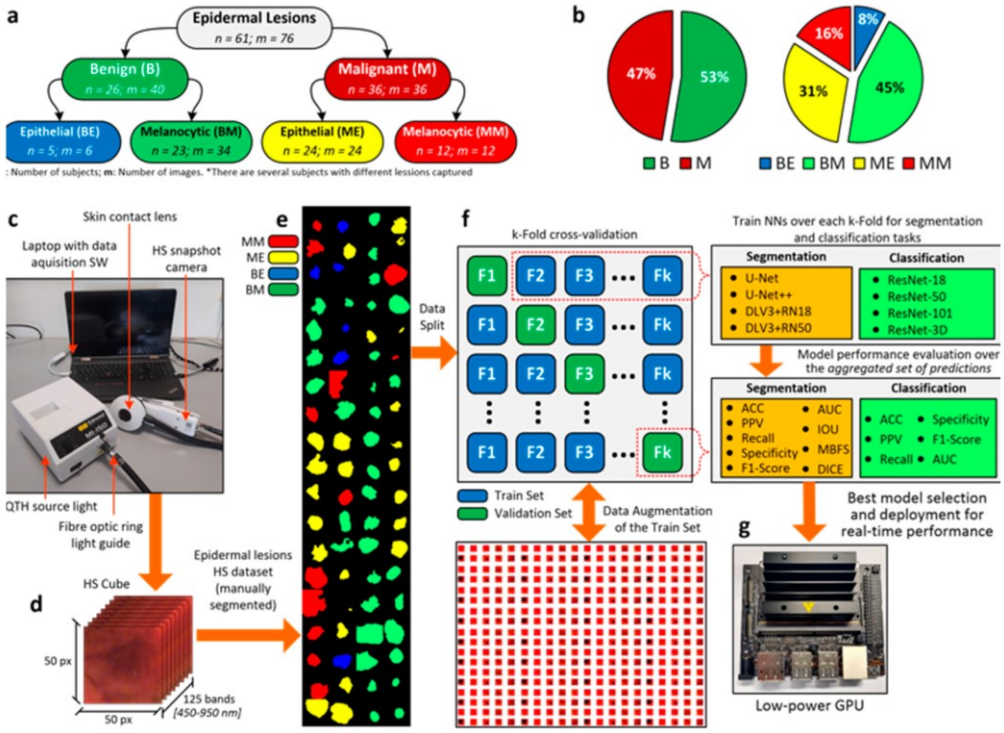


Figure 90. Proposed experimental framework. (**a**) Taxonomy of the epidermal lesions included in the HS database, including number of subjects and images in each category; (**b**) Distribution of images for the binary (left) and multilabel (right) classification problems; (**c**) Different elements of the HS acquisition system; (**d**) HS cube characteristics; (**e**) HS dataset ground-truths; (**f**) Proposed processing framework based on a k-fold cross-validation, including data augmentation and aggregated model evaluation; (**g**) Low-power Nvidia Jetson GPU for algorithm deployment to reach real-time performance[18]

Pathologists analysed the tumours through biopsy-proven histological assessment, ranking each lesion in the proposed taxonomy (Figure 90.a,b). Data originated from the customised HS acquisition system[38,99] in Figure

90.c, and segmentation masks derived from the HS cubes (Figure 90.d) to delimit the lesion boundaries in the images (Figure 90.e).

The ML algorithms described in the second research of this chapter yielded outcomes that encouraged the analysis of novel approaches comprising a set of CNNs to identify, segment, and classify epidermal lesions operating k-fold cross-validation (Section 3.11 - Figure 90.f).

This research also provides a lesion-border segmentation map. Researchers highlighted the need for more semantic information delivered to physicians. Undoubtedly, lesion boundary identification could decrease the chances of lesion reoccurrence and increase the healing probability[11,12,18].

Furthermore, the investigation eventually deployed a semantic segmentation network in a portable device[38,99] equipped with a low-power GPU (i.e., TS3 in Section 4.8), targeting daily clinical testing (Figure 90.g). It answered the market for an AI pipeline to serve a real-world medical scenario, which could assist dermatologists in scaling up skin cancer screening and early detection, reducing the number of false positives and negatives and, hence, the number of biopsies and histopathological evaluations[45,47,98].

Recent examinations[1,3] reported encouraging developments of AI in various disciplines, again emphasising the demand for adequate computing power to process DL algorithms. The proposed architectures, targeting handheld medical instrument deployment, attained and enhanced the well-known dermatologist human-level detection performances for both malignant-benign and multilabel classification tasks, as they were able to diagnose HS data considering real-time constraints for on-site diagnostic examinations [45,47,98].

6.44. Deep learning methodology

The research trained eight CNNs architectures to classify and segment the HS skin lesion images. On the one hand, ResNet-18, ResNet-50, ResNet-101, and a ResNet-50 variant, which exploits 3D convolutions, classified the images into the taxonomy presented in Section 2.7. On the other hand, U-Net, U-Net++, and two versions of the DeepLabV3+ architecture - one having as backbone structure a ResNet-18 and the other a ResNet-50 - performed semantic segmentation of the epidermal lesions. This doctoral thesis reported the detailed description of the abovementioned architectures' fundamentals in Chapter 3.

Furthermore, transfer learning (Section 3.13) improved the results of the learning-based architectures by exploiting features belonging to the previous training task. Consequently, all the listed architectures had already undergone optimisation based on the ImageNet dataset[58]. MATLAB offers the possibility of instantiating already-trained deep learning models that can be modified to accept different image sizes.

The training set statistical assortment increased through data augmentation using several diversifications, including geometric (i.e., rotation, mirroring, scaling, cropping), filtering, random centre cropping, colour transformations, and pixel substitutions. The research included either a linear combination of random pixels of tumours belonging to the same category or directly exchanged them. The same procedure occurred to skin pixels. Eventually, the dataset comprised approximately ten thousand images in the training set.

Data augmentation produces effective results in computer vision tasks, significantly reducing overfitting[65]. Furthermore, we introduced salt-and-pepper white noise in random image bands to enlarge the training set. The augmentation procedure was iterative. One of the data augmentation techniques took part in the training set, and a new data cluster originated by unifying the original and transformed images. Following this, a second technique participated in the new group. Finally, this procedure recursively happened to broaden the training set exponentially. The investigation did not apply such augmentation techniques to either the validation or the test sets to reject the hypothesis of biased results.

All architectures receive input size $50 \times 50 \times 116$, concerning height, width, and the number of wavelengths. We not only placed a dropout layer in each ResNet architecture, but we also introduced the L2 weights penalty in the loss function to additionally reduce overfitting. The semantic segmentation networks already met the requirement in their original design. Cross-entropy loss function and the Adam method[93] participated in the training. The learning step decreased by multiplication by the dropping factor: it steadily and linearly decreased after each predetermined number of epochs. Batch size, number of epochs, learning rate, and drop factor period were 32, 10, 9×10^{-5}, and 3, respectively, for all architectures. The drop factor and L2 penalty were 5×10^{-1} and 10^{-4}, respectively, for the semantic segmentation models and 10^{-2} and 9×10^{-2}, respectively, for the classification models.

The investigation operated the first test system from Section 4.8 to train and test the DL pipeline. Accordingly, MathWorks' MATLAB 2021b Release - Deep Learning Toolbox was used for the network design and implementation.

6.45. K-fold cross-validation and aggregated testing

This research adopted the cross-validation procedure described in Section 3.11. It randomly shuffled the original HS dataset comprising 76 images and split it into ten groups.

Next, each k-th unique group constituted the test data and the model trained on the remaining groups. Accordingly, data augmentation participated in the k-1 groups used for training. The researchers trained the

model on the training set and evaluated it on the test set, retaining the prediction evaluated at each iteration and discarding the model.

Accordingly, we trained the model k times and recorded its estimate for each test set. Consequently, the performance metrics for classification and semantic segmentation lean on the aggregated group of predictions, namely the union of each k-fold test set generated for each DL architecture through the procedure.

6.46. Performance evaluation

This research evaluated the DL architectures' performance operating the metrics in Section 3.11. This research assessed the pixel-based occurrences for semantic segmentation. The assessment computed the following metrics: accuracy, sensitivity, specificity, precision, Receiver Operating Characteristic Area Under the Curve (AUC), precision, and F1-Score[2,57]. For the segmentation task, it also computed the Mean Boundary-F1 Score (MBFS), the Intersection Over Union (IOU), and the DICE coefficient. These final set of evaluations comprised the join of the prediction set of each architecture conveyed through the k-fold cross-validation[18].

Similarly, the GPU accelerated-computing performance assessment comprised elapsed time, measured in seconds (s), and power dissipated, measured in Watts (W).

6.47. Architecture selection for GPU deployment

The investigation assessed each architecture's semantic segmentation performance. Consequently, the comparison yielded the model having the best predictive capabilities: the U-Net++. Consequently, the investigation produced a custom C/CUDA code in terms of both the architecture's weight and the HS epidermal lesion classification (Section 4.6). This first serial stage ended with image pre-processing, and the subsequent stage consisted of parallel semantic inference, exploiting the U-Net++ layers. The choice of a hybrid C/CUDA code proved valid concerning the real-time classification of skin cancer HS images described in the second research of this chapter (Section 6.16). U-Net++ was a 130 layer-wise network having 130 M parameters.

6.48. High-performance computing development

Several literature reviews stated the challenge of engineering an AI pipeline to scale up the global accessibility of epidermal screening at an expert level[1,3]. The instrument should meet board-certified dermatologist diagnosis and feature a semantic segmentation to determine the tumour boundaries, thus improving remission and avoiding reoccurrence[1,3,11,12].

Similarly, a GPU could play a crucial role in AI applications for healthcare. CNNs consist of millions of parameters arranged in a matrix manner across their layers. Their multiplication with input data lets neurons fire and highlight features to determine the diagnostic outcome. Accordingly, DL models can be computationally pricey. GPU deployment enables high-performance parallel computing and opens the possibility of deploying the diagnostic model on handheld devices (Chapter 4).

Accordingly, this research addressed the CUDA extension to C language and a custom code to embed the U-Net++ inside a low-power Nvidia Jetson GPU (TS3 in Section 4.8). The Jetson board is a 128-core Maxwell architecture designed for embedded applications and equipped with a quad-core ARM A57 running at 1.43 GHz. The board runs applications consuming 5 or 10 W, depending on the power budget mode set on the device.

The investigation extensively operated the CUBLAS and cuDNN libraries, described in Chapter 4 of this thesis. They contain efficient routines for linear algebra concerning DL, such as convolutional and normalisation layers, activation functions, and feedforward inference. The CUDA kernels operate on tensors having the following shape: number of examples (N), number of channels (C), height (H), and width (W)[79].

The investigation compared the C/CUDA codes to the previously developed MATLAB script at each U-Net++ building stage. We assessed each intermediate result of the inference pipeline.

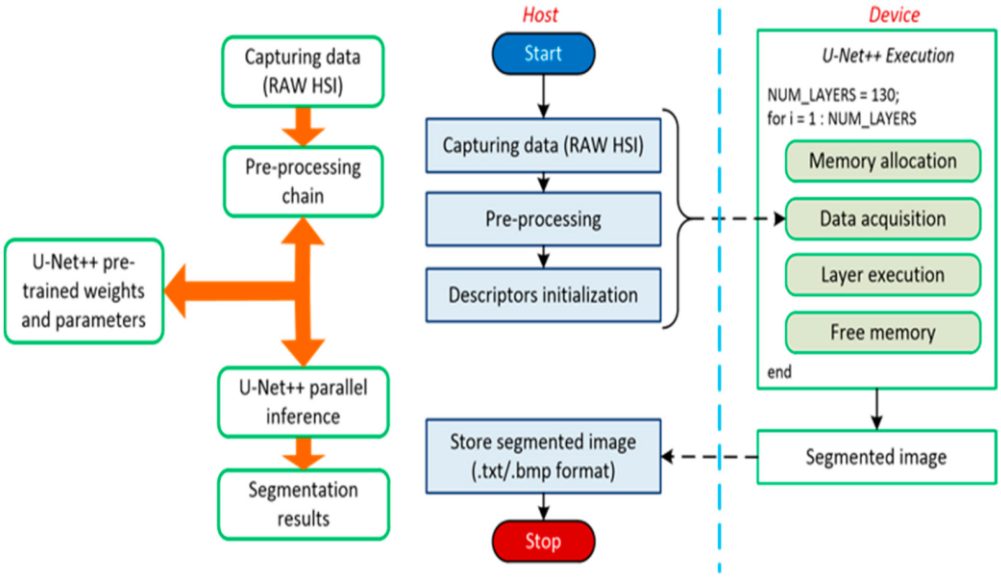


Figure 91. CUDA execution logic and data transfer flow[18]

Figure 91 shows the outcome of the custom development, whose execution starts on the CPU, the Host. We collect the HS epidermal lesion image and the neural network weights. Once we initialised all the necessary

elements and descriptors, we moved to the device memory, namely the GPU memory, the data needed from the U-Net++ for inference.

Due to the limited memory of the Jetson GPU (TS3 in Section 4.8), we arranged the prediction to compute a layer output at that time. Remarkably, we allocated memory to each layer, acquired the previous dataflow outcome, executed the layer, produced the new result, and finally freed the memory.

Once the loop ended, a segmented image originated from it, which we moved back to the Host, where the result was saved and displayed on the handheld device. The semantic segmentation of HS skin cancer images runs in less than a second.

6.49. Classification of epidermal lesions

The CNNs operated a small-sized dataset. We then evaluated the performance of the architectures, employing a 10-fold cross-validation methodology. Additionally, the taxonomy proposed in Figure 90.a and Figure 11 is a trade-off between being medically comprehensive, consistent, and to fit DL classifiers. Undoubtedly, the tree-structure categorisation is well-suited to treat patients according to the highest healthcare standards and provides the best classification performance[108]. Indeed, we propose coarse-grained malignant-benign classifications and fine-grained classifications, allowing expert professionals to differentiate between numerous severe conditions.

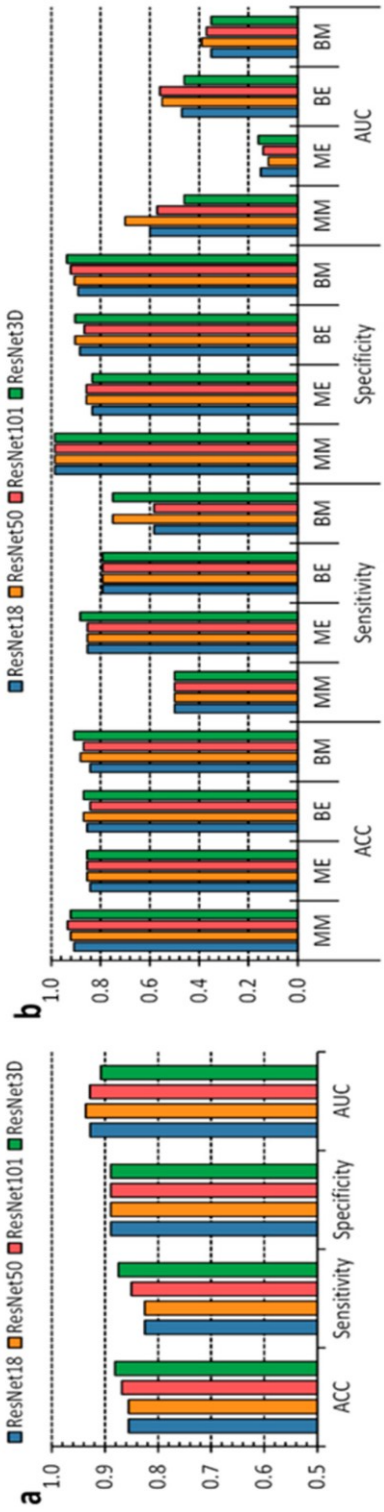


Figure 92. Performance of the epidermal lesion classification. (**a**,**b**), Binary and multilabel classification performance of the four different approaches, respectively[18]

Discrimination between benign and malignant lesions offered stable and robust measurements, meeting sensitivity and specificity above 80% (Figure 92.a). We observed the ResNet-3D achieving the best results (87% sensitivity and 88% specificity). Furthermore, we can determine through AUC outcomes that adequate thresholding could increase performance by over 90%. On the other hand, the multilabel classification retains an Malignant Melanocytic (MM), Benign Epithelial (BE), and Benign Melanocytic (BM) sensitivity performance below 80%. Regardless, the specificity for all classes is above 80% (Figure 92.b).

Considering that having more groups induces each group to have fewer examples, the diminished number of images in the BE and Malignant Epithelial (ME) categories elicits sparse information regarding inter-patient variability. The multilabel classification scenario demonstrated the ResNet-50 and the ResNet-3D as having the best performances.

A drawback of considering an aggregated validation set, namely the union of each k-fold test set, is the risk of having inconsistent AUC results. Academic authors usually compute AUC over a single classifier whose prediction probability retains a classification. Aggregating the results means we unify possibly disharmonious likelihoods from different classifiers trained at each k-fold iteration. That is why we can appreciate acceptable classification metrics measurements related to low AUCs.

6.50. Anatomical segmentation of epidermal lesions

Tumour border detection is a crucial step towards patient healing and disease remission. This step is significant for skin cancer but also gains relevance when experts consider other tumour types, such as brain cancer. Indeed, the more complex the disease is to reach inside the human body, the better its boundary detection should be to sidestep its reoccurrence and enhance remission chances.

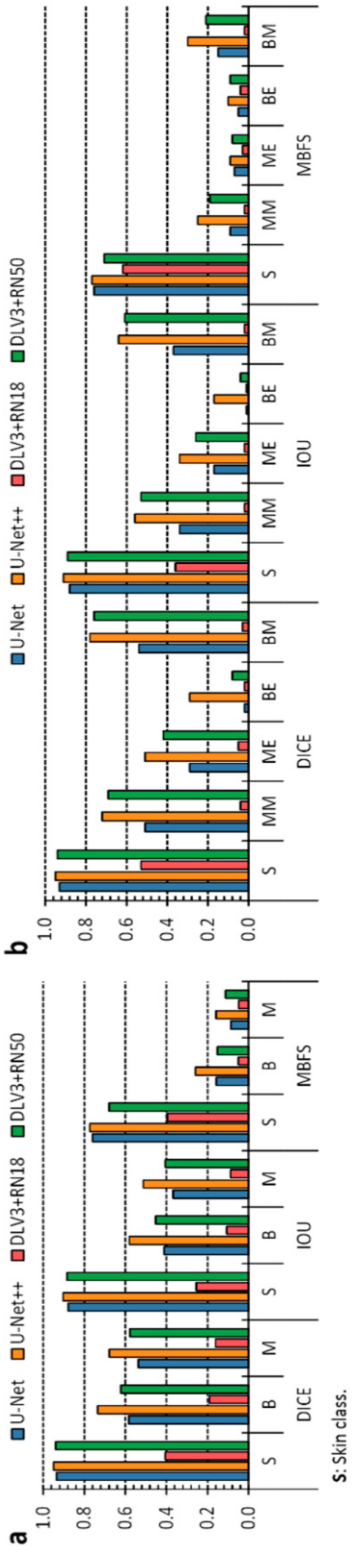


Figure 93. Performance of the epidermal lesion segmentation. (**a**,**b**), Binary and multilabel segmentation performance of the four different approaches, respectively. The acronyms have the following meanings: Skin (S), Benign (B), Malignant (M), Benign Epithelial (BE), Benign Melanocytic (BM), Malignant Epithelial (ME), and Malignant Melanocytic (MM)[18]

We trained the U-Net, the U-Net++, and two DeepLabV3+ versions, having ResNet-18 and ResNet-50, respectively, as the backbone structure, to answer the market for semantic information concerning the skin lesion boundaries. We evaluated each semantic architecture using some of Section 3.11 metrics per class. These results also include the skin class in the results. Concerning binary segmentation (Figure 93.a), this thesis reports skin DICE and IOU higher than 0.9, apart from the DeepLabV3+ RN18 architecture, which yielded a lower segmentation performance. Nonetheless, the investigation observed limited performance regarding benign and malignant classes - specifically, DICE measurements below 0.8 and IOU under 0.6. The U-Net++ exhibited the best segmentation results over all the categories.

Similarly, the U-Net++ offered the best outcomes concerning the multilabel segmentation scenario (Figure 93.b). Regardless, the IOU measurements for the ME and BE categories were lower than 0.4. The results might be due to the high inter and intra-patient variabilities concerning lesion etiologies and the few samples belonging to different groups.

6.51. U-net++ results and rationale

This investigation evaluated the U-Net++ for embedded deployment for two main reasons. First, it exhibited the best performance in multilabel and binary assignments (Figure 93). Similarly, the architecture presents the highest number of layers and parameters. In other words, satisfying real-time constraints[12] with such architecture firmly ensures that the same time limitation could exist with smaller CNNs. Researchers define a real-time constraint as a mandatory temporal deadline to carry out a task[11,12,19]. A reasonable time limit for skin cancer diagnosis can be arranged around a few minutes since its growth takes several weeks.

This doctoral thesis chose U-Net++ as the network for embedded deployment and met a real-time constraint specified for epidermal lesion classification and segmentation: recording time stamps ranging from 0.230 to 1.210 s concerning different GPU architectures, which we compared in terms of time and power consumption (Figure 93).

6.52. U-net++ embedded deployment

This research developed a custom code through the CUDA extension to C language[78] to embed the U-Net++ inside a low-power Nvidia Jetson GPU (TS3 in Section 4.8).

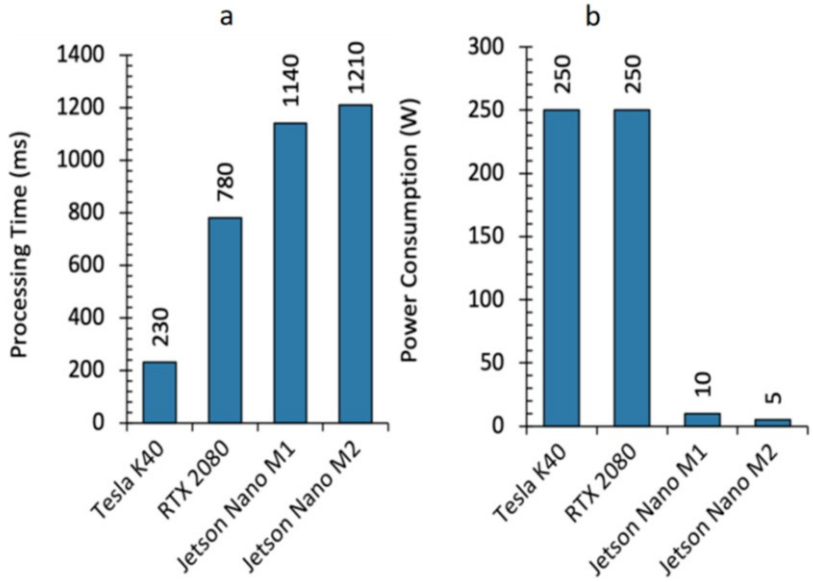


Figure 94. Deployment performance. (a) Processing time and (b), power consumption comparisons of the different Nvidia GPUs considered in this study. Jetson Nano M1 and Jetson Nano M2 indicate the two possible power configurations of the Jetson Nano board, which are 10 and 5 W of power budget, respectively[18]

It extensively used CUBLAS and cuDNN libraries and tested them on three different GPU boards produced by Nvidia (Chapter 4), namely the three test systems from Section 4.8. The three boards cover the range of products proposed by the vendor. The RTX 2080 is a consumer board featuring 2944 cores working at 1.8 GHz and equipped with 8 GB of DDR6 memory. The Tesla K40 GPU is a board developed for computationally intensive applications and equips 2880 cores working at 750 MHz and 12 GB of DDR5 memory. The Jetson Nano board is a low-power GPU featuring 128 cores at 1.6 GHz and 4 GB of DDR4 memory. The Tesla K40 and RTX2080 obtained the best processing times (Figure 94.a), with elaborations ranging from 230 to 780 ms and power consumption of 250 W (Figure 94.b).

On the other hand, the Jetson Nano board took from 1.14 s to 1.21 s to process the images, consuming 10 to 5 W (Figure 94.a,b).

Eventually, all three boards yielded processing times that would fit the target application well. Additionally, the Jetson Nano board has a power consumption which enables the development of a portable and handheld diagnostic instrument, especially the M2 power configuration.

6.53. Comparison with expert dermatologists

Researchers believe it is difficult and not reasonably fair to compare studies due to different settings, from data employed to algorithm

structures and the proposed taxonomy. In general, HS images contain broader information different from classical RGB pictures. Several reviews evaluated more than fifty studies retaining the different settings discussed and whose research involved hundreds of expert dermatologists[45–47,98,109]. Consequently, this thesis can establish a well-known plateau of performance. Considering the proposed taxonomy, expert dermatologist sensitivity, specificity, and accuracy concerning benign and malignant lesions lie at approximately 80%, 75%, and 70–85%, respectively. We must highlight that we reported the highest measurements when expert professionals, rather than trainee dermatologists, participated in the experiments.

Nonetheless, they reached around 55–60% accuracy when targeting more classes in taxonomy. Accuracy decreased to 40–45% when trainee dermatologists performed the same task[45–47,98,109]. Each mentioned performance evaluation does not belong to the same study, and we must stress that researchers traded off high sensitivity with low specificity in some scenarios.

Accordingly, the AI-based pipeline proposed in this study met and exceeded the dermatologist-level classification of skin cancer, which does not usually include an automatic anatomical segmentation of the boundaries of the lesions. Undoubtedly, ResNet-3D achieved the best accuracy in the multilabel scenario, attaining peak performance at 92.10% for the MM class (Figure 92).

6.54. Discussion and conclusions

This doctoral thesis presented several critical matters. It designed an AI system to assist dermatologists in clustering epidermal tumours, despite the limitation of the small-sized HS dataset. Notably, it researched a consistent methodology to develop DL algorithms and cope with small-sized datasets to meet and improve the well-known dermatologist diagnostic performance plateau.

Cursed by the absence of large datasets (Section 2.12), it took some time for HSI-based applications to become feasible in terms of tasks operating classical RGB or multispectral images. Accordingly, the studies considered by the authors of several systematic reviews consisted of databases with significant data, thus highlighting the diagnostic performance plateau reached. Consequently, classification techniques for HSI often exploit transfer learning and data augmentation to improve classification performances in different research fields[45–47,98,109]. Algorithms employing HS images usually comprise the classical pixel-wise models we mentioned in the second research of this chapter. Even though the algorithms only work with spectral and not spatial information, their sensitivity and specificity concerning MM and NMSC evaluated through leave-one-out lie around 80 and 77%, recently improved to 87.5 and 100%, respectively[38,99].

This investigation responded to the market for AI clinical applications and the need for computational power to assist it in engineering a handheld instrument equipped with a low-power GPU. The tool should replace the expensive and time-consuming gold-standard diagnostic procedure to turn modern DL algorithms into medical equipment[3].

Recently published articles highlighted that the Food and Drug Administration (FDA) is moving towards approving AI-based medical devices[3]. It is a crucial turning point considering challenging historical periods, such as those raised due to the Covid-19 pandemic. AI-based medical instruments should aid professionals during challenging times and participate in frontline emergency clinics, remote places, or the developing world.

This doctoral thesis conceived a blueprint dermatological instrument to improve the worldwide accessibility of epidermal screening at the professional level. Expert dermatologist classification accuracy of epidermal lesions usually depends on the number of classes considered. At most, it reaches 85% in a malignant-benign classification scenario. The gold-standard procedure implies clinical and dermoscopic inspection, followed by biopsy and histopathological examination. In other words, the subjective nature of the inspection biases the classification accuracy measurement of malignant lesions. Undoubtedly, physicians only diagnose lesions already marked as suspicious[45–47,98,109].

This doctoral path designed CNNs to attain and enhance well-known dermatologist human-level classification performance concerning specificity, sensitivity, and accuracy. To the best of the thesis' knowledge, no research exists yet concerning HS skin cancer image segmentation to produce a mask to inform doctors about lesion boundaries. Similarly, other studies mainly focused on producing high-end results considering classification scenarios with unessential clinical applicability[45–47,98,109].

This research improved the classification taxonomy and avoided scenarios such as MM compared against a specific lesion type. It developed an HS system containing much more information regarding RGB, multispectral, and other spectroscopy strategies. It used artificially intelligent architectures and algorithms to build on the existing literature concerning statistical approaches for spectral signature analysis[45–47,98,109].

This thesis was eager to respond to the demand for an AI-based pipeline to assist or replace the expensive and time-consuming gold-standard procedures. Accordingly, it deployed a semantic segmentation network on a low-power Nvidia Jetson GPU device targeting a portable instrument containing an HS camera. The designed proof-of-concept AI system can classify and segment epidermal lesions in, at most, 1.21 s, and expert professionals could use the future implementation in real-world clinical scenarios.

Nonetheless, the study exhibits limitations. The main limitation is related to dataset size, which in turn produces others. Indeed, HS imaging is a powerful tool compared to classical RGB pictures. Chromophores

characterise skin's spectral properties and allow lesion clustering into different etiologies. HS imaging systems gather skin-reflected and transmitted light into several wavelengths ranges on the electromagnetic spectrum, enabling potential skin-lesion differentiation through machine and DL algorithms. Indeed, each pixel contains meaningful information concerning an object's properties. Not only are some lesions in the dataset transitioning from benign to malignant lesions, but lesions and skin signatures might differ slightly.

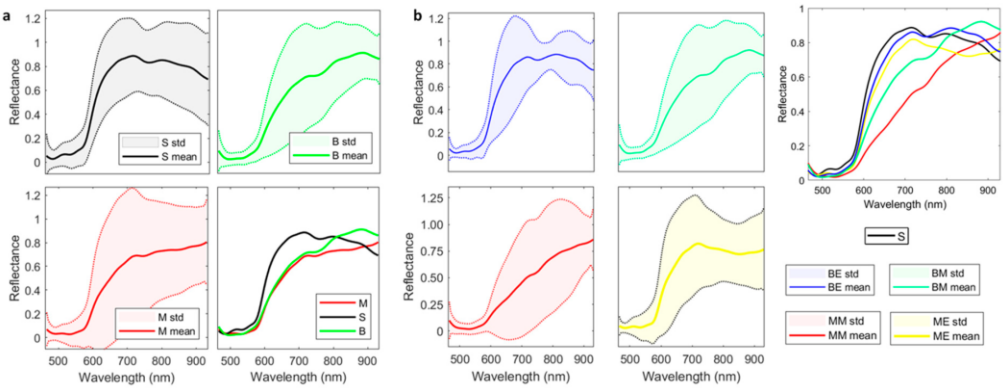


Figure 95. Mean and standard deviation (std) of the spectral signatures of the HS dataset. (**a**) Spectral signatures of skin, Benign and Malignant; (**b**) Spectral signatures of benign epithelial and melanocytic and malignant epithelial and melanocytic. S: Skin, B: Benign; M: Malignant; BE: Benign epithelial; BM: Benign melanocytic; ME: Malignant epithelial; MM: Malignant melanocytic[18]

Moreover, each patient has a unique skin signature which causes the test images to be very different from the training ones, increasing inter-patient variability. Figure 95.a represents the spectral signature means and standard deviations of normal skin (S), benign (B), and malignant (M) lesions. Notably, Figure 95.b reports each subtype lesion's spectral signature mean and standard deviation. Consequently, an extensive dataset should cope with this problem and allow CNNs to concentrate more on the insightful parts of the wavelengths, enhancing the semantic segmentation outcomes yielded in this thesis. Accordingly, future literature should focus more on algorithms that better exploit the massive amount of information in a single spectral cube to improve current classification and segmentation performance.

6.55. Attention-based skin cancer classification through hyperspectral imaging

This chapter's fifth and last research proposes a Vision Transformer (ViT)[100] based classifier targeting HS skin cancer images. It represents a

leap forward concerning the investigation mentioned earlier in the text. While the transformer architecture represents the highest standard for tasks involving Natural Language Processing (NLP), its usage concerning Computer Vision (CV) remains limited, particularly with HSIs. The attention mechanism is the ground basis of transformers, and it either works in conjunction with CNNs or substitutes certain aspects of CNNs while keeping their entire composition intact[100].

The attention mechanism is the flexible control of limited computational resources: it enables sequence learners, namely neural networks working with time series, to understand better the relationship between different tokens in the sequences they are training on. For a given sequence, when a specific token attends to another, it means they are closely related and have an impact on each other in the context of the whole sequence. Transformers comprise multiple so-called self-attention layers, whose task is to weigh the significance of each input data part differentially[100].

The self-attention layer in ViT makes it possible to embed global information across the overall image. The model also learns from training data to encode the relative location of the image patches to reconstruct the structure of the image later.

The analysis operated the database, pre-processing and taxonomy we illustrated in Section 6.2.

6.56. Vision Transformers (ViT) for hyperspectral imaging

Vision transformers (ViT) are DL architectures that lean on the self-attention mechanism[100]. Figure 96 shows the structure of a ViT. Typically, a ViT receives as input a 1-D array. Consequently, N-D tensors such as HS images transform into 1-D arrays through original image division into patches of the same dimension. This partitioning occurs through a convolution operation. Let $X \in \mathbb{R}^{H \times W \times C}$ be an HS image with a spatial dimension of $H \times W$ and C spectral channels. We can define each patch with $X_p \in \mathbb{R}^{H \times P \times P \times C}$, where $P \times P$ is the resolution of a single patch. Accordingly, the number of patches forming an image is $N = \frac{H \times W}{P^2}$.

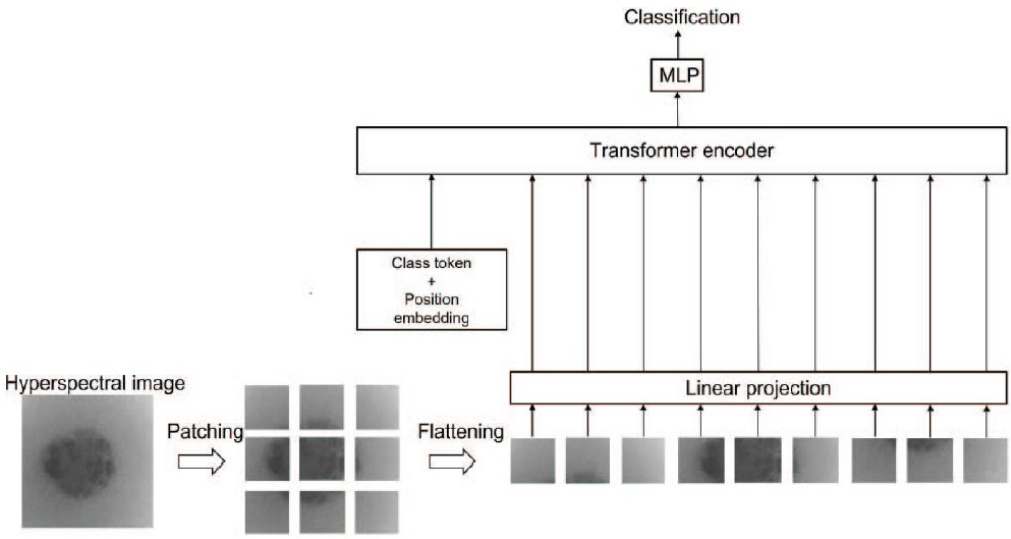


Figure 96. Architecture of a Vision Transformer

The ViT uses a Q-D array of latent variables to project the patches in a new space. Then, the architecture associates a class token to each patch, together with an array containing information about the relative position of each patch concerning the original image, the so-called position embedding. These data represent the input to the transformer encoder based on three main components: Multi-head Self Attention (MSA), Multi-Layer Perceptron (MLP) and normalisation. The architecture links these components, as shown in Figure 97.

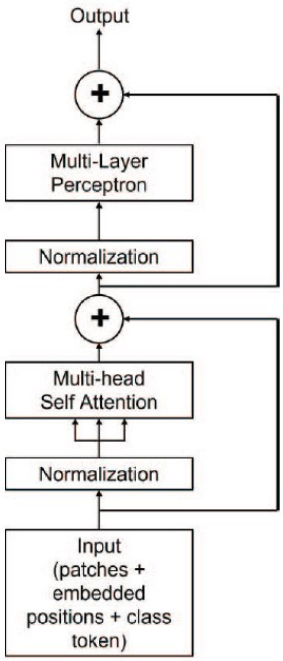


Figure 97. The architecture of a transformer encoder

In the self-attention mechanism, the architecture projects each input vector to generate three matrices: Key (K), Query (Q), and Value (V). For each input vector, we can retain the attention map according to Equation 33[100], where d stands for a scaling factor:

$$\mathbf{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathbf{Softmax}(\frac{\mathbf{Q}\times\mathbf{K}^{\mathbf{T}}}{\sqrt{\mathbf{d}}}) \times \mathbf{V} \quad \text{Equation 33}$$

The Multi-head attention mechanism leans on Equation 33. The main difference is that the architecture linearly projects the Q, K and V vectors into a suitable space. Then, in parallel, it applies the attention mechanism to the new vectors. Consequently, we concatenate the attention values to obtain the output. Typically, L-concatenated MSA layers produce the input for the MLP that generates the final classification.

HS images feature a higher number of channels than standard RGB images. Accordingly, the number of multiplications performed by the MSA layer is very high. This research solved this issue by introducing convolution operations before applying the patching procedures. Remarkably, this thesis proposes three convolutional layers, each featuring a 2-D CNN layer, a normalisation and a ReLU activation. Each convolutional layer comprises 3 × 3 filters. The number of filters is 58, 29 and 14 for the first, the second and the last convolutional layer, respectively.

These layers reduced the channels from 116 to 14. Therefore, the image size given as input to the ViT is 50 × 50 × 14. The proposed strategy reduces both the computational complexity and the memory occupancy of the ViT architecture compared to giving the original HS image as input.

6.57. Performance metrics

The investigation trained the ViT to classify the lesions into the four categories of the taxonomy in Section 2.7: malignant melanocytic (MM), benign melanocytic (BM), malignant epithelial (ME) and benign epithelial (BE).

Likewise, this investigation also operated the 10-fold cross-validation as described in Section 3.11. We divided the original HS dataset comprising 76 images into K groups. Next, each unique group tested the model, which instead trained on the remaining K-1 groups. Thus, we augmented the data in the training groups as described in Section 3.10. The model was fit on the training set and was evaluated on the test set, retaining the prediction evaluated at each iteration and discarding the model. Therefore, the model trained k times, and we stored the estimations for each k-th test set. Consequently, we assessed the performance metrics on the joined group of estimations. We assessed the classification in terms of accuracy, specificity, and False Negative Rate per class (FNRc) defined in Section 3.11 and 6.27.

6.58. Experimental results

This research developed the ViT architecture in MATLAB 2020a by writing custom scripts exploiting the Deep Learning Toolbox. The code runs on the first test system described in Section 4.8.

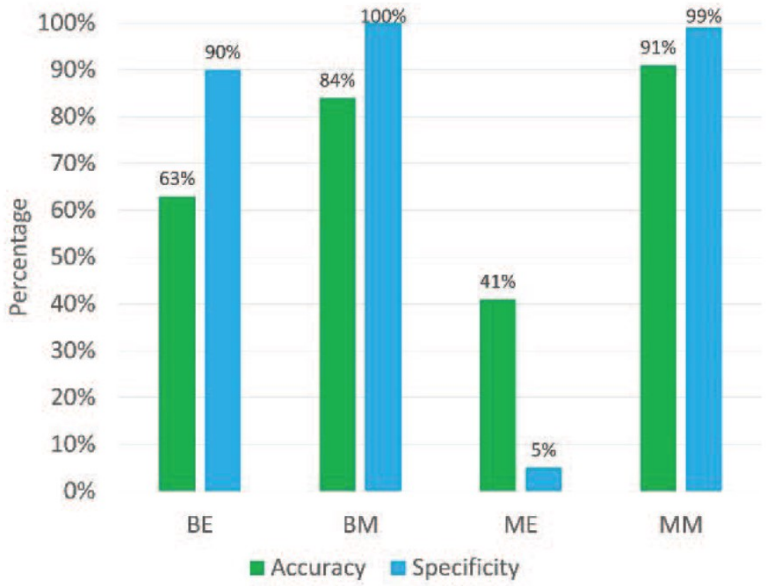


Figure 98. Performance of the proposed ViT. Accuracy and Specificity are reported as percentages. BE, BM, ME and MM represent Benign Epithelia, Benign Melanocytic, Malignant Epithelial and Malignant Melanocytic, respectively

Figure 98 reports the performance computed on the aggregated predictions of the 10-fold cross-validation technique. This chart shows that the proposed architecture can classify benign and malignant melanocytic lesions with high accuracy and specificity. On the other hand, considering the BE class, the network features a high specificity, but the accuracy is around 60 %. Therefore, the low number of images in the original database labelled as BE is insufficient to train the proposed network efficiently. At the time of writing, few works operated HS imaging for skin cancer detection, including the investigations described in this chapter. The works rely on the same processing chain exploiting K-Means clustering and SVM classification, and the classification taxonomy adopted in these works is different from the proposed research. Thus, a direct comparison is not fair and can be carried out only in terms of FNRc. The second investigation in this chapter computed the FNRc for 18 images, obtaining values up to 60%. In this study work, instead, the metric assesses the K-fold cross-validation, which is more robust and reliable than the values reported in the state-of-the-art. The ViT obtained FNRc values ranging from 6% to 30%.

Therefore, the proposed attention-based network represents a thrilling and promising solution for skin cancer detection through HS images.

This research also described the network's performance in terms of processing time, considering classifying 100 images and computing the mean and standard deviation. The mean processing time is equal to 65.2 ms, with a standard deviation of 7.5 ms. The system proposed in the second investigation of this chapter, whose parallelisation was carried out in this thesis, takes varying processing times, ranging from 350.0 ms to 2.06 s. Consequently, the proposed work outperforms previous investigations in terms of processing speed.

Additionally, this work's variability of the processing time featured is significantly lower than in previous investigations.

Undoubtedly, the ViT architecture has a fixed number of layers that perform a fixed number of operations. On the other hand, the processing chain proposed in this thesis' second research includes the K-Means clustering, which iterates the operations based on the clustering error. Thus, the number of iterations performed by it is not deterministic and strictly depends on the initial values of cluster centroids.

6.59. Final remarks

This thesis proposed a novel attention-based network to classify skin cancer through HS images. The proposed network is designed and validated using a real HS dataset, adopting the K-fold cross-validation technique to produce robust results. Since the original dataset featured only 76 image, we applied data augmentations to the training data. Performed augmentations included geometrical transformations, filtering, random centre cropping, colour transformations, pixel substitution and random addition of gaussian white noise. The model was trained augmenting at runtime the training set and then performing the tests only on the real imaging, considering a number of folds equals to 10.

The results emphasise that the attention-based mechanism is an interesting and promising solution for medical HS images classification, since the false negative rate is half compared to the state-of-the-art. Eventually, the classification times are significantly lower than the best solutions proposed in the literature. Finally, the proposed network adopts a fixed number of layers whose number of mathematical operations is deterministic, making the measured processing time more stable than the results reported in previous works.

Future research will focus on improving the proposed network and evaluating different layers configurations.

6.60. Main contributions summary

Here, the thesis proposes a list of main contributions deriving from the pieces of study described in the earlier sections.

- *Hyperspectral imaging acquisition set-up for medical applications*
    - **Addressed problem**: There is a limited availability of efficient hyperspectral imaging systems for medical applications, which can limit the adoption and advancement of this technology for skin cancer diagnosis. Additionally, there is a limited availability of skin cancer hyperspectral dataset for research, which can limit the development and testing of new models.
    - **Proposed solution**: Hyperspectral acquisition system engineered to gather diagnostic clinical data concerning skin cancer.
    - **Advantages**: Using low-cost imaging techniques for skin cancer diagnosis can be more cost-efficient and accessible, as they rely on readily available hardware and open-source software.
    - **Disadvantages**: One limitation of using low-cost imaging techniques for skin cancer diagnosis is that they have not yet been extensively applied in a real-world scenario, where large and diverse datasets can be gathered. This can limit the ability of these techniques to generalize to a wider range of skin cancer cases and populations.
    - **Main contributions**: Enhanced efficiency, potential for more cost-efficient and widely accessible systems with respect to state of the art.
- *Parallel classification pipelines for skin cancer detection exploiting hyperspectral imaging on hybrid systems*
    - **Addressed problem**: Lack of real-time and accurate diagnostic systems for skin cancer detection.
    - **Proposed solution**: A parallel classification framework based on HSI using K-means and SVM algorithms for automatic in-situ PSL identification.
    - **Advantages**: One advantage of using real-time classification for skin cancer diagnosis is that it can potentially assist dermatologists in identifying different types of pigmented skin lesions (PSLs) quickly and accurately.
    - **Disadvantages**: The study is limited to a single hospital and further testing is necessary to determine its generalizability to other hospitals or populations. It also

requires access to laboratory data, which may not be available in all settings.
    - **Main contributions**: Improved accuracy, potential for earlier diagnosis.
- *Deep convolutional Generative Adversarial Networks to enhance Artificial Intelligence for skin cancer applications*
    - **Addressed problem**: Limited HS datasets available for skin cancer analysis.
    - **Proposed solution**: Convolutional DCGAN architecture to generate HS medical data.
    - **Advantages**: Federated learning can provide researchers with access to a large and diverse dataset while maintaining patient privacy through anonymous data. This can accelerate the development and application of deep learning methodologies in general clinical practice.
    - **Disadvantages**: The technology has the potential to improve clinical practice by accelerating deep learning methodologies and increasing access to anonymous data, but large-scale data acquisition campaigns are necessary to include diverse skin lesion types and clinical centers.
    - **Main contributions**: GAN architecture for generating hyperspectral synthetic data with low sample size, evaluated by FID metric and validated using resnet-18 trained on synthetic data to classify real images.
- *Neural Networks-Based On-Site Dermatologic Diagnosis through Hyperspectral Epidermal Images*
    - **Addressed problem**: End-to-end dermatologic diagnosis using HSIs.
    - **Proposed solution**: AI system to assist dermatologists in clustering epidermal tumors and improve classification taxonomy.
    - **Advantages:** Improves dermatologist's classification performance in specificity, sensitivity, and accuracy; achieved real-time classification on a low-power Nvidia Jetson GPU device using a semantic segmentation network for a portable instrument containing an HS camera.
    - **Disadvantages**: The main limitation is dataset size, leading to other limitations, including unique skin signatures for each patient and inter-patient variability in these signatures.
    - **Main contributions**: The study concentrates on developing deep learning algorithms for small datasets to improve dermatologist diagnostic performance. The future research should emphasize exploiting the vast amount of

information in a single spectral cube for better classification and segmentation performance.

- *Attention-based skin cancer classification through hyperspectral imaging*
    - **Addressed problem**: The study proposes an end-to-end dermatologic diagnosis using HSIs and suggests exploiting the vast amount of information in a single spectral cube for better classification and segmentation performance.
    - **Proposed solution**: A novel attention-based network that utilizes data augmentations to classify skin cancer through HS images.
    - **Advantages:** The proposed solution shows a lower false-negative rate than the state-of-the-art solutions and significantly reduces classification times compared to the best solutions in the literature.
    - **Disadvantages**: The proposed network was only tested on a dataset of 76 images, which may not be representative of all cases. Future research is required to enhance the network and assess different layer configurations.
    - **Main contributions**: The proposed network is an interesting and promising solution for medical HS images classification, especially due to the lower false-negative rate and lower classification times. The utilization of data augmentations and a fixed number of layers also provide more stable results compared to previous works.

Chapter 7

Intraoperative brain cancer contours assessment through deep learning, high-performance computing and hyperspectral imaging

In this doctoral thesis' Chapter 2, we mentioned that glioblastoma surgical resection is challenging for neurosurgeons. Tumour complete resection sweetens patients healing possibilities and prognosis, whilst disproportionate resection could lead to neurological deficits. Regardless, surgeons' eyesight hardly drafts the tumour's area and boundaries. Undoubtedly, most surgical processes result in subtotal resections. Histopathological testing might facilitate entire tumour elimination, though it is not feasible due to the time required for tissue breakdown.

Several studies reported tumour cells having unique molecular signatures and properties, which the minimally-invasive hyperspectral imaging we described in Chapter 2 can seize, delivering information concerning the observed tissue at the molecular level.

This chapter will address two pieces of research concerning glioblastoma targeting surgical contexts. The state of the art concerns the ML pipeline provided by the HELICoiD framework[11,12,39], which consists of algorithms such as K-means clustering and SVMs like the research this thesis described in the previous chapter.

First, this chapter describes research operating data augmentation and transfer learning to train the U-Net++ and the DeepLab-V3+ (Chapter 3). These models segment intraoperative glioblastoma HS images end-to-end, producing competitive processing times and segmentation results concerning the gold-standard procedure. Based on ground truths provided by the HELICoiD framework, it dramatically improved HSIs processing times, enabling the end-to-end segmentation of glioblastomas targeting the real-time processing to be employed during open craniotomy in surgery, thus improving the gold-standard ML pipeline. As we will acknowledge from this chapter's reading, the research measured competitive inference times concerning the standard CUDA environment offered by MATLAB 2020a. The HELICoiD fastest parallel version took 1.68 s to elaborate the

most prominent image of the database, whilst this methodology performs segmentation inference in 0.29 ± 0.17 s, hence being real-time compliant concerning the 21 seconds constraint imposed on processing. Eventually, it evaluated segmentation results qualitatively and quantitatively regarding the ground truth produced by HELICoiD.

Regardless, the first research presents limitations directly derived from the HELICoiD. First, not all pixels are histopathologically labelled; consequently, the pipeline-produced ground truth yields spurious results. Second, the HELICoiD ML pipeline contains unsupervised algorithms whose duration varies on the data, yielding an extended processing time which the previously mentioned research improved[11,12].

Brain tumour detection from HSI's immediate need is to extrapolate patterns and information to better highlight the tumour contours and aid surgeons. The partial supervision provided from the dataset limits supervised approaches. Furthermore, the first research proved that operating the gold-standard ground truth produced from the HELICoiD ML pipeline does not improve the results regarding segmentation performance but only concerning processing duration.

Consequently, this chapter presents a second research, constituting the last of this doctoral manuscript, comprising a novel attention-based self-supervised methodology to improve current research on hyperspectral medical imaging as a tool for computer-aided diagnosis. Namely, it describes the design of a novel architecture comprising the U-Net++ and the attention mechanism on the spectral domain, trained in a self-supervised framework to exploit the contrastive learning capabilities and overcome the dataset size problems arising in medical scenarios.

This research as well operated fifteen glioblastomas HS images from the HELICoiD dataset. Similarly, it applied extensive data augmentation and transferred learning to serve the end-to-end segmentation, achieving competitive segmentation results concerning the gold-standard procedure.

This chapter concentrates on AI and high-performance computing approaches this doctoral thesis researched to aid brain tumour surgical resection from hyperspectral imaging. The studies concern deep learning strategies, handling the dataset described in Chapters 2 and 3.

Close collaboration with Universidad de Las Palmas de Gran Canaria enabled the mentioned research investigations.

All the investigations in this chapter concern the dataset we illustrated in Section 2.9, together with its acquisition system, pre-processing and calibration stages. Furthermore, all operated the first test system described in Section 4.8.

In the following lines and sections, this chapter describes the methodologies found in the literature and the results applied to the problems mentioned above. Afterwards, it contains a brief exploratory analysis concerning Chapter 2's dataset from the HELICoiD European project employed in all the projects contained in this chapter.

The chapter describes the investigation strategies, the results with their discussion, and the ending remarks for all the investigations. These sections report the results, conclusions, and implications that advance the field based on current knowledge and our achievements. Accordingly, this chapter will cover investigations concerning all the theoretical aspects listed in chapters 2 to 4.

7.1. AI and HPC literature review concerning intraoperative brain tumour resection

AI solutions emerged as a tool to analyse and cluster different cancer types using HSI in recent years. Undoubtedly, HSIs could be more visually interpretable. Consequently, researchers usually carry out HS image analysis via AI approaches.

Previous chapters of this thesis extensively mentioned the literature focus on brain, skin, colon and oesophageal cancer diagnosis through ML, DL and HSIs[11,12,34,35,37,48,49].

Concerning intraoperative glioblastoma segmentation of HS images, research mainly emerged within the European project HELICoiD (HypErspectraL Imaging Cancer Detection)[39]. Researchers gathered an in vivo human-brain HS database on which they developed several ML pipelines, comprising Support Vector Machines (SVMs), K-Nearest Neighbours (KNN), Principal Component Analysis (PCA), and K-Means Clustering as the supporting algorithms[11,12]. The main challenge is retrieving a target ground truth to supervise the ML algorithms.

Neurosurgeons can only partially identify the tumour and its boundaries when diagnosing them with traditional imaging systems. Accordingly, HELICoiD-based ML studies comprised unsupervised algorithms to overcome this problem and automatically segment the intraoperative-captured HSIs.

7.2. AI-based segmentation of intraoperative glioblastoma hyperspectral images

Here, the chapter investigates the feasibility of supervised deep learning architectures, namely U-Net++ and DeepLab-V3+ (Chapter 3), as proof-of-concept to perform the automatic segmentation of fifteen intraoperative glioblastomas HS images retained from the HELICoiD database.

The investigation operated the ground truths coming from the HELICoiD ML-based pipeline for algorithms supervision as it currently represents the gold-standard procedure to retrieve a segmentation map of brain cancer. Undoubtedly, this procedure represents the only feasible way to label medical data when ground truth is unavailable via pathology-confirmed testing.

The main goal is to differentiate GB from healthy and other brain tissues, analysing the HSIs end-to-end to improve the time required to process differently supervised and unsupervised algorithms in a unique ML pipeline.

7.3. Deep learning methodology in brain cancer

This investigation trained three CNNs architectures to perform the semantic segmentation of the HS brain lesion images. It assessed the UNet++ and two versions of the DeepLabV3+ architecture, having as backbone structure a ResNet-50 but presenting alternatively 2D and 3D convolutions to perform semantic segmentation of the glioblastoma (Chapter 3). Likewise, it adopted the transfer learning strategy from Section 3.13 to improve the results of the learning-based architecture by exploiting features belonging to the previous training task. Consequently, all the listed architectures optimised on the HAM10000 dataset[105]. Furthermore, the study increased the training set statistical variability by applying the data augmentation procedure from Section 3.10 to the HS images. It used several methodologies, including geometric, filtering, colour transformations and pixel substitution. Notably, it either performed a linear combination of random pixels of tissues belonging to the same category or directly exchanged them. As we reported in this doctoral thesis' other investigations, data augmentation yields promising results in computer vision, significantly reducing overfitting[65].

Furthermore, it introduced salt-and-pepper white noise in random image bands to enlarge the training set. The augmentation procedure occurred iteratively and randomly. Namely, one of the data augmentation techniques was applied randomly to the training set with a certain probability. Each image, at maximum, received a predetermined number of augmentations. Such augmentation techniques did not apply to the test sets to reject the hypothesis of biased results.

The research modified all architectures to input $384 \times 384 \times 128$ HS images concerning height, width, and the number of wavelengths. It resized by cropping the HS GB images accordingly to fit the GPUs' RAM. It also introduced the L2 weights penalty in the loss function to reduce overfitting. Training settings included the cross-entropy loss function and the Adam method[57,93]. The learning step decrease by multiplication by the dropping factor, as we mentioned in the previous chapter's analyses. Training settings also included batch size, number of epochs, learning rate and drop factor period at 4, 200, 10^{-4}, and 80, respectively, for all the architectures. For the semantic segmentation models, the drop factor and L2 penalty were 0.75 and 5×10^{-3}.

7.4. Aggregated k-fold cross-validation and performance assessment

This research operated the cross-validation strategy described in Section 3.11. The original in-vivo human-brain HS database consisted of twenty-six images from sixteen adult patients[11,12,39]. Nine patients had histopathologically confirmed Grade IV glioblastoma, while the remaining seven patients were either affected by other types of tumours or other pathologies requiring a craniotomy. Regardless, only fifteen images offered the necessary ground truth quality obtained from the HELICoiD ML pipeline. Hence, the investigation randomly shuffled the original HS dataset comprising the 15 HS images and performed a leave-one-out cross-validation. Therefore, it trained the model k times and recorded its estimate for each test set. Consequently, the performance metrics for classification and semantic segmentation assessed the aggregated group of predictions, namely the union of each K-fold test set, generated from each DL architecture through the procedure (Section 3.11).

The investigation assessed the pixel-based classification performance. The assessment outcomes exploited accuracy, sensitivity, and specificity as described in Section 3.11. These metrics evaluated the joined prediction set of each architecture, which the investigation conveyed through the k-fold cross-validation strategy.

Furthermore, the GPU accelerated computing performance concerned the elapsed time for each image inference to compare the researched end-to-end methodology with the HELICoiD ML pipeline processing times.

7.5. Performance evaluation and discussion

The investigation trained and fine-tuned the three CNNs (Table 16) with a small-sized dataset and evaluated the architectures by employing leave-one-out cross-validation.

Table 16. U-Net++ and DeepLab V3+ (DLV3 presented in two versions with either ResNet-50 or ResNet-50 3D as backbone structures) segmentation results in terms of pixel-wise Accuracy, Specificity and Sensitivity

	Metrics	**U-Net++**	**DLV3 RN50**	**DLV3 RN3D**
Accuracy	Healthy Tissue	0.52	0.52	0.55
	Tumour Tissue	0.61	0.71	0.76
	Hypervascularized Tissue	0.26	0.45	0.42
	Background	0.28	0.36	0.33
Specificity	Healthy Tissue	0.74	0.74	0.73
	Tumour Tissue	0.72	0.85	0.83
	Hypervascularized Tissue	0.81	0.71	0.74
	Background	0.92	1.00	1.00
Sensitivity	Healthy Tissue	0.52	0.52	0.55
	Tumour Tissue	0.61	0.71	0.76
	Hypervascularized Tissue	0.26	0.45	0.42
	Background	0.28	0.36	0.33

Likewise, it adopted the taxonomy proposed in Chapter 2's Figure 13 as a trade-off for a comprehensive and medically appropriate diagnosis, well-suited for DL classifiers. The structure allows physicians to treat patients according to the highest healthcare criteria whilst retaining the best feasible segmentation performance.

Discrimination between tumour tissue and the other classes of lesions offered fair measures, meeting specificity ranging from 71% to 92% across both architectures and the considered tissue types (Table 16).

We report the DeepLab V3+ with ResNet-50 backbone structure achieving the best results in tumour tissue identification with 85% Specificity. Similarly, we report tumour tissue accuracy above 70% for both DeepLab V3+ architectures.

On the other hand, the multilabel segmentation retains sensitivity performance below 80%. Considering four distinct group classes induces each set to have fewer examples, eliciting sparse information regarding the inter-patient variability.

The research discussed in this chapter introduced a few essential matters. It designed an AI-based system to assist neurosurgeons in outlining the contours of glioblastomas during surgery, despite the limitation of the small-sized HS dataset. Notably, it explored a robust process to conceive DL algorithms and manage the small-sized dataset to answer the demand for a deep learning end-to-end pipeline to meet the real-time constraints of the surgical procedure[1,3,11,12].

Undoubtedly, CNN architectures are mainly composed of matrix computations that effectively fit the high-performance computing hardware employed in the HELICoiD project. The HELICoiD ML pipeline comprised pre-processing, band selection, SVM, KNN and K-Means clustering. It worked on HS images whose size varied from 329 × 377

towards 548 × 459 pixels at most and 128 bands. The ML pipeline yielded processing times ranging from 1.68 s to 2.68 s concerning the same test system employed in this work, both in single and multi-GPUs environments[11,12].

On the other hand, this investigation performed inference in single GPU mode on hypercubes sized 384 × 384 × 128, concerning height, width, and the number of selected bands, to fit the GPU RAM. It repeated measurement 100 times and yielded an average time of 0.29 s with 0.17 s standard deviation.

Consequently, this research proved the feasibility of the end-to-end methodology proposed to improve the gold-standard results. The unsupervised algorithms' non-deterministic processing times mainly limited the latter.

The tool should replace the current gold-standard procedure that exploits supervised and unsupervised ML algorithms, with the latter representing the computational bottleneck, to turn the modern DL algorithms into medical equipment. In the future, the equipment must comply with what policymakers like the FDA are moving towards approving concerning AI-based medical devices[3].

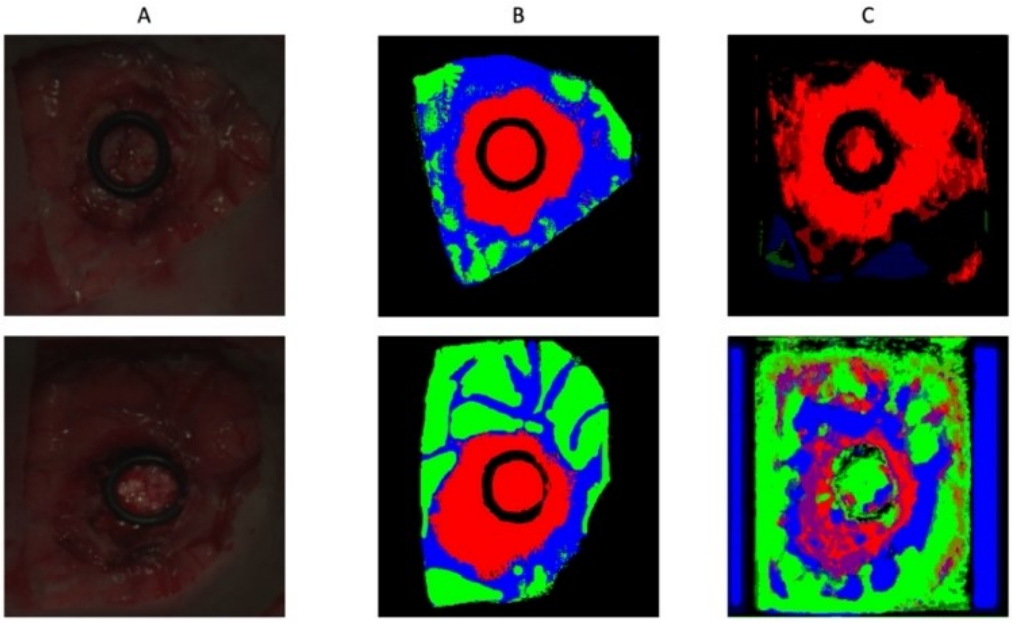


Figure 99. U-Net++ examples of semantic segmentation predictions compared to the cleaned HELICoiD ground truth. A) Synthetic RGB image. B) Cleaned HELICoiD ground truth. C) U-Net++ result

This research designed a proof-of-concept whose results strongly depend on the target ground truth produced via the gold-standard HELICoiD procedure. Figure 99 depicts an example of the semantic segmentation produced by the U-Net++.

Other works evaluated the segmentation metrics on the SAM result (Figure 99.A). Nonetheless, supervised CNN architectures need the entire mask to allow the gradient descent algorithm to meet convergence. Hence, this investigation not only evaluated Section 3.11's metrics over the ground truth produced by the HELICoiD pipeline and not over a few sets of pixels, targeting a future whole mask generation, but also employed modern DL methodologies on the HELICoiD dataset. Consequently, measuring segmentation metrics below a safety threshold, usually above 90% in medical contexts, is not necessarily a red flag sign. Indeed, the HELICoiD ground truth only sometimes labels every HS pixel correctly.

Nevertheless, the analysis reveals boundaries. The major one, which yields the others, concerns the dataset dimensions. Indeed, HS imaging is a powerful instrument compared to classical RGB pictures. Several studies highlighted that tumour cells present a unique molecular spectral signature and reflectance characteristics[33,48,49]. They allow the classification of pixels of tissues into different aetiologies. HS imaging systems gather brain-reflected and transmitted light into several wavelength ranges of the electromagnetic spectrum, enabling potential glioblastoma lesion tracing through DL algorithms. Indeed, each pixel contains meaningful information concerning the captured tissue properties. Regardless, not only are some brain portions transitioning from healthy to malignant tissue, but cancer, healthy and hypervascularised signatures might differ slightly. Each patient possesses a unique tissue signature which causes the test images to be very different from the training ones, increasing inter-patient variability.

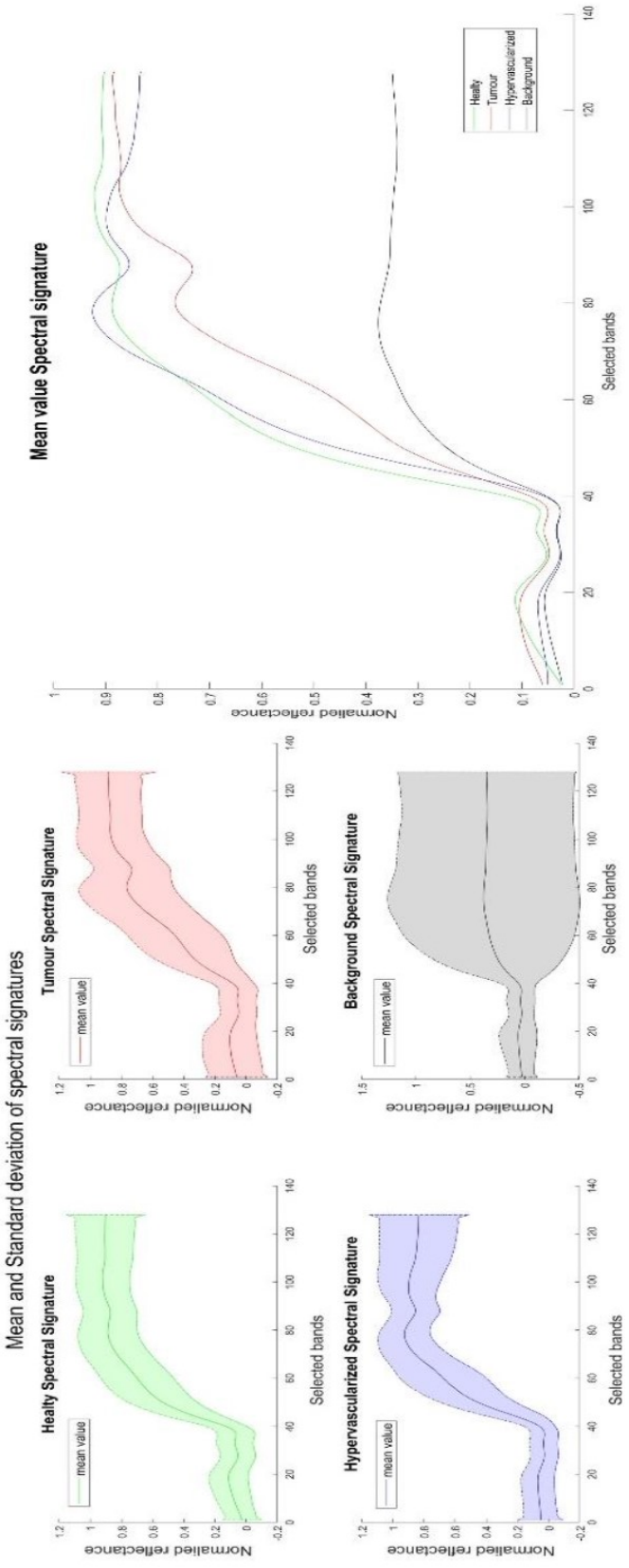


Figure 100. Average spectral signature and standard deviation of all brain HS images tissues and background

Figure 100 represents the spectral signature means and standard deviations of the investigated tissue classes concerning the labels assigned by the HELICoiD ML pipeline. Therefore, a broader dataset should cope with this issue and let CNNs concentrate more on the significant parts of the wavelengths, enhancing the segmentation outcomes of this work.

7.6. Ending remarks

Here, we discussed three DL architectures targeting the semantic segmentation of fifteen HS images belonging to the HELICoiD dataset. Modern DL methodologies allow the end-to-end segmentation of the HS images targeting the real-time processing to be employed during open craniotomy in surgery, thus improving the gold-standard ML pipeline. The investigation measured competitive inference times calculated with the standard CUDA environment offered by MATLAB 2020a, without a custom implementation, concerning the HELICoiD processing times. HELICoiD's fastest parallel version took 1.68 s to elaborate the most prominent image of the database, whilst the described methodology performs segmentation inference in 0.29 ± 0.17 s, thoroughly satisfying the real-time constraint, classifying the images in less than 21 seconds.

Eventually, it compared segmentation results qualitatively and quantitatively with the ground truth produced by the HELICoiD project.

7.7. Attention-based self-supervised U-net++ for the segmentation of intraoperative glioblastoma hyperspectral images

This second investigation from this last chapter presents a novel attention-based self-supervised methodology to improve current research on hyperspectral medical imaging as a tool for computer-aided diagnosis.

Namely, it concerns the design of a novel architecture comprising the U-Net++ and the attention mechanism on the spectral domain, trained in a self-supervised framework to manipulate contrastive learning and overcome the dataset size problems arising in medical scenarios.

As the previous research, it operated the fifteen glioblastomas HS images from the HELICoiD dataset.

Lately, Self-Supervised Learning (SSL) is emerging as a framework to operate small-sized datasets with limited labelling[1]. SSL algorithms function by distilling representative characteristics from unlabelled and unstructured data. Accordingly, SSL-trained networks learn shared and distinct features in a contrastive manner, surpassing supervised architectures on many domains[1]. At the time of writing, no prior work exists concerning medical brain cancer HS images and SSL.

Self-supervised learning is a machine learning paradigm in which a model is trained to predict a property of the input data, without being explicitly labeled with the correct output. Instead, the model is given a set of unlabeled data and learns to predict some property of the data, using only the input data itself as supervision.

For example, a self-supervised learning model might be trained to predict the position of a randomly masked word in a sentence, given the rest of the sentence as input. The model learns to predict the masked word based on its context within the sentence, without being given any explicit labels for the masked word. This type of self-supervised learning can be used to pre-train a model for downstream tasks, such as natural language processing or image classification.

Self-supervised learning has become increasingly popular in recent years as a way to improve the performance of machine learning models, particularly in cases where labeled data is scarce or expensive to obtain. It has also been used to improve the generalization and robustness of models, by encouraging them to learn more about the structure and patterns of the data.

Hence, here we will discuss the feasibility of a novel self-supervised deep learning architecture, an attention-based U-Net++, as a proof-of-concept to perform the automatic end-to-end segmentation of fifteen intraoperative glioblastomas HS images retained from the HELICoiD database, and explained in detail in the previous investigation.

7.8. Attention-based U-net++ and self-supervised STEGO framework

This chapter proposes a novel deep learning architecture, namely the attention-based U-Net++, comprising the attention mechanism along the spectral dimension and the well-known U-Net++ architecture (Chapter 3) along the spatial frame.

During the last years, transformer-based architectures have proven themself worthy of investigation in vision contexts[100]. Consequently, this investigation foresaw the alteration of the U-Net++ architecture, designing a parallel path analysing the spectral signatures of the HS cube after a first pooling step, set to reduce the networks' parameters (Figure 101.d.a). Hence, the novel architecture merged the attention-based neural path and the U-Net++ averaging their outcomes.

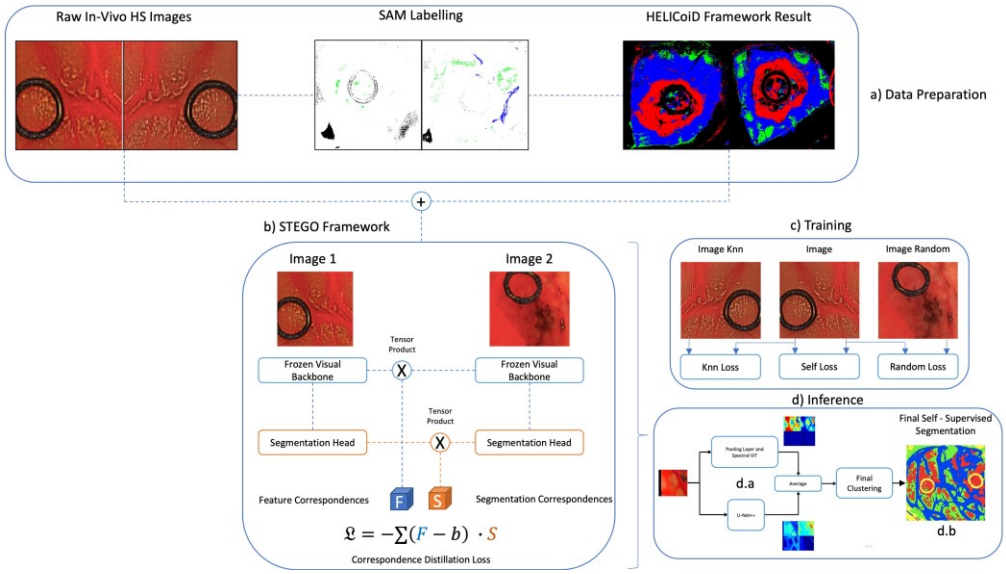


Figure 101. Data preparation, Self-Supervised STEGO framework, and inference result

Furthermore, this investigation used the attention-based U-Net++ inside the STEGO (Self-supervised Transformer with Energy-based Graph Optimisation)[110]. This novel framework distils unsupervised features into high-quality discrete semantic labels. The training settings carefully modified the algorithmic structure, developed from scratch in MATLAB 2020a, to accept the glioblastoma HS images.

Regarding Figure 101, STEGO extracts the features from the backbone architecture, the U-Net++ path, and later retains the segmentation results corresponding to the selected image characteristics (Figure 101.b). By adopting a contrastive learning methodology, the network learns feature correspondences in an unsupervised fashion. At its core, STEGO yields a novel contrastive loss function (Figure 101.b) designed to encourage features to form compact clusters while preserving their relationships across the entire dataset during the training (Figure 101.c)[110].

The proposed methodology could enhance hyperspectral medical research, overcoming labelling and dataset size challenges. At the time of writing, it is the first time a self-supervised structure as the one proposed in this chapter operates with medical hyperspectral images.

7.9. Discussion on performance

The investigation evaluated the SSL quantitative and qualitative results concerning the SAM labelling retrieved from the HELICoiD dataset since it represents the safest and most honest way of performance assessment.

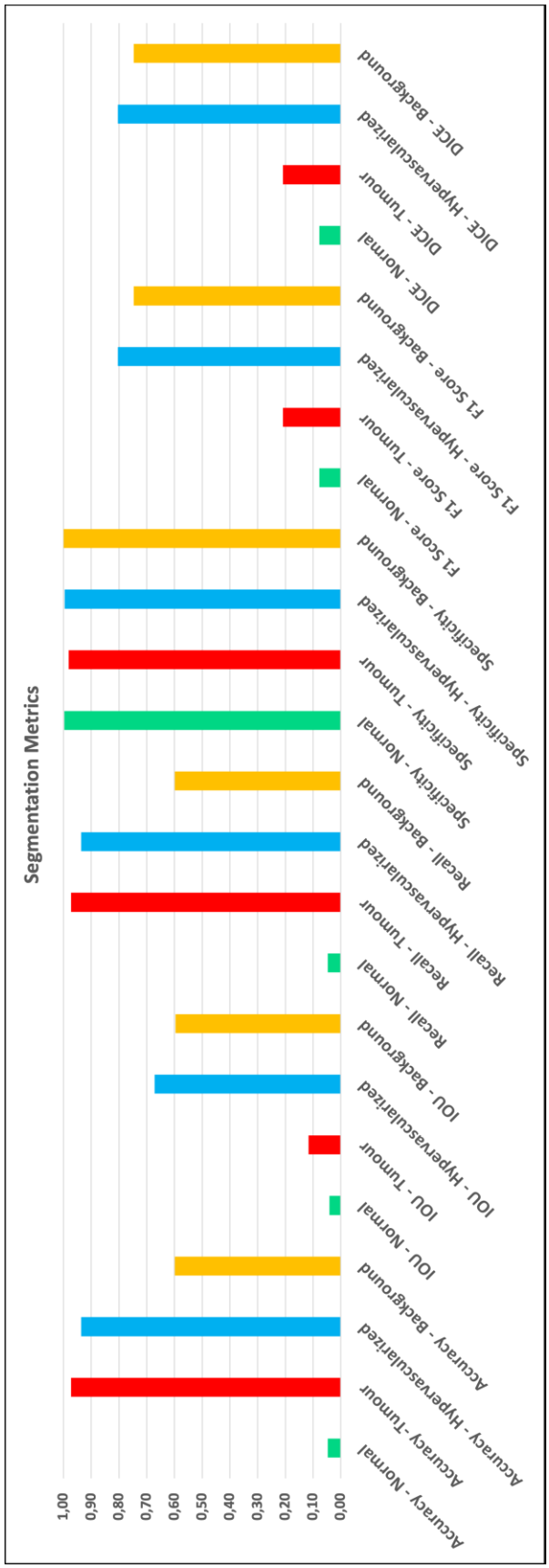


Figure 102. Self-Supervised Learning segmentation results

Figure 102 exhibits the set of evaluation metrics considered in this study. Although the attention-based U-Net++ retains high specificity, recall, and Accuracy concerning the tumour class, it yields yet optimal results for the healthy class. On the other hand, Figure 100 probably accurately explained the cause behind this misclassification. We report that the architecture misclassifies the healthy signatures for malignant ones, and the same happens for the background.

Concerning the Hypervascularised tissue, the self-supervised architecture proposed in this study can precisely outline the class. It is worth noting that hypervascularised tissues represent areas full of blood that nourish brain tumours and could represent other risk zones. Furthermore, the investigation measured competitive inference times compared to the standard CUDA environment offered by MATLAB 2022a, without a custom implementation, concerning the HELICoiD processing times.

As the previous research reported, HELICoiD's fastest parallel version took 1.68 s to elaborate the most prominent image of the database. On the other hand, STEGO SSL performs segmentation inference in 0.34 ± 0.25 s, thoroughly meeting the real-time requirement imposed by the HELICoiD European project, ranking the HSIs in less than 21 seconds.

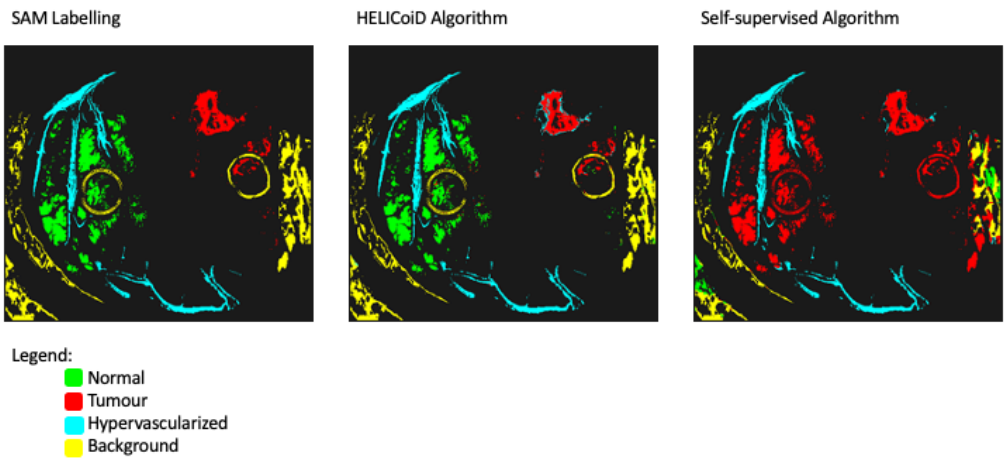


Figure 103. Visual comparison of the ground-truths derived from the algorithms considered (HELICoiD, SAM and SSL)

Eventually, Figure 103 reports the visual comparison of the ground truth and estimated segmentation maps yielded by the diverse algorithms discussed in this doctoral manuscript.

7.10. SSL final remarks

This last investigation addressed a novel DL methodology targeting the end-to-end semantic segmentation of fifteen HS images belonging to the HELICoiD dataset. Namely, it researched an SSL algorithm to train an innovative segmentation architecture.

The investigation proposed methodologies to enable the end-to-end segmentation of the HS images targeting the real-time processing to be employed during open craniotomy in surgery, thus improving the gold-standard ML pipeline. Additionally, it offered competitive results in terms of pixel-wise classification. We measured competitive inference results for identifying unhealthy tissue, exceeding 90% in Accuracy, specificity, and recall.

The framework performs poorly when the architecture classifies healthy and background image portions as tumours. It is an open research topic which academia should aim to improve and clarify in the evolution of this algorithm. The proposed SSL methodology could improve medical hyperspectral image segmentation, thus improving the state of the art computer-aided diagnostic systems.

At the time of writing, no prior work on SSL applied to medical hyperspectral images was carried out, but only on larger datasets concerning remote sensing applications.

7.11. Main contributions summary

Here, the thesis proposes a list of main contributions deriving from the pieces of study described in the earlier sections.

- *AI-based segmentation of intraoperative glioblastoma hyperspectral images*
    - **Addressed problem**: Semantic segmentation of intraoperative glioblastoma hyperspectral images in real-time.
    - **Proposed solution**: The study proposes three DL architectures for real-time processing of hyperspectral imaging to differentiate between tumor and healthy tissue based on spectral information. CNNs are utilized to classify hyperspectral images into tumor and non-tumor regions.
    - **Advantages**: The proposed method achieves competitive inference times and satisfies real-time constraints, providing end-to-end pixel-wise classification. It achieves 85% specificity and 70% accuracy for the tumor tissue class.

  - **Disadvantages**: The method's performance is poor in classifying healthy and background image portions as tumors. It relies on hyperspectral imaging, which is not widely available in clinical settings, and its performance may be affected by the quality of the hyperspectral data.
  - **Main contributions**: Improved gold-standard ML pipeline form European HELICoiD project for intraoperative glioblastoma segmentation using hyperspectral images.
- *Attention-based self-supervised U-net++ for the segmentation of intraoperative glioblastoma hyperspectral images*
  - **Addressed problem**: End-to-end semantic segmentation of intraoperative glioblastoma hyperspectral images in real-time.
  - **Proposed solution**: Innovative SSL algorithm to train a novel segmentation architecture.
  - **Advantages**: The proposed method achieves competitive pixel-wise classification results and competitive processing times.
  - **Disadvantages**: The method's performance is poor in classifying healthy and background image portions as tumors, as mentioned previously.
  - **Main contributions**: The study proposes a SSL methodology that could improve medical hyperspectral image segmentation, and it is the first work on SSL applied to medical hyperspectral images.

Chapter 8

Conclusions

Several aspects of the future of personalised medicine require compelling technologies able to elaborate vast amounts of data in real-time. This doctoral thesis demonstrated the crucial role that AI and HPC play in medical contexts with their ML and DL paradigms.

The educational path reviewed in this manuscript described a collection of works following the data science process mentioned in the introduction (Figure 3). Namely, the thesis addressed projects concerning different data types and sources, ranging from the HypErspectraL Imaging Cancer Detection (HELICoiD) brain cancer images to the clinical and radiology data related to the SARS-CoV-2 disease.

The document described the data collection, cleaning, and exploration strategies to gather knowledge and support decisions. Notably, it discussed state of the art AI and HPC methodologies. We understood how to enlarge the statistical variance of the information at our disposal, envisioning either standard data augmentation techniques or generative models that make AI systems robust, enabling disturbance rejection to adversarial attacks.

The thesis operated a collection of machine and deep learning models evaluated according to specific performance metrics.

Eventually, this doctoral thesis comprised a set of investigations addressing GPU deployment through contemporary CUDA libraries to meet the real-time constraints that medicine demands.

Medical AI applications, as they mature, face many challenges:

1. Clinical contexts often need better datasets, making it hard to exploit complex DL architectures. Not only do DL architectures require vast amounts of data to extract functional patterns, but they also need HPC hardware. Accordingly, the more complex the model, namely presenting larger structures with many parameters, the more we need to exploit performing hardware to carry out computations
2. Medical data present challenges specific to its domain. For example, different experts may even give contrasting opinions regarding a diagnosis. Hence, we must set hierarchically structured and standardised evaluations to enable AI-based diagnosis

3. From a regulatory perspective, clinical AI systems must be certified before large-scale deployment

Accordingly, the manuscript proposed model deployment approaches employing low-level hardware, Nvidia GPUs, and CUDA/C code development to accelerate the AI algorithms and allow the design of blueprints. Eventually, Modern AI should translate to blueprints for policy-maker entities such as the Food and Drug Administration (FDA) or the European Commission to evaluate them concerning compliance with certifications currently under investigation (Artificial Intelligence Act), including time-sensitive and performance criteria. Regulators have struggled to interpret existing frameworks for oversight concerning perceptive algorithms, whose functioning can change with ongoing training and optimisation and whose output we often need help explaining. Many clinical applications of AI are seeking regulatory approval, and governments' new interpretation is a substantial step toward rules that protect patients without inhibiting innovation. This document described the same challenges encountered during the design and development of the projects accomplished in this PhD school.

This doctoral thesis addressed artificial intelligence applied to diverse medical data, especially Hyperspectral Images (HSIs), and matured around state of the art. Notably, the investigations designed novel deep learning approaches when the literature only discussed standard machine learning processes. Not only was a robust artificial intelligence methodology applied to the medical context, but the investigations also engineered novel GPU approaches and frameworks to embed or accelerate the devised models, complying with the time-sensitive criteria required for industry translation. The work presented in this manuscript was carried out thanks to tight and robust collaborations outside the academy, particularly with the Fondazione IRCCS Policlinico San Matteo of Pavia, the University of Las Palmas de Gran Canaria and the Innovation Center Computer-Assisted Surgery (ICCAS) of the University of Leipzig.

The thesis described three main groups of investigations: SARS-CoV-2, epidermal lesions assessment, and intraoperative glioblastoma tumour boundaries detection.

The first group of investigations addressed the proposed solutions to counteract the SARS-CoV-2 pandemic, accounting for three investigations. Chapter 5 concentrated on the statistical and AI approaches to counteract SARS-CoV-2 spreading. The studies addressed the operation of specific diagnostic measurements, also used to collect the dataset described in Chapter 2, the classification of LUS clips and assessing patients for SARS-CoV-2 positivity through blood tests.

Based on an observational cohort of Covid-19 patients evaluated at the Fondazione IRCCS San Matteo University Hospital in Pavia (Italy), the first study demonstrated that AaDO2 is a valuable clinical parameter to stratify the evolutionary risk of patients with Covid-19. At the time of writing, it was the first investigation evaluating the function of AaDO2

measured at hospital admission from the ABG analysis to characterise Covid-19 patients better. ABG testing is readily available in the emergency setting, giving crucial information about pulmonary involvement and respiratory function. Hence, the study evaluated the role of the alveolar-to-arterial oxygen difference, particularly in Covid-19 patients with P/F values ranging between 300 and 400. According to the literature, this range represents patients without significant acute lung injury. Nonetheless, this study proved the opposite. Indeed, although this subgroup of patients possessed typical P/F values, AaDO2 was higher than regular. Moreover, more than half of these patients subsequently required oxygen therapy support.

Interestingly, patients who subsequently needed oxygen support had a more severe extent of lung involvement, as assessed by LUS, than those who did not. Indeed, literature reported that patients with Covid-19 pneumonia often do not register dyspnoea, despite extreme hypoxemic values. Academia defined this clinical presentation as silent hypoxemia or happy hypoxia, with physical signs that may either overestimate or underestimate patient discomfort.

In conclusion, patients might have presented with few clinical signs and symptoms, a chest X-ray not indicating the significance of lung involvement, and P/F still within normal limits. Consequently, it is essential to obtain elements that predict the risk of subsequent clinical worsening. This first research described the importance of the data collection, which produced Sections 2.3 and 2.11's database. Physicians relied on the analysis this section described to gather data and LUS clips, which set the stage for the research described in this chapter's subsequent sections.

Nevertheless, we should acknowledge some limitations of this study. The retrospective single-centred configuration leads to missing information and unavoidable biases in specifying and recruiting participants. Fondazione IRCCS Policlinico Hospital of Pavia gathered the data in contingency times concerning the SARS-CoV-2 pandemic, and the sample size was relatively small. Despite these limitations, the study reflects an actual world clinical scenario in the ED during a pandemic outbreak. The promising results open the doors for further validation in future multi-centred extensive prospective studies to consolidate LUS and AaDO2 assessments.

The second investigation leaned on the data collected from routine hospital operations between 1 March and 30 June 2020, featuring the confirmed clinical parameters crucial in the first research described. The research proved the feasibility of developing reliable algorithms to diagnose SARS-CoV-2 with high classification performance.

Physicians examined routine blood tests, clinical history, symptoms, arterial blood gas analysis, and lung ultrasound quantitative examination. The investigation produced two diagnostic tools for Covid-19 detection and oxygen therapy prediction. In addition to what other studies had reported, it demonstrated how to estimate dangerous dyspneic scenarios. Namely,

whether the subjects at the ED need CPAP or invasive aided ventilation, and this prediction is noteworthy to handle resources in contingency times. It yielded promising classification results with F1 score levels meeting 92% and engineered a user-friendly interface for healthcare providers during daily screening operations. This research proved machine learning models as a potential screening methodology during contingency times.

The close and stable collaboration with the IRCCS Policlinico San Matteo's ED of Pavia granted highly reliable clinical data for the study. It made it possible to develop two artificially intelligent systems, one of which the personnel tested as a supporting decision-making device in a real-world clinical scenario after we equipped it with a GUI.

The novelty of the designed approach stood in the next passage:

- A careful clinical features collection: this thesis' classifiers operated on the features that physicians employed during triaging and daily clinical operations, whose importance was stressed in the first investigation related to SARS-CoV-2
- Extensive and robust data analysis before ML clustering
- Exploitation blood tests to assess patients rather than imaging data
- Assessment of patients' need for oxygen therapy to carefully engage limited resources in contingency scenarios
- A quantitative lung involvement examination to produce robust results: studies report lung ultrasound examination as a fast, cheap, and agile tool to assess patients' lung involvement

The third and last investigation about Covid-19 concerns LUS frames classification to assess the severity of lung involvement. This third SARS-CoV-2-related research engineered a highly reliable diagnostic instrument to satisfy exhausted medical personnel's growing request for cheap and trustworthy detection systems. With close collaboration with Fondazione IRCCS Policlinico San Matteo's ED, the investigation leaned on validated LUS data.

The research comprised modern DL methodologies, data augmentation processes, and transfer learning to grade people's lungs operating documented scoring scales, which the investigation extended with pleural line information. The investigation relieved the severe drawbacks of data heterogeneity (tolerable sensitivity causing lack of treatment for patients and cross-contamination) and enhanced currently accessible state-of-the-art in Covid-19 detection employing LUS data.

This study provided a method for sidestepping the AI challenges debated by the literature about the ranking inconsistencies between ultrasounds due to different doctors examining different lungs at the same disease stage. Notably, the Fondazione IRCCS Policlinico San Matteo ED inspected every test to homogeneously appoint lungs of the same disease stage with the same score.

Ultrasound requires substantial expertise to reach diagnostic reliability – high sensitivity and overall accuracy. This research developed a DL-based system to automatically detect Covid-19 pneumonitis marks in LUS frames and rank them concerning two standardised scales with innovative, reliable, and revolutionary results. Hence, employing methodological hyperparameter tuning, the thesis produced state-of-the-art results meeting F1 score levels, averaged over the number of classes considered, exceeding 98%, and manifesting stable measurements over precision and recall. Also, the architectures ranked the LUS frames in less than one second, proving compliance with real-time requirements.

The second group of investigations addressed epidermal tumour diagnosis, accounting for five AI and HPC applications. Chapter 6 concentrated on AI and high-performance computing approaches this doctoral thesis researched to counteract epidermal tumours from hyperspectral imaging. The studies on machine and deep learning strategies handled the dataset presented in Section 2.7.

The first of the five pieces of research presented a hyperspectral acquisition system engineered to gather diagnostic clinical data concerning skin cancer. It is enhanced by a linear synchronous motion, an appropriate illumination system, a 3D-printed circular crown containing targeting and distancing emitting diodes, and software modules supported by open-source packages. The hyperspectral system enables image collection with any GigE-compliant hyperspectral pushbroom camera. Furthermore, the investigation validated the architecture to check synchronisation between motor and camera frame rate, calibration, and capturing repeatability. In the future, the research aims to collect an online database of clinical hyperspectral images.

The main contribution of this work is to serve as a guide for any research group working on hyperspectral technologies. All the sections report details to accurately capture spectral information and techniques to validate the correct operation of the system. First, the whole system works with any GenICam protocol-compliant camera. Secondly, the thesis operated cheap and promptly available hardware and open-source software to enable research groups to work with hyperspectral systems most efficiently. Indeed, all software modules used in this development are open source, allowing high flexibility and representing a lower-cost approach compared to market solutions.

The second research, instead, laid the foundations for the remaining three. This research presented a parallel classification framework based on HSI exploiting the K-means and the SVM algorithms to perform an automatic in-situ PSL identification. The framework used an in-vivo dataset, and the algorithms' parameters tuning happened in MATLAB for later implementation of the processing framework on HPC platforms.

Several parallel versions, exploiting multicore and many-core technologies, have been developed to ensure a real-time classification.

This preliminary study demonstrated the potential use of HSI technology to assist dermatologists in the discrimination of different types of PSLs.

However, additional research must occur to validate and improve the results obtained before being used during routine clinical practice using a real-time and non-invasive handheld device. Notably, a multicenter clinical trial with more patients and samples in the database will be necessary to validate the proposed approach further.

Then, this thesis researched strategies to overcome AI challenges concerning dataset size. In this context, the doctoral activity proposed a convolutional DCGAN architecture to generate HS medical data, particularly for skin lesion analysis, by operating a small-sized dataset to train the framework. It adopted the FID metric to evaluate the similarity between the real and the synthetic data. Outcomes yielded a 17.37 FID, which indicates sound synthesis and similarity between the distributions of the two datasets.

Additionally, a ResNet-18 was trained only on synthetic data and tested on authentic images. The accuracy, precision, recall, and F1 score were all above 80%, demonstrating that the synthetic data and the authentic images are comparable. Finally, the thesis compared the spectral signatures qualitatively and quantitatively.

The literature reports only one work considering medical HS data. Regardless, this work validated the results only in terms of visual similarity between the mean spectral signature of original and generated images.

Future research lines will investigate novel GAN architectures for medical HS images. Finally, the conditional GAN could produce different tumour etiologies besides benign and malignant ones.

Eventually, the last two investigations, belonging to the second group of works analysing strategies to counteract and assess skin cancer from HSIs, concerned DL architectures and GPU deployment.

Cursed by the absence of large datasets, it took some time for HSI-based applications to become feasible in terms of tasks employing classical RGB or multispectral images. Indeed, the studies considered by the authors of several systematic reviews consisted of databases with significant data, thus highlighting the diagnostic performance plateau reached. Consequently, classification techniques for HSI often exploit transfer learning and data augmentation to improve classification performances in different research fields. Algorithms employing HS images usually comprise the classical pixel-wise models we mentioned in the second research of this chapter. Even though the algorithms only work with spectral and not spatial information, their sensitivity and specificity concerning Malignant Melanoma (MM) and Non-Melanoma Skin Cancer (NMSC) evaluated through leave-one-out lie around 80% and 77%, recently improved to 87.5 and 100%, respectively.

This investigation responded to the market for AI clinical applications and the need for computational power to assist it in engineering a handheld instrument equipped with a low-power GPU. The tool should replace the expensive and time-consuming gold-standard diagnostic procedure to turn modern DL algorithms into medical equipment.

This thesis conceived a blueprint dermatological device to improve the global accessibility of epidermal screening at the expert level. Expert dermatologist classification accuracy of epidermal lesions usually depends on the number of classes considered. At most, it reaches 85% in a malignant-benign classification scenario. The gold-standard procedure implies clinical and dermoscopic inspection, followed by biopsy and histopathological examination. In other words, the subjective nature of the inspection biases the classification accuracy measurement of malignant lesions. Undoubtedly, physicians only diagnose lesions already marked as suspicious.

The fourth investigation of the second group designed CNNs to attain and enhance well-known dermatologist human-level classification performance concerning specificity, sensitivity, and accuracy. At the time of writing, no research existed yet concerning HS skin cancer image segmentation to produce a mask to inform doctors about lesion boundaries. Similarly, other studies mainly focused on producing high-end results considering classification scenarios with unessential clinical applicability.

This thesis was eager to respond to the demand for an AI-based pipeline to assist or replace the expensive and time-consuming gold-standard procedures. Accordingly, it deployed a semantic segmentation network on a low-power Nvidia Jetson GPU device targeting a portable instrument containing an HS camera. The designed proof-of-concept AI system can classify and segment epidermal lesions in, at most, 1.21 s, and expert professionals could use the future implementation in real-world clinical scenarios.

Nonetheless, the study exhibits limitations. The main limitation is related to dataset size, which in turn produces others. Indeed, HS imaging is a powerful tool compared to classical RGB pictures. Chromophores characterise skin's spectral properties and allow lesion clustering into different etiologies. HS imaging systems gather skin-reflected and transmitted light into several wavelengths ranges on the electromagnetic spectrum, enabling potential skin-lesion differentiation through machine and DL algorithms. Indeed, each pixel contains meaningful information concerning an object's properties. Not only are some lesions in the dataset transitioning from benign to malignant lesions, but lesions and skin signatures might differ slightly.

Eventually, the second group's last research concerns a different learning strategy to overcome the abovementioned problems. Accordingly, this thesis addressed the attention mechanism and ViT to assess whether a more complex and perceptive learning mechanism could cope with dataset size challenges and deliver better pattern extraction. The proposed network is designed and validated using the skin HS dataset, adopting the K-fold cross-validation technique to produce robust results. The model was trained by augmenting the training set at runtime and then performing the tests only on the real images, considering the number of folds equal to 10.

The results emphasise that the attention-based mechanism is an interesting and promising solution for medical HS classification, since the

false negative rate is half compared to the state-of-the-art. Eventually, the classification times are significantly lower than the best solutions proposed in the literature. Finally, the proposed network adopts a fixed number of layers whose number of mathematical operations is deterministic, making the measured processing time more stable than the results reported in previous works.

Future research will focus on improving the proposed network and evaluating different layers configurations.

Eventually, this doctoral thesis addressed the last group of two investigations targeting intraoperative HS glioblastoma images.

The first study, and nineth of this thesis, discussed three DL architectures targeting the semantic segmentation of fifteen HS images belonging to the HELICoiD dataset. Modern DL methodologies allow the end-to-end segmentation of the HS images targeting the real-time processing to be employed during open craniotomy in surgery, thus improving the gold-standard ML pipeline. The investigation measured competitive inference times calculated with the standard CUDA environment offered by MATLAB 2020a, without a custom implementation, concerning the HELICoiD processing times. HELICoiD's fastest parallel version took 1.68 s to elaborate the most prominent image of the database, whilst the described methodology performs segmentation inference in 0.29 ± 0.17 s, thoroughly satisfying the real-time constraint, classifying the images in less than 21 seconds.

The last investigation addressed in this doctoral path concerns a novel DL methodology targeting the end-to-end semantic segmentation of fifteen HS images belonging to the HELICoiD dataset. Namely, it researched a Self-Supervised Learning (SSL) algorithm to train an innovative segmentation architecture.

The investigation proposed methodologies to enable the end-to-end segmentation of the HS images targeting the real-time processing to be employed during open craniotomy in surgery, thus improving the gold-standard ML pipeline. Additionally, it offered competitive results in terms of pixel-wise classification. We measured competitive inference results for identifying unhealthy tissue, exceeding 90% in accuracy, specificity, and recall.

The framework performs poorly when the architecture classifies healthy and background image portions as tumours. It is an open research topic which academia should aim to improve and clarify in the evolution of this algorithm. The proposed SSL methodology could improve medical hyperspectral image segmentation, thus improving the literature on computer-aided diagnostic systems.

At the time of writing, no prior work on SSL applied to medical hyperspectral images was carried out, but only on larger datasets concerning remote sensing applications.

In conclusion, this thesis has explored the use of high-performance computing solutions in artificial intelligence, specifically in the medical domain. It has been demonstrated that the increasing size of AI models

requires powerful computing resources to ensure real-time performance. Therefore, future research in this area must focus on developing low-power consumption solutions.

Additionally, the thesis has presented diverse computer vision models for medical tasks, including skin cancer diagnosis, brain cancer contours delineation, and SARS-CoV-2 assessment and severity scoring. The state of the art AI techniques employed in these models have shown proficient diagnostic performance. However, it is essential to note that the variability of these models strictly depends on the training data available. The available data depends on the research budget and strategic planning. Therefore, the medical community must invest in data collection and annotation efforts to improve the performance of these models.

Finally, this thesis has highlighted the importance of international collaborations in AI and medicine. The collaborations with international universities have allowed for sharing expertise and resources, resulting in the development of accurate and robust models. Table 17 summarizes the main contributions of this doctoral thesis.

Table 17. Doctoral thesis summary of main contributions and detailed description of advantages, disadvantages of proposed solutions to specific problems.

Research topic	Addressed problem	Proposed solution	Advantages	Disadvantages	Improvement concerning the state of the art
State-of-the-art review for SARS-CoV-2 pandemic management	Lack of a reliable and accurate diagnostic tool to diagnose SARS-CoV-2 and score its pneumonitis severity	CT scan, PCR, IgM-IgG bloodwork, and chest X-ray	CT scan radiation exposure and cross-contamination, PCR false negatives in early infection, and low sensitivity of Chest X-Ray	CT scan has high accuracy, PCR has high specificity, and Chest X-ray is widely available	None
Alveolar-arterial difference and lung UltraSound to help the SARS-CoV-2 clinical decision-making	SARS-COV-2 patients often require prompt diagnosis and risk stratification. Also, predict patients need for aided ventilation	Using A-a gradient and LUS to diagnose and stratify risk for pandemic management	A-a gradient and LUS can be obtained quickly and safely, provide valuable diagnostic and prognostic information, and	A-a gradient may lack specificity for SARS-COV-2, LUS requires experienced operators, and small sample size limits generalizabili	A-a gradient and LUS can provide important information for diagnosing and risk stratifying SARS-COV-2 patients, especially in

			combining them can improve diagnostic accuracy	ty	resource-limited settings. Study found the A-a gradient and LUS combination had 83.6% sensitivity and 90.5% specificity, with 90.7% positive predictive value (PPV) and 83.5% negative predictive value (NPV) in predicting the need for high flow of oxygen
Machine-learning-based SARS-CoV-2 and dyspnoea prediction systems for the emergency department	Developing an accurate and reliable system to predict SARS-COV-2 and oxygen therapy requirement in emergency department patients	A machine-learning-based prediction system that uses a combination of clinical and laboratory data to predict SARS-COV-2 and oxygen therapy requirement	The model has an area under the curve exceeding 93%, recall for SARS-COV-2 detection of 96%, F1-score for SARS-COV-2 detection of 92%, and F1-score for oxygen therapy prediction of 83%. The precision for SARS-COV-2 detection and oxygen therapy prediction is continuously above 80%	The study is limited to a single hospital and further testing is needed to determine its generalizability to other hospitals or populations. It also requires access to laboratory data, which may not be available in all settings	The model has improved results compared to existing models that use a smaller, unbalanced dataset and fewer features. It uses both clinical and laboratory data, which increases accuracy and reliability, and has the potential to aid clinical decision-making in emergency departments. The

					machine-learning algorithm can also be easily updated as new data becomes available
Deep learning and Lung UltraSound for SARS-CoV-2 pneumonia detection and severity classification	Lack of reliable and accurate and prompt diagnostic tools for SARS-CoV-2 pneumonitis detection and severity classification using traditional methods	A deep learning-based model using Lung Ultrasound (LUS) images for pneumonia detection and severity classification	LUS is non-invasive and widely available, provides high accuracy and sensitivity, reduces exposure to ionizing radiation, enables comprehensive diagnosis of SARS-COV-2 pneumonia using LUS images, and allows for high accuracy in both pneumonia detection and severity classification, reducing diagnosis time	LUS requires substantial expertise and high-quality data, which may not be widely available in all clinical settings, particularly in resource-poor regions, although LUS is cheaper than other technologies. There is also a lack of large-scale data for model training and a need for expert annotation of LUS images	The use of LUS data improves the accuracy and efficiency of SARS-CoV-2 pneumonitis diagnosis and enhances the state-of-the-art SARS-CoV-2 detection. The proposed model provides a comprehensive diagnosis of SARS-COV-2 pneumonia and outperforms traditional methods in accuracy and time efficiency
Review of Hyperspectral imaging in skin cancer detection	Existing imaging techniques are insufficient in providing accurate diagnosis, and the gold standard ABCDE rule followed by histopathologi	The majority of techniques used for skin cancer diagnosis using hyperspectral imaging (HSI) involve machine	Non-invasive techniques for skin cancer diagnosis offer high accuracy, enhanced accuracy, and earlier and more	Non-invasive techniques for skin cancer diagnosis can be costly, and further validation is needed since limited research is available.	None

	cal examination is time-consuming and invasive	learning pipelines. There is a small presence of convolutional neural networks (CNNs) used for skin cancer diagnosis using HSIs. Early solutions attempted to reproduce the ABCD rule with shallow neural networks	accurate diagnosis	Additionally, there are only small datasets available, which can limit the accuracy and generalizability of the models	
Hyperspectral imaging acquisition set-up for medical applications	There is a limited availability of efficient hyperspectral imaging systems for medical applications, which can limit the adoption and advancement of this technology for skin cancer diagnosis. Additionally, there is a limited availability of skin cancer hyperspectral dataset for research, which can limit the development and testing of new models	Hyperspectral acquisition system engineered to gather diagnostic clinical data concerning skin cancer	Using low-cost imaging techniques for skin cancer diagnosis can be more cost-efficient and accessible, as they rely on readily available hardware and open-source software	One limitation of using low-cost imaging techniques for skin cancer diagnosis is that they have not yet been extensively applied in a real-world scenario, where large and diverse datasets can be gathered. This can limit the ability of these techniques to generalize to a wider range of skin cancer cases and populations	Enhanced efficiency, potential for more cost-efficient and widely accessible systems with respect to state of the art

Parallel classification pipelines for skin cancer detection exploiting hyperspectral imaging on hybrid systems	Lack of real-time and accurate diagnostic systems for skin cancer detection	A parallel classification framework based on HSI using K-means and SVM algorithms for automatic in-situ PSL identification	One advantage of using real-time classification for skin cancer diagnosis is that it can potentially assist dermatologists in identifying different types of pigmented skin lesions (PSLs) quickly and accurately	One limitation of using pixel-wise analysis for skin cancer diagnosis is that it may require a large and diverse dataset to accurately train the model. Another limitation is that this approach analyzes each pixel separately, which may not capture the overall pattern and structure of the lesion	Improved accuracy, potential for earlier diagnosis
Deep convolutional Generative Adversarial Networks to enhance Artificial Intelligence for skin cancer applications	Limited HS datasets available for skin cancer analysis	Convolutional DCGAN architecture to generate HS medical data	Federated learning can provide researchers with access to a large and diverse dataset while maintaining patient privacy through anonymous data. This can accelerate the development and application of deep learning methodologies in general	The technology has the potential to improve clinical practice by accelerating deep learning methodologies and increasing access to anonymous data, but large-scale data acquisition campaigns are needed to include diverse skin lesion types and clinical	GAN architecture for generating hyperspectral synthetic data with low sample size, evaluated by FID metric and validated using resnet-18 trained on synthetic data to classify real images

			clinical practice	centers	
Neural Networks-Based On-Site Dermatologic Diagnosis through Hyperspectral Epidermal Images	End-to-end dermatologic diagnosis using HSIs	AI system to assist dermatologists in clustering epidermal tumors and improve classification taxonomy	Improves dermatologist's classification performance in specificity, sensitivity, and accuracy; achieved real-time classification on a low-power Nvidia Jetson GPU device using a semantic segmentation network for a portable instrument containing an HS camera	The main limitation is dataset size, leading to other limitations, including unique skin signatures for each patient and inter-patient variability in these signatures	The study concentrates on developing deep learning algorithms for small datasets to improve dermatologist diagnostic performance. The future research should emphasize exploiting the vast amount of information in a single spectral cube for better classification and segmentation performance
Attention-based skin cancer classification through hyperspectral imaging	The study proposes an end-to-end dermatologic diagnosis using HSIs and suggests exploiting the vast amount of information in a single spectral cube for better classification and segmentation performance	A novel attention-based network that utilizes data augmentations to classify skin cancer through HS images	The proposed solution shows a lower false-negative rate than the state-of-the-art solutions and significantly reduces classification times compared to the best solutions in the literature	The proposed network was only tested on a dataset of 76 images, which may not be representative of all cases. Future research is required to enhance the network and assess different layer configurations	The proposed network is an interesting and promising solution for medical HS images classification, especially due to the lower false-negative rate and lower classification times. The utilization of data augmentatio

					ns and a fixed number of layers also provide more stable results compared to previous works
AI and HPC literature review concerning intraoperative brain tumour resection	Intraoperative glioblastoma segmentation of hyperspectral images: Accurate segmentation of glioblastoma during brain surgery is challenging due to its infiltrative nature and morphological similarity with surrounding healthy tissues	ML pipelines mainly from the European HELICoiD project, including unsupervised algorithms	The study introduces ML pipelines for intraoperative glioblastoma segmentation using hyperspectral images, allowing real-time processing during open craniotomy with competitive processing times	It is challenging to obtain the target ground truth for supervision.	None
AI-based segmentation of intraoperative glioblastoma hyperspectral images	Semantic segmentation of intraoperative glioblastoma hyperspectral images in real-time	The study proposes three DL architectures for real-time processing of hyperspectral imaging to differentiate between tumor and healthy tissue based on spectral information. CNNs are utilized to classify	The proposed method achieves competitive inference times and satisfies real-time constraints, providing end-to-end pixel-wise classification. It achieves 85% specificity and 70% accuracy for	The method's performance is poor in classifying healthy and background image portions as tumors. It relies on hyperspectral imaging, which is not widely available in clinical settings, and its performance	Improved gold-standard ML pipeline form European HELICoiD project for intraoperative glioblastoma segmentation using hyperspectral images

		hyperspectral images into tumor and non-tumor regions	the tumor tissue class	may be affected by the quality of the hyperspectral data	
Attention-based self-supervised U-net++ for the segmentation of intraoperative glioblastoma hyperspectral images	End-to-end semantic segmentation of intraoperative glioblastoma hyperspectral images in real-time	Innovative SSL algorithm to train a novel segmentation architecture	The proposed method achieves competitive pixel-wise classification results and competitive processing times	The method's performance is poor in classifying healthy and background image portions as tumors, as mentioned previously	The study proposes a SSL methodology that could improve medical hyperspectral image segmentation, and it is the first work on SSL applied to medical hyperspectral images

References

1. Barragán-Montero, A. et al. Artificial intelligence and machine learning for medical imaging: A technology review. Physica Medica 83, 242–256 (2021).
2. Goodfellow, I. et al. A. Deep Learning. (MIT Press, 2016).
3. Yu, K.-H. H. et al. Artificial intelligence in healthcare. Nature Biomedical Engineering 2018 2:10 2, 719–731 (2018).
4. Castiglioni, I. et al. AI applications to medical images: From machine learning to deep learning. Physica Medica 83, 9–24 (2021).
5. Rong, G. et al. Artificial Intelligence in Healthcare: Review and Prediction Case Studies. Engineering 6, 291–301 (2020).
6. Rajpurkar, P. et al. AI in health and medicine. Nature Medicine 2022 28:1 28, 31–38 (2022).
7. Morley, J. et al. The ethics of AI in health care: A mapping review, Social Science & Medicine 260, 113172 (2020).
8. Saraswat, D. et al. Explainable AI for Healthcare 5.0: Opportunities and Challenges. IEEE Access 10, 84486-84517 (2022).
9. de Hond, A.A.H. et al. Guidelines and quality criteria for artificial intelligence-based prediction models in healthcare: a scoping review. npj Digit. Med. 5, 2 (2022).
10. Sun, L., et al. A. Review and potential for artificial intelligence in healthcare. Int J Syst Assur Eng Manag 13 (Suppl 1), 54–62 (2022).
11. Florimbi, G. High performance modelling and computing in complex medical conditions: realistic cerebellum simulation and real-time brain cancer detection. (Università degli Studi di Pavia, 2018).
12. Florimbi, G. et al. Towards Real-Time Computing of Intraoperative Hyperspectral Imaging for Brain Cancer Detection Using Multi-GPU Platforms. IEEE Access 8, 8485–8501 (2020).
13. Biaoyang, L. et al. Digital Transformation in Personalized Medicine with Artificial Intelligence and the Internet of Medical Things. OMICS: A Journal of Integrative Biology 26 (2022).
14. Harmon, S. A. et al. Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets. Nat Commun 11, 1–7 (2020).
15. La Salvia, M. et al. Deep learning and lung ultrasound for Covid-19 pneumonia detection and severity classification. Comput Biol Med 136, (2021).
16. Secco, G. et al. Can alveolar-arterial difference and lung ultrasound help the clinical decision making in patients with covid-19? Diagnostics 11, (2021).
17. La Salvia, M. et al. Machine-Learning-Based COVID-19 and Dyspnoea Prediction Systems for the Emergency Department. Applied Sciences 12, 10869 12 (2022).
18. La Salvia, M. et al. Neural Networks-Based On-Site Dermatologic Diagnosis through Hyperspectral Epidermal Images. Sensors 22, (2022).

19. Torti, E. et al. Parallel Classification Pipelines for Skin Cancer Detection Exploiting Hyperspectral Imaging on Hybrid Systems. Electronics 9, 1503 9 (2020).
20. Marini, T. J. et al. Lung ultrasound: The essentials. Radiol Cardiothorac. Imaging 3, (2021).
21. Aiosa, G. et al. Role of lung ultrasound in identifying COVID-19 pneumonia in patients with negative swab during the outbreak. Emergency Care Journal 16, (2020).
22. Buda, N. et al. Lung ultrasound in the diagnosis of COVID-19 infection - A case series and review of the literature. Advances in Medical Sciences 65, 378–385 (2020).
23. Secco, G. et al. Lung ultrasound in COVID-19: a useful diagnostic tool. Emergency Care Journal 16, (2020).
24. Li, C. et al. Classification of severe and critical COVID-19 using deep learning and radiomics. IEEE J Biomed Health Inform, 3585–3594 (2020).
25. Buonsenso, D. et al. COVID-19 outbreak: less stethoscope, more ultrasound. The Lancet Respiratory Medicine 8, e27 (2020).
26. Shi, H. et al. Radiological findings from 81 patients with COVID-19 pneumonia in Wuhan, China: a descriptive study. Lancet Infect Dis 20, 425–434 (2020).
27. Chen, M. et al. Clinical applications of detecting IgG, IgM or IgA antibody for the diagnosis of COVID-19: A meta-analysis and systematic review. International Journal of Infectious Diseases 104, 415–422 (2021).
28. Ai, T. et al. Correlation of Chest CT and RT-PCR Testing in Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases. Radiology, 200642 (2020).
29. Peng, Q. Y. et al. Findings of lung ultrasonography of novel corona virus pneumonia during the 2019 -- 2020 epidemic. Intensive Care Medicine 46, 6–7 (2020).
30. Niederman, M. S. et al. Guidelines for the Management of Adults with Community-acquired Pneumonia. Am J Respir Crit Care Med 163, 1730–1754 (2001).
31. Mongodi, S. et al. Modified Lung Ultrasound Score for Assessing and Monitoring Pulmonary Aeration. Ultraschall in der Medizin 38, 530–537 (2017).
32. Khan, U., Paheding, S., Elkin, C. P. & Devabhaktuni, V. K. Trends in Deep Learning for Medical Hyperspectral Image Analysis. IEEE Access 9, 79534–79548 (2021).
33. Kumar, D. et al. Hyperspectral Image Classification Using Deep Learning Models: A Review. J Phys Conf Ser 1950, 012087 (2021).
34. Lu, G. et al. Medical hyperspectral imaging: a review. Journal of Biomedical Optics 19, 010901 (2014).
35. Barberio, M. et al. Intraoperative Guidance Using Hyperspectral Imaging: A Review for Surgeons. Diagnostics 11, 2066 11 (2021).
36. Reshef, E. R. et al. Hyperspectral Imaging of the Retina: A Review. Int Ophthalmol Clin 60, 85–96 (2020).
37. Johansen, T. H. et al. Recent advances in hyperspectral imaging for melanoma detection. Wiley Interdiscip Rev Comput Stat 12, e1465 (2020).

38. Leon, R. et al. Non-Invasive Skin Cancer Diagnosis Using Hyperspectral Imaging for In-Situ Clinical Support. Journal of Clinical Medicine 9, 1662 9 (2020).
39. Fabelo, H. et al. HELICoiD project: a new use of hyperspectral imaging for brain cancer detection in real-time during neurosurgical operations. Scientific Sensing and Imaging 9860, 986002 (2016).
40. Li, Q. et al. Review of spectral imaging technology in biomedical engineering: achievements and challenges. Journal of Biomedical Optics 18, 100901 (2013).
41. Nadeem, M. W. et al. Brain Tumor Analysis Empowered with Deep Learning: A Review, Taxonomy, and Future Challenges. Brain Sciences 10, 118 (2020).
42. Siegel, R. L. et al. A. Cancer statistics, 2022. CA Cancer J Clin 72, 7–33 (2022).
43. Ferlay, J. et al. Cancer statistics for the year 2020: An overview. Int J Cancer 149, 778–789 (2021).
44. Yousef, H. et al. Skin (Integument), Epidermis (StatPearls, 2021).
45. Dildar, M. et al. Skin Cancer Detection: A Review Using Deep Learning Techniques. International Journal of Environmental Research and Public Health 18, 5479 (2021).
46. Esteva, A. et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature 542, 115–118 (2017).
47. Rey-Barroso, L. et al. Optical Technologies for the Improvement of Skin Cancer Diagnosis: A Review. Sensors 2021, Vol. 21, Page 252 21, 252 (2021).
48. Ozdemir, A. et al. Deep Learning Applications for Hyperspectral Imaging: A Systematic Review. Journal of the Institute of Electronics and Computer 2, 39–56 (2020).
49. Signoroni, A. et al. Deep Learning Meets Hyperspectral Image Analysis: A Multidisciplinary Review. Journal of Imaging 5, 52 (2019).
50. Miller, K. D. et al. Brain and other central nervous system tumor statistics, 2021. CA Cancer J Clin 71, 381–406 (2021).
51. Fabelo, H. et al. Spatio-spectral classification of hyperspectral images for brain cancer detection during surgical operations. PLoS One 13, e0193721 (2018).
52. Fabelo, H. et al. In-Vivo Hyperspectral Human Brain Image Database for Brain Cancer Detection. IEEE Access 7, 39098–39116 (2019).
53. Pipitone, G. et al. Alveolar–Arterial Gradient Is an Early Marker to Predict Severe Pneumonia in COVID-19 Patients. Infect. Dis. Rep. 14, 470-478 (2022).
54. Chen, R. J. et al. Synthetic data in machine learning for medicine and healthcare. Nature Biomedical Engineering 5:6, 493–497 (2021).
55. Creswell, A. et al. Generative Adversarial Networks: An Overview. IEEE Signal Process Mag 35, 53–65 (2018).
56. La Salvia, M. et al. Deep Convolutional Generative Adversarial Networks to Enhance Artificial Intelligence in Healthcare: A Skin Cancer Application. Sensors 22, (2022).
57. Hastie, T. et al. The Elements of Statistical Learning. (Springer, 2009).

58. Deng, J. et al. ImageNet: A large-scale hierarchical image database. IEEE, 248–255 (2010).
59. Yosinski, J. et al. How transferable are features in deep neural networks? Advances in Neural Information Processing Systems 4, 3320–3328 (2014).
60. Shin, H. C. et al. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. IEEE Trans Med Imaging 35, 1285–1298 (2016).
61. Minaee, S. et al. Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning. Medical Image Analysis 65, 101794 (2020).
62. Rawat, W. et al. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. Neural Comput 29, 2352–2449 (2017).
63. Srivastava, N. et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research 15, 1929–1958 (2014).
64. Hoeser, T. et al. Object Detection and Image Segmentation with Deep Learning on Earth Observation Data: A Review-Part I: Evolution and Recent Trends. Remote Sensing 12, 1667 (2020).
65. Shorten, C. et al. A survey on Image Data Augmentation for Deep Learning. J Big Data 6, 1–48 (2019).
66. Tan, C. et al. A Survey on Deep Transfer Learning. Lecture Notes in Computer Science. Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics 11141 LNCS, 270–279 (2018).
67. He, K. et al. Deep residual learning for image recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 770–778 (2016).
68. Weng, W. et al. U-Net: Convolutional Networks for Biomedical Image Segmentation. IEEE Access 9, 16591–16603 (2015).
69. Zhou, Z. et al. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. Lecture Notes in Computer Science 11045 LNCS, 3–11 (2018).
70. Chen, L. C. et al. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Trans Pattern Anal Mach Intell 40, 834–848 (2016).
71. Chen, L. C. et al. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. IEEE Transactions on Pattern Recognition, 1–12 (2014).
72. Chen, L. C. et al. Rethinking Atrous Convolution for Semantic Image Segmentation. (Preprint, 2017).
73. Chen, L. C. et al. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. Lecture Notes in Computer Science 11211 LNCS, 833–851 (2018).
74. Goodfellow, I. J. et al. Generative Adversarial Networks. Sci Robot 3, 2672–2680 (2014).
75. Salimans, T. et al. Improved Techniques for Training GANs. Adv Neural Inf Process Syst 2234–2242 (2016).

76. Karnewar, A. et al.MSG-GAN: Multi-Scale Gradient GAN for Stable Image Synthesis. (2019).
77. Wang, T. C. et al. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 8798–8807 (2018).
78. Nvidia Corporation. CUDA C++ Programming Guide. (NVIDIA Corporation - 2788 San Tomas Expressway, Santa Clara, CA 95051, 2022).
79. Nvidia Corporation. Nvidia cuDNN Documentation. (NVIDIA Corporation - 2788 San Tomas Expressway, Santa Clara, CA 95051, 2022).
80. Mellemgaard, K. et al. The Alveolar-Arterial Oxygen Difference: Its Size and Components in Normal Man. Acta Physiol Scand 67, 10–20 (1966).
81. Alballa, N. et al. Machine learning approaches in COVID-19 diagnosis, mortality, and severity risk prediction: A review. Inform Med Unlocked 24, 100564 (2021).
82. Musulin, J. et al. Application of Artificial Intelligence-Based Regression Methods in the Problem of COVID-19 Spread Prediction: A Systematic Review. International Journal of Environmental Research and Public Health 8, 4287 (2021).
83. Rehman, A. et al. COVID-19 Detection Empowered with Machine Learning and Deep Learning Techniques: A Systematic Review. Applied Sciences 11, 3414 (2021).
84. Roy, S. et al. Deep learning for classification and localization of COVID-19 markers in point-of-care lung ultrasound. IEEE Trans Med Imaging, 1–1 (2020).
85. Muhammad, G. et al. COVID-19 and Non-COVID-19 Classification using Multi-layers Fusion From Lung Ultrasound Images. Information Fusion 72, 80–88 (2021).
86. Barros, B. et al. Pulmonary COVID-19: Learning Spatiotemporal Features Combining CNN and LSTM Networks for Lung Ultrasound Video Classification. Sensors 21, 5486 (2021).
87. Brinati, D. et al. Detection of COVID-19 infection from routine blood exams with machine learning: a feasibility study. J Med Syst 44, (2020).
88. Goodman-Meza, D. et al. A machine learning algorithm to increase COVID-19 inpatient diagnostic capacity. PLoS One 15, 1–10 (2020).
89. Cabitza, F. et al. Development, evaluation, and validation of machine learning models for COVID-19 detection based on routine blood tests. Clin Chem Lab Med 59, 421–431 (2021).
90. Plante, T. B. et al. Development and external validation of a machine learning tool to rule out COVID-19 among adults in the emergency department using routine blood tests : a large, multicenter, real-world study. Med Internet Res 22, 1–12 (2020).
91. Soldati, G. et al. Proposal for International Standardization of the Use of Lung Ultrasound for Patients With COVID-19: A Simple, Quantitative, Reproducible Method. Journal of Ultrasound in Medicine (2020).
92. Horry, M. J. et al. COVID-19 detection through transfer learning using multimodal imaging data. IEEE Access 8, 14982–49808 (2020).
93. Kingma, D. et al. ADAM: a method for stochastic optimization. (Preprint, 2015).

94. Monshi, M. M. A. et al. CovidXrayNet: Optimizing data augmentation and CNN hyperparameters for improved COVID-19 detection from CXR. Comput Biol Med 133, 104375 (2021).
95. Selvaraju, R. R. et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. Proceedings of the IEEE International Conference on Computer Vision 618–626 (2017).
96. Mento, F. et al. Deep learning applied to lung ultrasound videos for scoring COVID-19 patients: A multicenter study. J Acoust Soc Am 149, 3626 (2021).
97. Morales, A. et al. Laboratory Hyperspectral Image Acquisition System Setup and Validation. Sensors 22, 2159 (2022).
98. Haggenmüller, S. et al. Skin cancer classification via convolutional neural networks: systematic review of studies involving human experts. Eur J Cancer 156, 202–216 (2021).
99. Fabelo, H. et al. Dermatologic Hyperspectral Imaging System for Skin Cancer Diagnosis Assistance. 2019 34th Conference on Design of Circuits and Integrated Systems, 1–6 (2019).
100. Dosovitskiy, A. et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. (Preprint, 2020).
101. Zhu, L. et al. Generative Adversarial Networks for Hyperspectral Image Classification. IEEE Transactions on Geoscience and Remote Sensing 56, 5046–5063 (2018).
102. Obukhov, A. et al. Quality Assessment Method for GAN Based on Modified Metrics Inception Score and Fréchet Inception Distance. Advances in Intelligent Systems and Computing 1294, 102–114 (2020).
103. Zhong, Z. et al. Generative Adversarial Networks and Conditional Random Fields for Hyperspectral Image Classification. IEEE Trans Cybern 1–12 (2019).
104. Yi, X. et al. Generative adversarial network in medical imaging: A review. Medical Image Analysis 58, 101552 (2019).
105. Tschandl, P. et al. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Nature 5, 1–9 (2018).
106. Mirza, M. et al. Conditional Generative Adversarial Nets. (Preprint, 2014).
107. Annala, L. et al. Generating Hyperspectral Skin Cancer Imagery using Generative Adversarial Neural Network. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1600–1603 (2020).
108. Fujisawa, Y. et al. Deep-learning-based, computer-aided classifier developed with a small dataset of clinical images surpasses board-certified dermatologists in skin tumour diagnosis. British Journal of Dermatology 180, 373–381 (2019).
109. Goyal, M. et al. Artificial intelligence-based image classification methods for diagnosis of skin cancer: Challenges and opportunities. Computers in Biology and Medicine 127, 104065 (2020).
110. Hamilton, M. et al. Unsupervised Semantic Segmentation by Distilling Feature Correspondences. (Preprint, 2022).