Universidade do Minho
Escola de Engenharia

Bárbara Malainho Pereira

Automatic Interpretation of Point-of-care Lung Ultrasound

outubro de 2022

Universidade do Minho
Escola de Engenharia

Bárbara Malainho Pereira

Automatic Interpretation of Point-of-care Lung Ultrasound

Dissertação de Mestrado
Mestrado Integrado em Engenharia Biomédica

Trabalho efetuado sob a orientação de:
Professor Doutor Jaime Francisco Cruz Fonseca
Doutor Sandro Filipe Monteiro Queirós

outubro de 2022

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

AGRADECIMENTOS / ACKNOWLEDGEMENTS

No culminar desta fase da minha vida, quero deixar o meu sincero agradecimento a todas as pessoas que contribuíram para esta dissertação.

Ao meu orientador, Professor Doutor Jaime Fonseca quero agradecer por toda a disponibilidade e apoio prestados. Ao meu coorientador, Doutor Sandro Queirós, quero agradecer pela aprendizagem, pela ajuda e pelo tempo despendido em torno desta dissertação, sem o qual a mesma não seria possível. Quero também deixar um agradecimento a todos os médicos envolvidos no projeto, à Dra.Ana Oliveira, à Dra.Marcela Karnikowski, ao Dr.Marco Carvalho-Filho e ao Dr.Mateus Lech, que foram fundamentais para este trabalho.

A todos os meus colegas e amigos, pelo incentivo e confiança constantes que depositam em mim. Um especial obrigada a quem me acompanhou nestes anos de mestrado, por tornarem agradáveis as infindáveis horas de trabalho, pela paciênca e pelo sentimento de pânico coletivo a que sempre me acostumaram.

Aos meus pais, que não só me apoiaram como sentiram comigo cada segundo desta etapa, que me depositam uma confiança cega e me fazem reacreditar sempre em mim, um obrigada nunca será suficiente.

E por fim, ao meu Miguel. Pelo amor, pelo tempo, pela calma, por me relativizar a vida e por acreditar, incondicionalmente, em mim. Obrigada por absolutamente tudo.

"Though this be madness, yet there is method in't"
William Shakespeare

The work present in this project was performed in the Life and Health Sciences Research Institute (ICVS, School of Medicine) and in Centro ALGORITMI (School of Engineering), University of Minho. Financial support was provided by National funds, through the Foundation for Science and Technology (FCT) - project PTDC/EMD-EMD/1140/2020 and scholarship UMINHO/BIM/2021/68. It is also acknowledged the donation of a RTX A6000 GPU by NVIDIA (USA).

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

RESUMO

Interpretação Automática de Ultrassom Pulmonar *Point-of-Care*

A ultrassonografia pulmonar *point-of-care* (POCUS) é uma modalidade de imagem médica segura, portátil e de baixo custo, útil em contextos de urgência para uma rápida examinação à cabeceira do paciente. Apesar de limitado pela presença de ar, o ultrassom pulmonar (LUS) tem demonstrado ser um recurso prestável na avaliação de doenças crónicas ou agudas, dependendo da interpretação de fenómenos de ultrassom, assim como de artefactos associados aos mesmos. Atualmente, com a situação pandémica criada pela COVID-19, a necessidade de uma interpretação de imagens médicas e correspondente diagnóstico célere tornou-se mais clara. Coincidentemente, a presença de soluções baseadas em métodos de *deep learning* (DL) na área da imagem médica têm aumentado, com resultados bastante promissores. Não obstante o seu potencial, a utilização destas metodologias em POCUS pulmonar permanece subexplorado.

Este trabalho propõe uma *framework* de DL para a interpretação de vídeos de POCUS pulmonar, cujos resultados são os achados presentes num dado vídeo (tais como linhas A, linhas B, consolidações, entre outros). A metodologia, baseda numa arquitetura 3D, inicializa-se com uma rotina de pré-processamento para a padronização dos dados. Sendo que os dados são escassos mas essenciais para treinar um modelo de DL eficaz, duas abordagens são exploradas : aprendizagem supervisionada e semi-supervisionada. O trabalho culmina com a proposta de uma estatégia inovadora de *ensemble* de modelos, que combina os resultados de modelos treinados para prever anotações distintas, bem como uma rotina de pós-processamento opcional e específica para o *dataset*, ambas com o intuito de potencializar a hierarquia inerente à interpretação de LUS.

A *framework* proposta e os seus módulos constituintes foram avaliados através de um conjunto extenso de testes, considerando tanto modelos multi-classe como *multi-label*, para contextos supervisionados e semi-supervisionados. Os resultados comprovam a versatilidade da

framework, possibilitando uma combinação customizável dos diferentes elementos da proposta de acordo com a tarefa pretendida. Num conjunto independente de dados de teste, a proposta categórica, útil para uma triagem célere, obteve uma média de valores de F1 de 92.61%, enquanto a proposta *multi-label*, vantajosa para o acompanhamento e encaminhamento do paciente, obteve uma média de valores de F1 de 70.45% quando considerados cinco achados de LUS relevantes.

De modo global, a proposta demonstra-se promissora numa área subexplorada, abrindo caminho para uma interpretação automática e precisa de ultrassom pulmonar em contexto clínico.

Palavras chave: Aprendizagem semi-supervisionada, Análise de vídeo, *Deep learning*, Ultrasonografia pulmonar

ABSTRACT

Automatic Interpretation of Point-of-Care Lung Ultrasound

Point-of-care ultrasound (POCUS) is a safe, portable and low-cost imaging technique useful in emergency rooms for a rapid bedside patient examination. Although being limited by the presence of air, lung ultrasonography (LUS) has proven to be a helpful resource in the evaluation of acute and chronic conditions, relying on the interpretation of ultrasound phenomena and associated image artifacts. Currently, with the COVID-19 pandemic, the necessity for an expeditious image interpretation and associated diagnosis has become clearer than ever. Coincidentally, deep learning-based solutions have increased their presence in the medical imaging field, with alluring results. Notwithstanding their potential, the usage of these techniques in lung POCUS remains underexplored.

This work proposes a flexible deep learning (DL) framework for the interpretation of lung POCUS videos, whose outputs are the finding(s) present in said video (such as A-lines, B-lines, consolidations, among others). The pipeline, based on a 3D architecture, is initialised with a pre-processing routine for video standardisation. Since data is scarce but a core necessity to train a successful DL model, two learning strategies are investigated: supervised and semi-supervised scenarios. The work culminates with the proposal of a novel model ensembling strategy, which aggregates the outputs of models trained to predict distinct label sets, and an optional dataset-specific post-processing routine, both aimed at leveraging of the hierarchy inherent to LUS interpretation.

The proposed framework and its building blocks were evaluated in an extensive set of experiments, considering both multi-class and multi-label models, for both supervised and semi-supervised settings. The results show the framework's versatility, allowing for a custom combination of the multiple proposed blocks accordingly to the task in question. In a held-out test set, the categorical proposal, which is useful for an expedite triage, achieved an average F1-score of 92.61%, while the multi-label proposal, helpful for patient management and referral, achieved an average F1-score of 70.45% when considering five relevant LUS findings.

Overall, the proposal shows promise in an underexplored field, paving the way for an accurate computer-assisted lung ultrasound interpretation in clinical practice.

Keywords: Automatic video analysis, Deep learning, Lung ultrasonography, Semi-supervised learning

TABLE OF CONTENTS

LIST OF ACRONYMS

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
ANN	Artificial neural network
AP	Average precision
BA	Balanced accuracy
BLUE	Bedside Lung Ultrasound in Emergency
BN	Batch normalisation
CCE	Categorical cross entropy
CNN	Convolutional neural network
COVID-19	Coronavirus disease 2019
CT	Computer tomography
DL	Deep learning
DNN	Deep neural network
ECE	Expected calibration error
FN	False negative
FOV	Field-of-view
FP	False positive
JSON	JavaScript Object Notation
LL - AI	Left lung anterior inferior
LL - AS	Left lung anterior superior

LL - DI	Left lung diaphragm insertion
LL - LI	Left lung lateral inferior
LL - LS	Left lung lateral superior
LL - PI	Left lung posterior inferior
LL - PS	LLeft lung posterior superior
LL - SL	LLeft lung sliding
LSR	Label smoothing regularisation
LSTM	Long short-term memory
LUS	Lung ultrasound
MCC	Matthews correlation coefficient
ML	Machine learning
MRI	Magnetic ressonance imaging
POCUS	Point-of-care ultrasound
ReLU	Rectified linear unit
RGB	Red-Green-Blue
RL - AI	Right lung anterior inferior
RL - AS	Right lung anterior superior
RL - DI	Right lung diaphragm insertion
RL - LI	Right lung lateral inferior
RL - LS	Right lung lateral superior
RL - PI	Right lung posterior inferior
RL - PS	Right lung posterior superior
RL - SL	Right lung sliding
SSL	Semi-supervised learning

TN	True negative
TP	True positive
UPS	Uncertainty-aware Pseudo-labelling Selection
US	Ultrasound

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

1.1 Lungs' topography and mechanics

The lungs are the core organs of the respiratory system and, in tandem with the respiratory tract, contribute to homeostasis by being responsible for the exchange of gases between the atmospheric air and the bloodstream, and from the latter to the tissue cells.

The basic structures of the respiratory system can be divided according to their location or functionality. Concerning the former, the upper respiratory system contains the nasal cavity, pharynx, and associated structures, while the lower respiratory system refers to the larynx, trachea, bronchial tree, pulmonary alveoli, and lungs. Similarly, the system is divided into two parts based on function: the conducting zone and the respiratory zone. The first consists of an ensemble of interconnected cavities and tubes responsible for transporting gases to and from the pulmonary alveoli. The second comprehends the tissues within the lungs responsible for gas exchange, such as the bronchioles, alveolar ducts, alveolar sacs, and alveoli.

To perform the task of transfusion, the respiratory system must fulfil several requirements regarding the corporeal aspect and the composition of its various components. When the air enters the nasal or oral cavity, it is filtered, warmed, and humidifies on its way to the lungs. This purification is the result of the passage of air in the coarse hairs in the nostrils, the cilia, and the mucus, while the warming is due to the heat emanating from the blood vessels near the surface of the lining of the airways. For these transformations to take place, the fluid exchange must occur deep within the body, hence the conductive zone. Moreover, to facilitate the process, the membrane of the alveoli, where most of the diffusion occurs, must be thin-walled and selectively permeable, while moisture is maintained by the extensive capillary structure to provide the ideal environment for the dissolution of oxygen and carbon dioxide. Air must be constantly renewed, so the system must incorporate an effective ventilation mechanism [1–3].

The lungs are paired, large, spongy, and conical organs enclosed in the thoracic cavity (Figure 1). The heart and mediastinum form a division between these structures. Each lung lies against the ribs anteriorly and posteriorly, just above the clavicle, and extends to the diaphragm, the muscle that separates the thoracic cavity from the abdominal one. Each lung precisely conforms to the contour

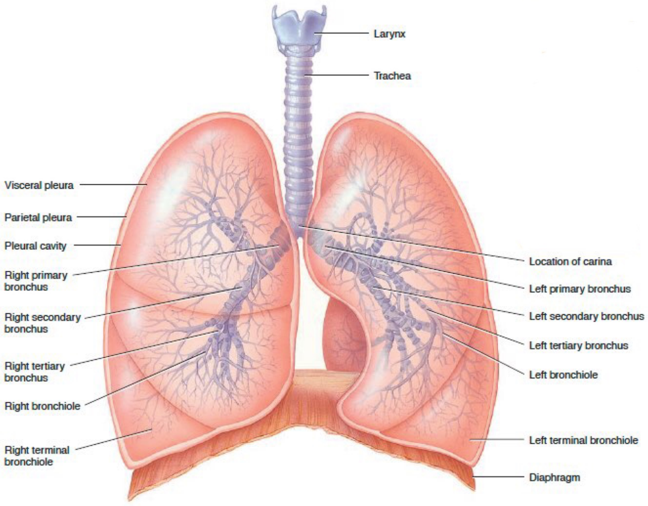


Figure 1: Lungs and the branching of airways from the trachea.
Adapted from [2].

of the thoracic cavity. The broad underside of the lung, called the base, has a concave shape and fits the convex dome of the diaphragm. In contrast, the apex is the narrow upper part of the lung, also named cupola, which extends above the clavicle. The costal surface of the lung which abuts the membranes covering the ribs conforms to the rounded curvature of these bones. In its turn, the mediastinal surface of the lung is also slightly concave and contains the hilum, a depression through which pulmonary blood and lymphatic vessels, nerves and bronchi pass [2, 3].

The left and right lungs, although largely mirror images, have one significant difference: the left lung is 10% smaller than the right due to an indentation called the cardiac notch, or cardiac impression, the site in the medial surface that accommodates the heart. In like manner, the right lung is not only wider and thicker but also shorter since the diaphragm is slightly higher on that side, to make space for the liver that lies inferiorly.

The lungs comprise a series of lobes and lobules separated by fissures (Figure 2). The right lung has three lobes while the left lung has only two lobes. An oblique fissure is present in both lungs, vertically symmetrical, and divides each lung into two sections, with the right lung having an additional horizontal fissure. In the left lung, the oblique fissure forms the superior and inferior lobes. Regarding the right lung, the upper part of the horizontal fissure gives rise to the superior lobe, while the oblique fissure separates the inferior and middle lobes. Each lobe comprehends its own secondary lobar bronchus, i.e. the right primary bronchus produces three secondary bronchi (superior, middle, and inferior), while the left primary bronchus originates the superior and inferior secondary lobar bronchi. The secondary bronchi themselves give rise to the tertiary bronchi, each lung having ten of these structures. These structures supply a segment of tissue called the bronchopulmonary segment. Within, there are a series of small lobules enveloped in elastic connective tissue and containing a lymphatic vessel, an arteriole, a venule, and a branch of a terminal bronchiole. The latter diverges

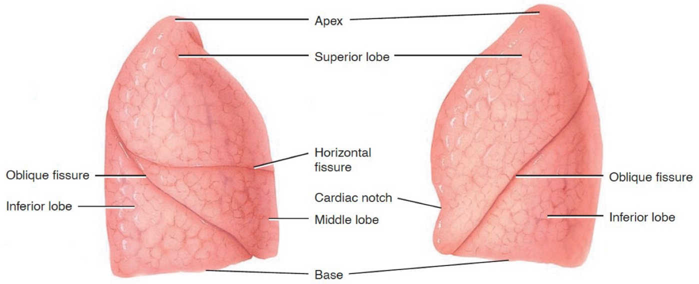


Figure 2: Lateral view of the right and left lungs.
Adapted from [2].

into microscopic branches, the respiratory bronchioles, which then culminate in the alveolar ducts that lead to the pulmonary alveoli [2, 3].

The lungs are surrounded by a double-layered serous membrane, the pleurae, that lines the thoracic cavity and protects its organs. The visceral pleura adheres to the lungs themselves and covers each of the interlobar fissures. The parietal pleura lines the wall of the thoracic cavity and the thoracic surface of the diaphragm. The pleural cavity, i.e. the small space between the pleurae, contains a lubricating fluid that allows membranes to slide easily over one another during breathing and reduces friction. In addition, the fluid helps the two membranes adhere to each other, while also adhering to the thoracic wall. The pulmonary ligaments help support the lungs, extending downward from the pleural layers. These two structures are responsible for the constant contact of the lungs with the thoracic wall, which causes the lung to change size as it expands and contracts according to the movement of the rib cage during breathing [2, 3].

Basic respiration is an involuntary mechanism of the human body that can be summarised in three fundamental steps: pulmonary ventilation, i.e. the breathing *per se*; external respiration, i.e. the transfer of gases from the alveoli into the bloodstream; and internal respiration, i.e. the exchange of gases between blood and tissue cells. In this project, one will focus primarily on the first step, which involves movement, hence its detailed description below.

Pulmonary ventilation is the passage of air from the atmosphere to the lungs, and comprises two steps: inhalation and exhalation, also termed inspiration and expiration (Figure 3). The airflow is a pressure-induced process, i.e. it occurs due to the coordinated changes in three different pressures: atmospheric pressure, intraalveolar or intrapulmonary pressure, and intrapleural pressure [1, 3].

Inhalation, the process of breathing in, is the active phase of ventilation, during which the thoracic cavity enlarges by contraction of the diaphragm and the external intercostal muscles, lowering the pressure inside the alveoli. Contraction of the diaphragm causes it to flatten, lowering its dome and increasing the diameter of the thoracic cavity vertically as the rib cage moves upward and outward. As the lung volume increases, the intrapleural pressure decreases and a partial vacuum is created. At this point, the alveolar pressure is lower than the atmospheric pressure. As air flows from an area of high pressure into a space of lower pressure, the air enters the lungs by extending a continuous

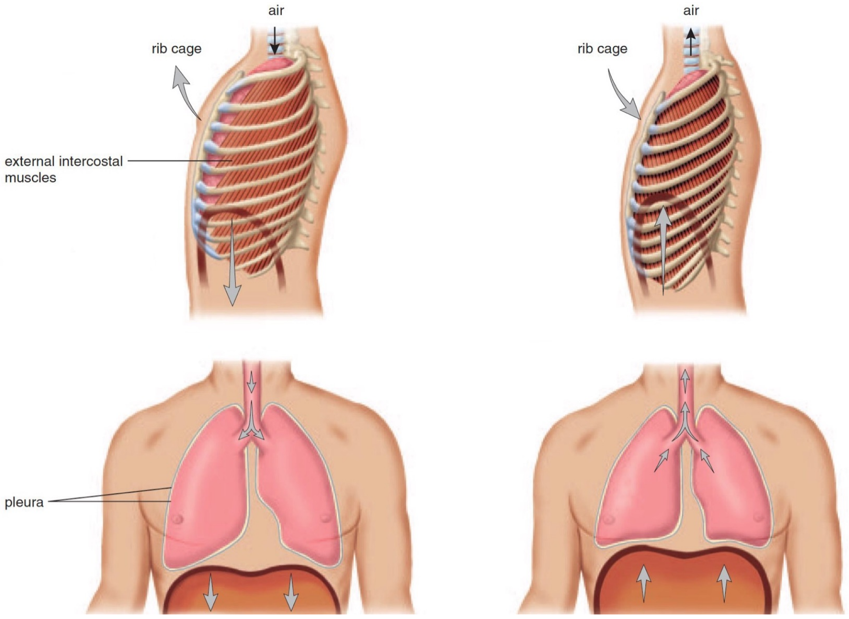


Figure 3: Inspiration vs. expiration.
Adapted from [1].

column of air from the pharynx to the alveoli. Note that, while the inhalation process is the active phase of breathing, the flow into the alveoli is passive; air is not forced into the lungs, it enters the lungs because they are expanded [1, 3].

Exhalation is largely a passive process, that occurs when the muscles, once contracted during inhalation, relax. Unlike inhalation, exhalation results from the elastic recoil of the chest wall and lungs, causing the diaphragm to return to its dome shape. All these phenomena combine to decrease the vertical, lateral, and anteroposterior diameters of the thorax, decrease lung volume, and consecutively increase alveolar pressure, making the air naturally flow out of the body. A lipoprotein substance called surfactant produced in some alveolar cells is responsible for the non-collapse of the alveoli by lowering the surface tension within the alveoli (by reducing the attractive forces of hydrogen bonds) and, in combination with the drop in pressure in the pleural cavity, ensures that the alveoli remain open [1–3].

The respiratory system is particularly susceptible to disease, as the lungs are the internal organ most vulnerable to infection and injury. Because the respiratory tract is warm and humid, it provides an ideal environment for pathogens to thrive, and many pathogens are airborne. Since humans are constantly exposed to particles, chemicals and infectious organisms in ambient air, respiratory diseases are an immense health burden worldwide. They are responsible for more than 10 % of all disability-adjusted life years and account for five of the thirty most common causes of death [3, 4].

1.2 Lung pathologies and their diagnosis

Lung pathologies have multiple causes, and their diagnosis and follow-up often require various imaging technologies and investigative procedures. Pulmonary diseases are defined as conditions preventing the proper functioning of the lungs and affecting either respiration or lung function, i.e. the ability to breathe and how properly the lungs work [5].

Every diagnosis must start with the fundamental approach, a physical examination. A physical examination of the respiratory system follows four consecutive steps: inspection, palpation, percussion, and auscultation. Inspection may reveal physical signs, like for instance abnormal breathing patterns, chest wall deformities, oedema, and cyanosis. Palpation may be an advantage in discovering enlarged lymph nodes and subcutaneous emphysema. Conditions such as pleural effusion and pneumothorax can be unveiled by percussion, since they may reveal areas of dullness or hyper resonance, respectively. Lastly, auscultation identifies characteristic sounds of some respiratory diseases, including wheezes, crackles, or a pleural effusion rub [6].

Laboratory methods are also helpful tools for lung assessment. Besides the conventional urine and blood tests, some specific blood and sputum analyses are used to detect diseases. Microbiological tests, for example, have an essential role in the examination of infectious respiratory diseases caused by viruses, bacteria, fungi, or parasites, using bacteriological, molecular, biological or serological techniques. Pathologies such as pulmonary embolism, cystic fibrosis, coronavirus disease 2019 (COVID-19), pneumonia, lung cancer, sarcoidosis, and asthma have specific markers that can be found when resorting to these methods. Moreover, histology and cytology play a major role in the diagnosis of infection, and malignant and benign diseases [6].

Besides being used for diagnosis, respiratory function tests have a big contribution in the assessment of severity, treatment monitoring and prognosis evaluation. These include spirometry, diffusion capacity, blood gas analysis, and cardiopulmonary exercise testing, among others [6, 7].

In addition, like in other fields of medicine, the diagnosis of respiratory pathologies rely heavily on medical imaging, such as X-rays, computer tomography (CT), magnetic ressonance imaging (MRI), nuclear imaging, bronchoscopy, and ultrasound (US). These modalities represent a significant tool for physicians to better understand the cause of the problem and have profoundly changed the practice of medicine and investigation, as they are able to provide better diagnosis and treatment [5].

Starting with chest radiography, the first step in the radiological evaluation of suspected respiratory distress patients; to computed tomography, which offers a much more detailed observation of the thoracic structures; pulmonary and bronchial angiography, for vessels imaging; and magnetic resonance imaging, when there are suspicions of a tumour in the mediastinum or chest wall, these techniques are some of the most used methods for lung assessment.

Ultimately, a well-established but underutilised imaging modality for lung assessment is the transthoracic ultrasound. Safe, low cost, mobile, fast, and allowing for a real-time evaluation, this technique allows the visualisation of the pleura and chest wall abnormalities, diagnosis of pneumothorax and works as an aid for minimally invasive procedures such as biopsies. Although not being able to identify chronic obstructive pulmonary disease, asthma or pulmonary embolism (as these are associated with images too similar to the ones of a healthy lung), ultrasound usage is increasing due to its unique ability to quickly diagnose some of the most common lung diseases. For example, if a patient presents a pneumothorax, the air is accumulated in the pleural cavity, which will appear in the ultrasound acquisition differently from a normal lung once the medium alters and thus the propagation of the sound wave. Atelectasis, resulting from a pleural effusion, a bronchial obstruction, or an alveolar-interstitial syndrome, all of which lead to the aggregation of extravascular water, diffuse alveolar oedema, and thickened interlobular septa, can also be identified in the ultrasound images. The same can be said about pneumonia, which is characterised by the exudes invading the pleural cavity because of an inflammation of the lung tissue. More recently, studies have shown that COVID-19 may also be detectable through ultrasound imaging [6, 8, 9]. For the above reasons, ultrasound has proven its usefulness and importance in modern medicine, especially in emergency scenarios, where it is termed point-of-care ultrasound (POCUS).

1.3 POCUS

Point-of-care ultrasound concerns a portable ultrasound system that allows a bedside patient assessment. Conceived for the purpose of being utilised in emergency settings, it differs from the generic ultrasound in its intended application. The latter is focused on precise images, which take time. On the other hand, POCUS is used for a quick evaluation to reach a reliable but not time-consuming diagnosis, due to the urgent context. Globally, in addition to reducing the diagnosis time, improving patient safety (as it is a non-ionizing technology) and decreasing complication rates, POCUS provides essential health care independently of the patient's location.

1.3.1 *Ultrasound fundamentals*

Acute care practitioners who perform point-of-care ultrasounds need to comprehend the basic concepts of how ultrasound equipment works to be able to make critical decisions [10]. This knowledge is fundamental to establish a solid foundation for the understanding of the nature and behaviour of ultrasound and to identify artifacts. This insight enables physicians to produce high-quality images and assures their interpretation of the same, translating into the ability to deliver improved patient care [11].

Ultrasound consists of waves. Ultrasonic waves are longitudinal progressive compression waves, meaning that the particles in the medium dislocate in the direction of the wave propagation. This generates areas of high and low particle density, as illustrated in Figure 4, corresponding to sections of

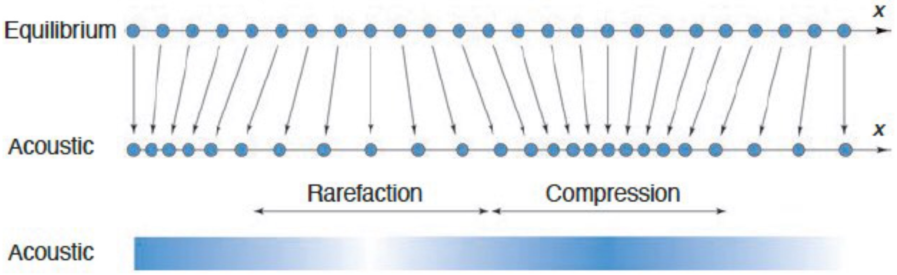


Figure 4: Schematic representation of a longitudinal wave.
Adapted from [12].

compression and rarefaction, respectively. This motion is achievable due to some inherent properties of the medium, such as elasticity and inertia. The elastic characteristic has its interaction as a neutraliser of the compression site, to restore equilibrium. Due to inertia, this equilibrium will be too large and, consequently, will create a region of rarefaction that will be counterbalanced by elasticity, forming a loop of action. After some cycles, the equilibrium is achieved, since in each iteration the propagation is damped as the wave propagates in space. The nomination of "ultrasonic" is referring to the frequency of the wave and refers to wave frequencies above the audible spectrum of the human ear (20 Hz – 20 kHz). In medical US imaging specifically, a frequency 100 to 1000 times higher than the maximum audible frequency is used [12].

The shape's wave is irrelevant to the propagation, meaning that any waveform generated can propagate through matter. Assuming a homogeneous medium and a linear wave, there are some key properties that are useful to characterise the wave. Wavelength is the distance between two similar points on a wave, over one cycle. The period is the time it takes to complete said cycle. However, when discussing ultrasound waves, the frequency, the inverse of the period, is more often used, since it defines the number of cycles repeated per second, being measured in Hertz (Hz). The strength of the wave is described by three parameters: its amplitude, i.e. the size of the wave that, in a graphic representation, is measured between the average and the maximum value; its power, which measures the energy per unit of time and is expressed in Watts; and its intensity, a measure of the concentration of energy in a cross-section of an ultrasound beam. In its turn, the acoustic impedance is defined by the ratio of the acoustic pressure by the frequency-dependent resistance that an ultrasound wave comes upon as it passes through a medium [12, 13].

The ultrasound, as mentioned, is composed of waves that propagate and interact at the interface where different tissues meet, depending on a medium of propagation, such as fluids, soft tissues, and solids. These interactions can be reflections, diffractions, refractions, attenuations, and scattering, for example. By measuring these time-dependent reflections and knowing the wave velocity that is defined by the product of the frequency and the wavelength, we can infer the position of the tissues [10, 13].

These waves can be generated and detected by a piezoelectric crystal, which is a phenomenon known as the piezoelectric effect. These crystals induce an electric field in response to a mechanical force, i.e. a deformation, and vice-versa. Therefore, a compression wave is formed every time an alternating voltage is applied to the crystal which is responsible for converting the electric energy to mechanical energy, with the wave having the same frequency as the signal used.

An ultrasound transducer, also named probe, is constituted by various piezoelectric elements that work sequentially, either to emit the pulse or receive the signal and can be composed of materials such as quartz, lithium niobate and tourmaline. The multiple elements present in the transducer allow the image to be obtained by shifting these elements in a systematic manner. Considering a transducer with 9 elements, the first line scanned can be generated using elements 1-4, the second from 2-5 and so on, scanning 6 lines in total. The number of lines can be altered by changing the number of elements activated per line scanned [10, 13].

Resolution is defined as the minimum distance between two structures that allows for differentiability as separate structures. In ultrasound imaging, the spatial resolution is distinguished into three resolutions: axial, lateral and elevation. Axial resolution (also known as longitudinal) refers to the resolution in the direction of the wave and can be improved by raising the transmitted frequency, knowing that with this the attenuation increases and therefore the penetration depth decreases [12, 13]. Mathematically, it is the same value as half of the spatial pulse length [14]. Lateral resolution is perpendicular to the axial direction of the acquired image, which can be improved by increasing the bandwidth and the central frequency of the pulse, translating to a reduction in the beam width [12, 13]. The width varies with the distance from the transducer, beginning approximately equal to the transducer's width, narrowing with distance until the smallest width possible (near-zone) and, after a certain point, returning to a wider beam (far zone) (Figure 5). With the balancing of these distances, the ultrasound develops the possibility of focusing. By diminishing the near zone to the focal length and considering that the beam diverges from that point onward, the focal region is created. Notwithstanding, the lateral resolution can also be improved by focusing on multiple depths within the same tissue by the repetition of pulses being emitted along the scan line [14]. Resolution in the perpendicular direction to the image plane is titled elevation. This resolution is dependent and, in some parts, controlled by the ultrasound transducer [12, 13]. One should note that being

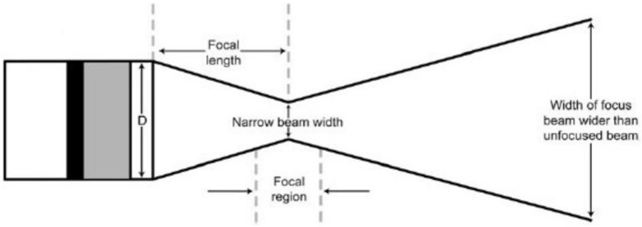


Figure 5: Focal region representation.
Adapted from [14].

ultrasound a real-time imaging method, the anatomical structures are displayed as sequential frames over time, meaning that temporal resolution is also an ultrasound imaging property. Representing the ability of the system to identify and distinguish between instantaneous events of moving structures, the temporal resolution is the time passed between frames. It can be increased by reducing the depth of penetration, reducing the number of focal points, or reducing the number of scan lines per frame [14].

Multiple types of transducers exist, being the main difference between them the shape obtained by the various piezoelectric elements placed together. Depending on the shape, the transducer will be more adapted to certain parts of the human body and therefore used for different purposes. For abdominal imaging, curvilinear transducers are the most used due to their penetration capability, allowing the observation of deeper structures and generating a pie-shaped field of view. For small and difficult access regions, the phased array is the ideal choice. It is mostly used in cardiac imaging, resulting in a pie-shaped image as well. Linear array transducers are best for analysing muscles, nerves, soft tissue, and vasculature, displaying a rectangular image. Lastly, endocavitary transducers are used inside a body cavity and the resulting images have a wide angle (up to 180 degrees).[13] With the choice of the transducer, the frequency range is chosen inherently, although the expert can select a frequency between certain permitted limits [13].

Two other features of ultrasound, besides frequency, that must be taken into consideration for the correct acquisition of the images are gain and depth. Gain, also termed brightness when talking about the resulting images, should be the minimal required to highlight the targeted structures without saturating the image. Similarly, depth must also be kept as low as needed to display the target anatomy only. When decreasing depth, the structures will appear amplified, and the resolution is increased. In opposition, scanning at higher depths will decrease frame rate and image resolution. As gain decreases with depth, ultrasound imaging employs time-gain compensation. This is called the attenuation phenomenon and is responsible for the different values of the amplitude of identical structures at different depths. The time-gain formula intends to counter this natural attenuation, keeping the gain uniform along the field-of-view (FOV) [10, 12, 13].

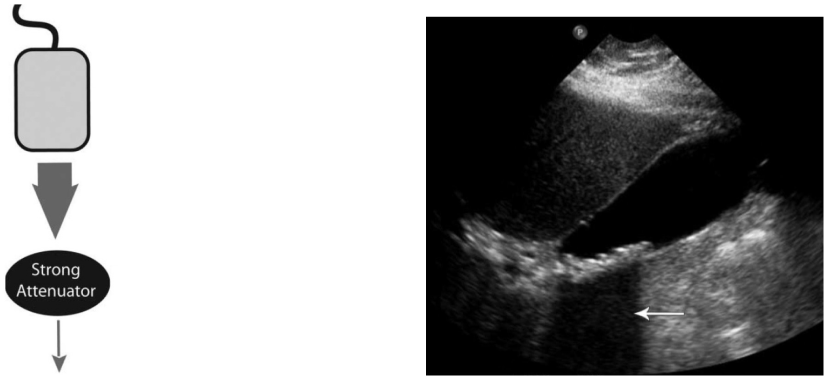
To obtain spatial information from the ultrasounds, instead of using a continuous electrical signal to power the transducer, pulses are applied to discover information about the human body, which can be done in three different manners. A-mode, also known as amplitude imaging, is based on the pulse-echo principle. A line is scanned through the target, and it utilises the transducer as a receiver immediately after the pulses were emitted, and the reflected and scattered waves that are identified by the crystal are converted and displayed as a time-dependent function. Note that the concept of time in ultrasonography is equivalent to depth since the velocity of sound is fairly constant within the tissue. M-mode, motion imaging, consists of the A-mode measure repeated, capturing the movement of objects over time, focusing the ultrasound beam on a stationary location. The third way to present the acquired information is the B-mode, brightness imaging. This two-dimensional (2D) image is possible with the translation of the transducer between two A-mode acquisitions. If this measure is repeated in time, an image sequence (or ultrasound video) is obtained. The name brightness imaging comes from the physical representation of a brightness point in the image since amplitudes from the returning ultrasound are proportionally displayed as different brightness values. In clinical ultrasound, although A-mode and M-mode are utilised, B-mode is the most widely employed, particularly in lung ultrasound (LUS) [12, 13].

Ultrasound image generation is based on several assumptions. It is assumed that the only source of sound waves is the transducer, that the sound wave travels linearly and at a constant velocity, that the tissues in the body are uniform and that each reflector will produce one single echo. When this is not verified, the phenomena that occur are named artifacts. Artifacts are any divergence from these suppositions. Their occurrence can provide useful information for diagnostics and their presence may be confirmed by shifting the transducer, as body structures will remain visible, but the artifacts will alter.

1.3.2 *Ultrasound artifacts*

Ultrasound artifacts can be an asset in the interpretation of US images. Structures that reflect strongly, such as calcifications, produce brighter reflections and are titled echogenic. Contrarily, hypogenic areas of the image are characteristic of weak reflection structures, like blood. However, in ultrasound imaging, to correctly perceive anatomy and findings, one must be aware of differences in speckle pattern or texture.

Some artifacts common to ultrasonography, in general, are shadowing, acoustic enhancement, mirror images, reverberation, ring down, and comet tail. Shadowing (Figure 6) is a highly common artifact that exists when the emitted sound waves are mostly reflected or absorbed at the interface



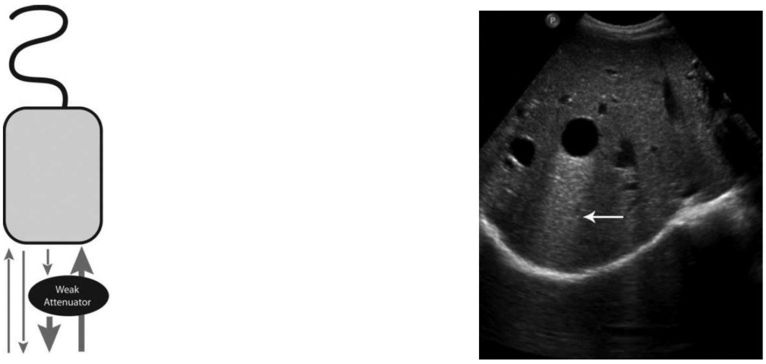
(a) (b)

Figure 6: Shadowing artifact.
(a) Illustration of an US beam encountering a strong attenuator. (b) Example US image showing a shadowing artifact (arrow). Adapted from [15].

and has two main causes. The first cause is the presence of air, as when the incident pulse faces an air-tissue interface most of the sound wave is reflected with only a small portion being transmitted through the interface to the proceeding air, with whom it will interact and create secondary reflections that will travel back to the transducer. It also occurs when facing osseous structures and calcifications, with strong attenuators. Since these elements are porous, there is an increase in the absorption of the sound wave. The first mentioned reason creates a "dirty shadowing" appearance, while the second translates into a "clean shadowing" [12, 13, 15].

Posterior acoustic enhancement (Figure 7) is a helpful artifact in the identification of structures containing fluid or other weak attenuators. Since sound waves are less attenuated by fluids than solids, when the incident pulse is passing through fluid-filled structures, like cysts, the result will be stronger reflections and a brighter appearance [12, 13, 15].

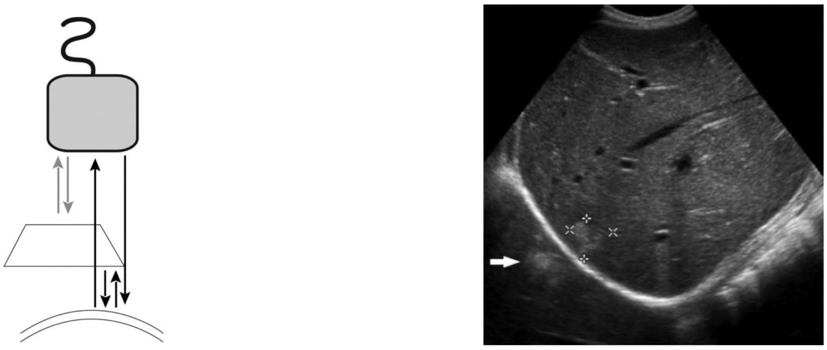
Mirror imaging (Figure 8) is another artifact caused by a highly reflective surface that is smooth, like an interface between air and soft tissue. This may happen, for example, in the base of the right



(a)

(b)

Figure 7: Posterior acoustic enhancement artifact.
(a) Illustration of an US beam encountering a weak attenuator. (b) Example US image showing a posterior acoustic enhancement artifact (arrow). Adapted from [15].



(a)

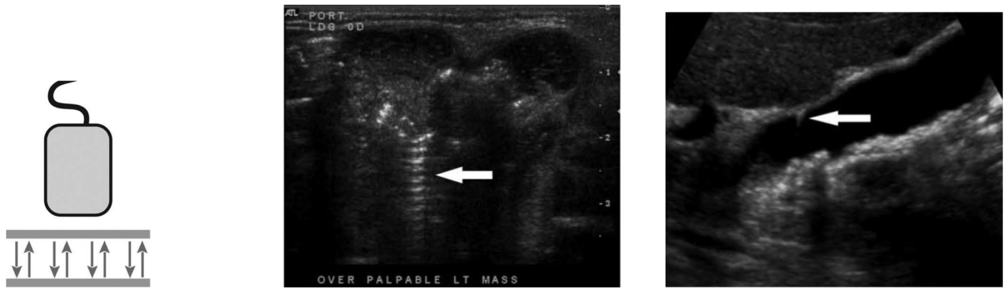
(b)

Figure 8: Mirror image artifact.
(a) Illustration of an US beam encountering a reflective interface. (b) Example US image showing a mirror artifact (arrow). Adapted from [15].

lung, since it is an air-filled structure. The base can serve as a mirror on abdominal ultrasonography, creating a duplicated image of the liver or diaphragm [12, 13, 15].

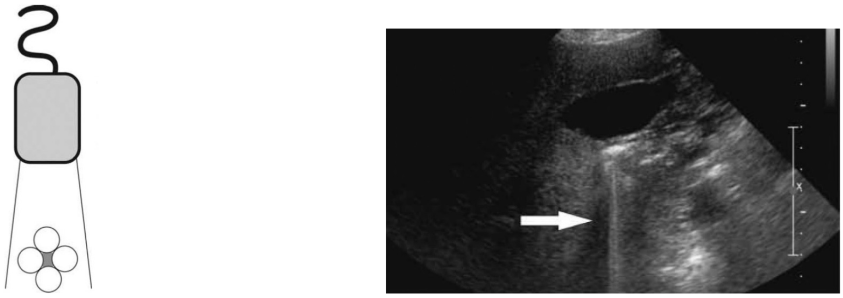
Reverberation (Figure 9b) is an artifact that happens when there is a reflective surface in the near field, such as cystic structures, from which a reflective sound wave is created. This wave is strong enough to be reflected in the transducer and again into the body, interacting with the surface multiple times and generating an effect called ping-pong. When the two interfaces that create the reflective surface are very closely spaced, the sequential echos may be so near that it becomes impossible to discern the individual signals. This will result in a display of hyperechoic reflections that have a narrow base and form a ray that goes from the structures to the bottom of the screen, with a triangular shape, named comet tail artifact (Figure 9c) [12, 13, 15].

In the presence of air bubbles with fluid encapsulated between them, the excitation of the fluid can cause the fluid to resonate, creating a continuous wave to be generated and transmitted back to the transducer with the original echo, forming a set of bright echoes deep to the air, called ring down artifact (Figure 10). It is common when analysing an air-filled bowel or when there is metal in the region of the scanned image [12, 13, 15].



(a) (b) (c)

Figure 9: Reverberation artifact.
(a) Illustration of an US echos being repeatedly reflected.(b) Example US image showing a reverberation artifact (arrow). (c) Example US image showing a comet tail artifact (arrow). Adapted from [15].



(a) (b)

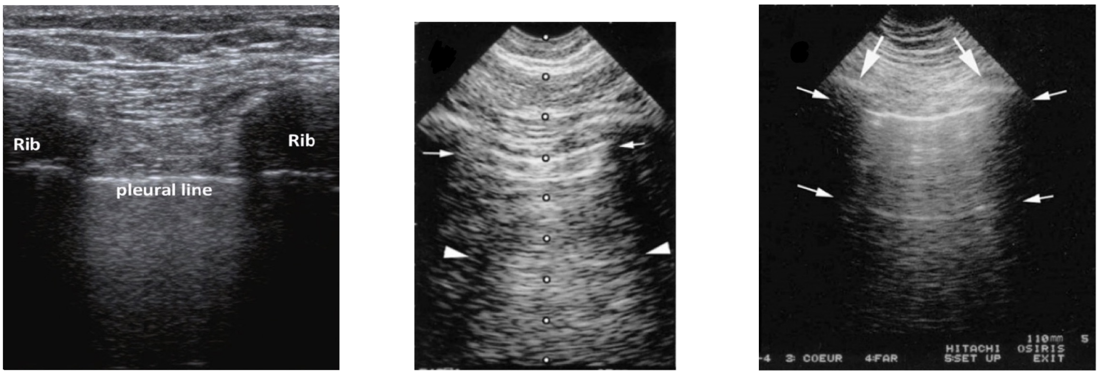
Figure 10: Ring down artifact.
(a) Illustration of an US beam encountering a ring of bubbles with fluid trapped in the middle. (b) Example US image showing a ring-down artifact (arrow). Adapted from [15].

These fundamental concepts are essential for a correct and accurate interpretation of pulmonary ultrasound images. Even though lungs are filled with air, a fact usually considered prohibitive for US imaging, with the interpretation of these artifacts, physicians, paradoxically, transform the acoustic limitations of this modality into a diagnostic advantage [8].

1.3.3 *LUS image interpretation*

As mentioned in previous sections, ultrasound waves need a medium to propagate, so one could naively believe that their use is very limited in lung imaging. Any process within the lung that is surrounded by an aerated lung will not be shown on the ultrasound images since the transmission of the ultrasound is blocked due to a high acoustic mismatch with the surrounding tissue. Although having these limitations, lung ultrasound relies in great part on the analysis of artifacts. Indeed, an intensivist specialised in ultrasound imaging can interpret several findings and pathologies by relying on ultrasound artifacts, some of which are characteristic of a normal healthy lung, while others are associated with certain pathologies. Ultimately, LUS has proven its value in the evaluation of different acute and chronic conditions, being a portable, safe, and quick-performing technology.

An ultrasound of a normal lung presents only one key structure, the pleura. A pleural line is a horizontal hyperechoic line that appears below the rib line. Doubts still exist concerning if the visualised line is the pleura itself or a reflection artifact at the interface between alveolar air and the soft tissues of the thoracic wall. Still, the artifacts surrounding the pleural line provide important information. The pleura moves in sync with respiration, and this dynamic movement observed in normal lungs is called lung sliding, representing the free movement of the visceral pleura against the parietal pleura. To correctly locate the pleura, professionals rely on another artifact, the bat sign (Figure 11a). When the US probe is positioned longitudinally, the upper and lower ribs will form the wings of a bat and, a little deeper, the pleural line will form the belly of the bat. When encountering



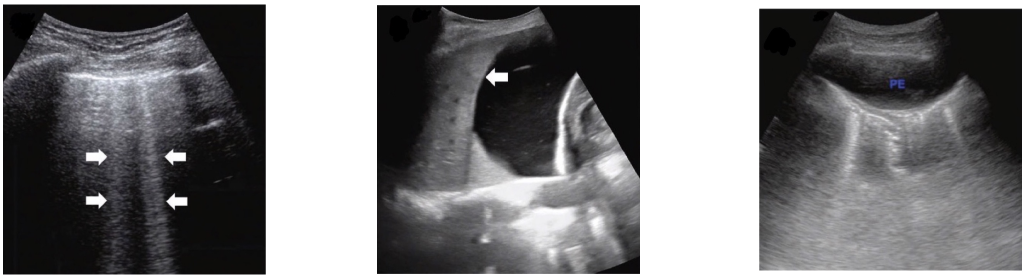
(a) (b) (c)

Figure 11: Example US images with artifacts distinctive of a normal lung.
(a) Bat sign. (b) A-lines. (c) The presence of the bat sign and A-lines simultaneously. Adapted from [13, 16].

the pleural membrane (an interface of different acoustic impedance), the ultrasound wave will be reflected by the pleura and bounce back and forward with the transducer, resulting in hyperechoic horizontal lines arising at regular intervals (whose space is the same as the distance between the probe and pleural line) and visible deeper than the pleural lines. These artifacts are known as A-lines (Figure 11b), and although their presence is not necessarily indicative of a normal lung, their absence is often a sign of existing pathology [8, 9, 13].

An artifact that works as a pathological indicator is the B-line (Figure 12a). B-lines are hyperechoic lines, deriving from the comet tail artifact and arising vertically to the pleural line. They result from a reverberation artifact due to the juxtaposition of thickened interlobular alveolar septa and air within the alveoli and move coherently with lung sliding. The number of B-lines detected is directly related to the disease severity. If the number of B-lines is higher than or equal to three, we are in the presence of a positive B-line pattern and, in the reverse situation, of a negative B-line pattern. Pleural effusion is another disease marker that presents itself as a dependent dark zone in the pleural cavity. Besides being a pathology itself, the manifestation of this artifact can uncover other underlying diseases (Figure 12b). The quad sign is an artifact that can indicate pleural effusion, confirming the presence of free fluid. It consists of the pleural line (upper limit), the lung line representing the visceral pleural (lower limit), and the two rib shadows (right and left limits) (Figure 12c). One last artifact is the lung pulse. Resulting of the propagation of the heartbeats to the adjacent lung, the lung pulse will not be detected when the pleural cavity is filled with air or the lung is overinflated. Like lung sliding, it cannot be observed in a still image [9, 13].

As mentioned, LUS interpretation is based on artifacts. For example, a normal healthy lung is characterised by lung sliding, presence of A-lines or scattering, and absence of any pathological finding. Although commonly not present, isolated B-lines can be found in some lung regions in healthy lungs. Lung sliding along with A-lines alone are characteristic of a “dry” lung, excluding the possibilities of pulmonary oedema, pneumonia, and pneumothorax. However, an A-profile can also appear in some diseases like asthma and pulmonary embolism. Since a pneumothorax is an air accumulation in the pleural cavity, the presence of B-lines rules out this pathology completely. In



(a) (b) (c)

Figure 12: Example US images with artifacts distinctive of lung pathologies.
(a) B-lines. (b) Pleural effusion. (c) Quad sign. Adapted from [13].

addition, the absence of lung sliding and lung pulse are also identifiers of the disease. Atelectasis, on the other hand, is diagnosed by the absence of lung sliding and the existence of a lung pulse, while alveolar-interstitial syndrome is a fluid condition, and consequently manifests as multiple B-lines. However, relying solely on ultrasound examination is not sufficient to discriminate between atelectasis and consolidation since both pathologies have similar artifact presentation. Pneumonia is another fluid disorder, recognised by the existence of consolidation (which is not considered an artifact but a finding on LUS; Figure 13), the presence of pleural effusion and the omission of lung sliding [8, 9, 13].

As implied above, there is a natural hierarchy in lung pathology diagnosed by LUS. More precisely, if any of the artifacts considered pathological are present, the presence of A-lines is irrelevant to the diagnosis. Similarly, while a negative B-line pattern and pleural irregularity are artifacts indicative of a pulmonary alteration, without any additional pathological artifact, these are not enough to diagnose pathology. This hierarchy can also be transposed to the pathologies within themselves. For example, since atelectasis and consolidation are within the most severe pathologies, they superimpose the remaining pathologies.

LUS is an imaging technique of value for bedside evaluation in critical care. Being a cost and time-efficient technique to diagnose multiple pulmonary pathologies, lung examination by ultrasound can be part of a protocol of action, hence the point-of-care lung ultrasound. To facilitate the examination process, a set of protocols exist that can be followed and a number of pulmonary views should be taken in order to perform a correct evaluation.

1.3.4 *LUS protocol and fields*

The use of POCUS as a pulmonary diagnostic tool has been increasing in recent years due to its availability and easy access. The ability to be performed in real-time at the patient's bedside is an advantage for critically care, emergency departments and trauma surgery. The POCUS approach can be used as an addition to standard testing, or even substitute for some, not so safe, standard

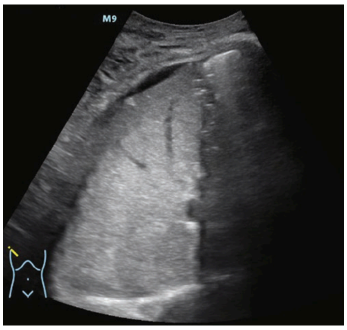


Figure 13: Example US image of a lung consolidation.
Adapted from [13].

tests. To aid the interpretation of the LUS exam, physicians have come up with a set of decision trees. These present the correct sequence and the combination of sonographic signs that either confirm the disease or discard it, with the Bedside Lung Ultrasound in Emergency (BLUE) protocol being an example (Figure 14). BLUE is a fast protocol, in which the examination should take under three minutes, and its purpose is to provide a flowchart for the diagnosis of acute respiratory failure [17].

To perform the pulmonary assessment, there are standard views, or ultrasound windows, normally used to evaluate lung and pleura. These act as a guideline and considering that the quality of the performed exam is highly user-dependent, the existence of a standard procedure normalises the examination. Since some artifacts may not be visualised in all parts of the lung and, if there is a pathology, it may not be identifiable across the entire lung, multiple windows (or fields) are usually acquired. Although variable across hospitals or clinical protocols, the lung ultrasound exam may be composed of eight, ten or even twelve fields. In the present project, the focus will be on the 12-field protocol (Figure 15) , given the additional views provided.

Roughly, for the purpose of the exam, one can observe each lung as divided into two sections: the superior and the inferior. Each zone is viewed from two different perspectives, anterior and posterior. The lateral portion of the lung is visualised with, also, two angles, one to focus on the parenchyma and another one on the diaphragm. In summary, for each lung, six fields are examined: anterior superior (AS), anterior inferior (AI), posterior superior (PS), posterior inferior (PI), lateral superior (LS) and lateral inferior (LI). Additionally, this protocol includes two extra views, one of the anterior superior, usually with a lower acquisition depth which aim to assess lung sliding (SL), and one of the lateral

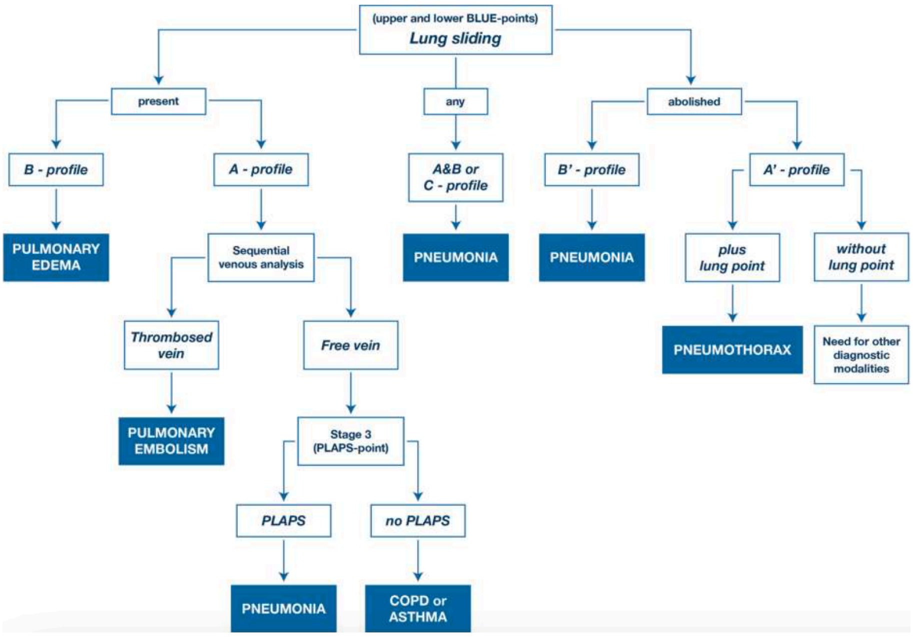


Figure 14: BLUE protocol flowchart.
Adapted from [18].

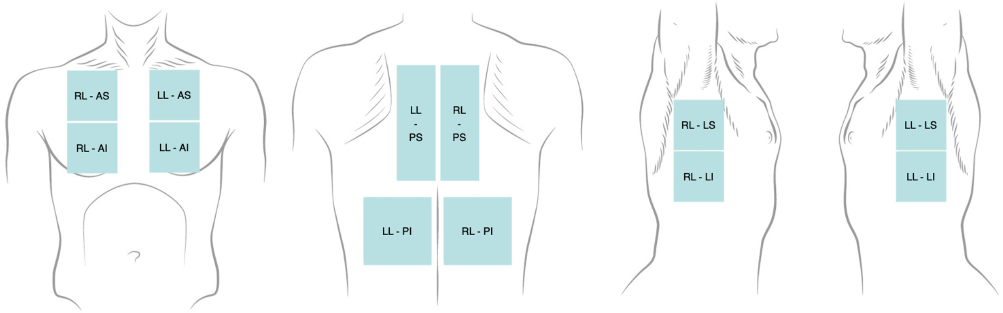


Figure 15: Twelve fields of LUS examination.
The prefix 'RL' is indicative of a right lung window, and the prefix 'LL' of a left lung window. Adapted from [19].

inferior with a different inclination, focusing on the diaphragm insertion (DI). Each of these fields is analysed for both lungs [8].

1.4 Motivation

The physics behind ultrasound generation and propagation into information ready to be used are complex. However, its application in clinical practice is much simpler and more direct. A basic understanding of how ultrasound works is a necessity for practitioners who use point-of-care ultrasound, especially in emergency scenarios, to make accurate and rapid decisions [10].

Lung ultrasound interpretation relies on common ultrasound phenomena since the medium is not favourable for wave propagation. Consequently, it is the clinician's responsibility to correctly interpret the images, as they portray a false anatomical representation. Hence the need for specialised training to perform a correct assessment with this imaging modality.

To date, some proposals have emerged to automatise ultrasound interpretation. Some authors focused their work on the identification of individual artifacts [20–23], while others concentrated their research on developing classifiers for a set of non-overlapping pathologies [20, 24]. However, most of the existing work does not contemplate multi-label problems, lacking applicability. Indeed, these fail to faithfully represent the reality of LUS, which may comprise multiple findings, all of which are relevant to reach a conclusion regarding the presence of a given pathology.

In clinical settings, the experts draw conclusions by observing the ultrasound throughout time. However, video-based approaches for LUS interpretation are still underexplored. Frame-based deep learning (DL) classifiers are a commonly used model to handle the data [20,22,25,26] and, in some cases, precede algorithms that aggregate the classifier's results to obtain a video-level conclusion [21, 24, 27]. However, only a few opt for networks that work directly with raw LUS data [23, 28]. Such information, lost in the previously mentioned implementations, is important since the inherent movements associated with certain findings are crucial for their appraisal.

Moreover, a LUS examination includes multiple videos, from different fields. Thus, an evaluation that aggregates the extracted information from all videos is favourable to obtain a global clinical outcome.

1.5 Aims and contributions

This dissertation focuses on the development of a computer-assisted interpretation and diagnosis algorithm for lung POCUS videos, to be an asset in clinical practice, specifically in emergency rooms or remote locations where specialised personnel is not present. To achieve this goal, the intent is to automatically interpret the artifacts-based findings in videos of LUS. The main contributions are thus as follows:

1. A survey of the state-of-the-art LUS interpretation algorithms and video handling techniques;
2. Development a framework that automates LUS video interpretation and diagnosis, which includes:
   a) Preparation of a large annotated dataset.
   b) Development of a custom pre-processing algorithm.
   c) Development of a flexible DL-based classification framework for automatic interpretation of videos and identification of relevant LUS findings, which can be used in a supervised or semi-supervised setting and works for multi-class (rough video classification) and multi-label (detection of findings) problems.
3. Validation of the developed algorithms.

1.6 Outline

The current document is composed of five chapters. In the current chapter, a clinical background was presented focused on the respiratory system, as well as lung pathologies and diagnostic techniques. The chapter culminates with the presentation of the project's motivation, goals and contributions. The following chapters further develop the introduced topics.

The second chapter starts by introducing some fundamental concepts of deep learning and relevant sub-classes, and afterwards presents the literature review on the state-of-the-art deep learning models for lung ultrasound interpretation.

The third chapter presents the developed methods of the proposal. It comprehends the data pre-processing algorithm, both supervised and semi-supervised training approaches, and the proposed dataset-specific model ensembling and post-processing routines.

The fourth chapter is dedicated to the validation of the proposed framework and each one of its core blocks. To do so, one considered both multi-class and multi-label scenarios with the latter investigated under both supervised and semi-supervised settings.

Finally, the fifth chapter concludes the work and presents the possible refinements and future directions of the proposal.

2

PULMONARY ULTRASOUND INTERPRETATION

This chapter focuses on relevant and fundamental concepts for the understanding of the project. Since the aim is to develop an automatic interpreter of LUS videos using DL, its basic concepts will be introduced and explained. An overview of the state-of-the-art techniques and models for image interpretation is then presented, with special attention to lung ultrasound and/or video-oriented interpretation, followed by a brief outline of some of the most common DL-associated problems and techniques on how to handle them. This chapter culminates with a brief compendium of semi-supervised learning techniques and their respective core ideas.

2.1 Deep learning

Artificial intelligence is based on the theory of computers being able to perform tasks that would normally require human intelligence. Machine learning (ML) is a sub-field of artificial intelligence that aims to give computers the “ability to learn without being explicitly programmed” for a certain task. ML makes it possible for the computer to build a mathematical model based solely on given data, recognise patterns within it, constantly learn and improve, and make predictions [29]. Among the possible tasks, ML is frequently used to develop regressors or classifiers, in which known data is fed to the system and the algorithm’s capacity is verified by testing it with unknown data. However, ML systems are often not capable of working with raw data and instead operate on hand-crafted (i.e. human-dependent) features [30, 31].

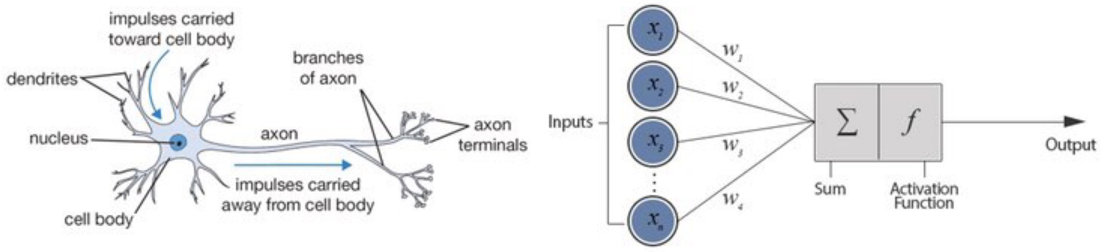
Deep learning (DL) is a type of ML with a structure similar to the human mind. It intends to, just as ML, analyse data and draw conclusions, but try to employ a logic comparable to the human one and, thus, much more complex than ML. Indeed, DL has birthed the development of deep neural networks, highly connected networks that pass information through nodes, simulating the distribution of information through neurons in biological systems. The distinctive detail of DL is that the programmers are not defining the features fed to the system,instead, the system learns these features directly from the raw data. The whole process is done end-to-end, from raw input to intended output. Consequently, DL has a much higher necessity for large amounts of training data [29, 30].

The best form to evaluate how well the model is able to generalise when presented with new data is to try it with new cases. To that end, the data must be split into three sets: training, validation and test. As implied, the model training will be performed in the training set and the test set is used to test the model, verifying its capability to generalise and, overall, how it responds to unseen data. The validation set is used to evaluate the candidate models and select the appropriate one. Multiple trainings are carried out with multiple hyper-parameters, and the model that presented the best results when applied on the validation set is selected [32].

Regarding the information provided with the data, there are two main groups of deep learning approaches, depending on whether the labels (the desired algorithm's output) are or are not at one´s disposal. **Supervised learning** implies a set of input-label pairs are provided so that the network can be directly trained to predict the predetermined output. In the polar opposite, there is **unsupervised learning**, when the input data is unlabelled and one relies on pattern recognition for clustering and associating the data within itself. **Self-supervised learning** is inserted in unsupervised techniques, in the sense that no target is defined *a priori*; however instead of finding patterns, it intends to solve tasks commonly associated with supervised learning, such as image classification. By creating an alternative task with labels produced by the machine itself, and then, after the underlying features of the dataset are learnt, leveraging of those learned parameters, the model is trained to predict the wanted output. In the middle of the spectrum is **semi-supervised learning**, in which part of the dataset is labelled but a larger portion is unlabelled [33, 34].

2.1.1 *Artificial neural network*

The human brain possesses characteristics that are impossible to reproduce. One of the most sought-after aspects of the brain is its ability to learn without any programming. Artificial neural networks (ANNs) were based on this biological behaviour and its organisation in neurons. Brain functioning is highly dependent on the signals received by synapses in each neuron, that controls them. Analogously to the brain, an ANN is formed by a network of artificial neurons in a computer. These neurons are connected by links that provide weights to each input, an adder sums these weights and feeds them to the activation function. The activation function is a monotone function that maps the output of the neuron. In analogous terms, a biological neuron is composed of dendrites, which correspond to the links and respective weights in an artificial neuron, a cell body, the biological equivalent to the weighted sum, and an axon, the activation function (Figure 16) [35, 36].



(a) (b)

Figure 16: Biological neuron vs. artificial neuron.
Adapted from [37].

Basic components

One of the first and simpler ANN is the perceptron. The perceptron has only a single layer and is a binary linear classifier. Introduced in 1957 by Frank Rosenblatt, the perceptron was, in the words of its creator, “the first machine capable of having an original idea” [38]. Notwithstanding its innovation factor, perceptrons are incapable of solving complex problems, which led to the development of the multi-layer perception, which consisted of the stacking of multiple perceptrons. In this network, the data is fed via the input layer, which passes the information to the subsequent layers. This information is processed by the weighted interconnections and then returned via the output layer. Between the input layer and the output layer, there are hidden layers, creating a multi-layer neural network, also called a feed-forward network. A network containing a large stack of hidden layers is called a deep neural network (DNN). When each neuron receives as information the outputs from all previous neurons, the layer takes the title of fully connected layer (or dense layer), as exemplified in Figure 17 [32].

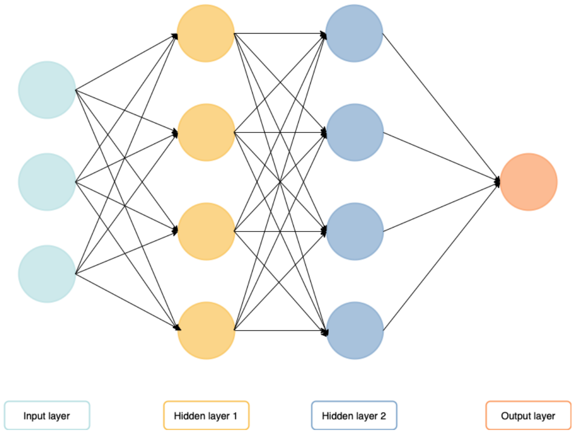


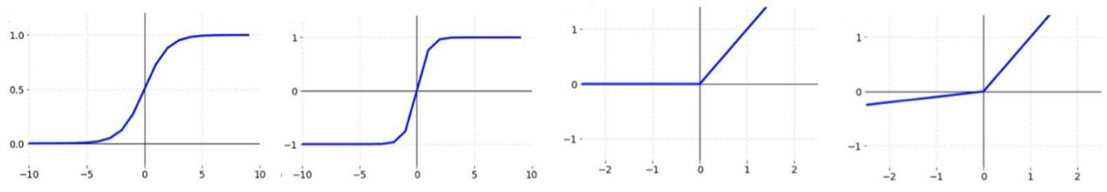
Figure 17: Basic representation of a multi-layer neural network.
Adapted from [39].

Although the premise of ANNs was promising, the research encountered a stalemate: there was no appropriate learning algorithm to train multi-layer perceptrons. Hence the emergence of the backpropagation training algorithm, which is still used nowadays. Training begins with the random initialisation of the hidden layer's connections weights, to allow backpropagation to train a diverse group of neurons. Each training instance (or sample) is fed to the network and passes through each layer, from the input to the output, obtaining the first prediction based on the seen data – the forward pass step. The output is then compared to the known result, that is, the label of the respective sample, and a measure of error is calculated (termed loss). Then, the algorithm goes through the architecture in reverse and the error from each connection in measured – reverse pass – and, ultimately, the connection weights are adjusted with the purpose of reducing the error – gradient descent step [32, 36].

To understand the algorithm on a deeper level, there are a few details that should be further explored, namely activation functions, losses, optimisers and regularisation. All of them influence the networks' ability to minimise the error, i.e. the network's expressivity.

The activation function plays a fundamental role in the learning ability. Without these functions, the neural networks would behave as linear regression models and would not be able to handle more complex data such as images, videos, or audio. Activation functions make it possible to recognise complex mappings in large non-linear datasets. There are two major groups of activation functions: the sigmoid functions, just like the logistic sigmoid function, hyperbolic tangent and arctangent; and the piece-wise linear functions, such as the rectified linear unit (ReLU) or LeakyReLU (a variant of ReLU). Some examples of these functions are present in Figure 18 [40].

Sigmoid functions are bounded functions that have only one inflexion point. Having the graphical representation of an "S" shape, these functions are mathematical approximations of step functions, being faithful to the correlation with the biological neuronal activation. Since they constrain the values to a certain interval (hence the name bounded or squashing functions), they are more susceptible to suffering from either vanishing or exploding gradient problems. Briefly, these issues are linked to gradient descent. The vanishing gradient problem, as the name implies, is associated with gradients diminishing as the algorithm progresses to deeper layers, making the weights unchanged and, as



(a) (b) (c) (d)

Figure 18: Graphical representation of several activation functions.
(a) Sigmoid function. (b) Hyperbolic tangent function (or tanh). (c) ReLU function. (d) Leaky ReLU function.
Adapted from [39].

a consequence, the training does not converge to an optimal result. The exploding question is the exact opposite: the gradients increase and the weights attributed to the connections get too large, diverging the algorithm. The piece-wise linear functions can be presented as a possible solution to this question. Although not a perfect approach as they converge to zero in the limit of infinitely small values functions like ReLU avoid the vanishing/exploding gradient in the limit of infinitely large values. To also tackle infinitely small values, the LeakyReLU appeared, avoiding the problem of dying ReLUs [32, 40, 41].

For the output layer of a network, the most common functions used are softmax and sigmoid. The softmax function (Equation 1) is commonly used in multi-class problems, normalising the input vector (z) into a probability distribution, by guaranteeing that the sum of the elements (i) of the output vector equals 1. The sigmoid (Equation 2) is applied in binary and multi-label problems and bounds the value of each class between 0 and 1. The output of the network will thus resemble the probability of each element belonging to each class. In the softmax case, the class with the highest value is assumed to be the prediction result of the network for said instance. However, for sigmoid, a threshold is defined, and every class whose value surpasses it belongs to the output labels [32].

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{1}$$

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2}$$

Loss functions are methods that evaluate how adequately a specific network models the input data. In other words, it evaluates the candidate model. The objective is to minimise the result of this function, which can be controlled by the change of the weights' values [32].

To minimise the loss function, one can implement optimisers, i.e. algorithms used to update the weights iteratively. There are three gradient descent computation variants, which differ only on the amount of data they require, namely: batch gradient descent, or vanilla gradient descent, which performs an average over the gradients of the entire dataset and uses that value to update the weights, which for large datasets becomes prohibitively costly in terms of time and memory, and thus rarely used in DNNs; the stochastic gradient descent, which updates the parameters for each training example; and, lastly, the mini-batch gradient descent approach that, as the name suggests, performs an update for every mini-batch (data subset used for one iteration). Irrespective of the amount of data used, optimisers often present a fixed learning rate (a parameter controlling how much the weights are adjusted in relation to the loss gradient), which can difficult the tuning of the network's training and often hampers its convergence. To tackle this and allow an adjustable convergence during training, optimisers like Adam [42], Adagrad [43] and Adadelta [44] have since appeared.

The ability to create a successful network for a specific task is achieved by fine-tuning these parameters. Notwithstanding, with the advancements in deep learning some architectures have been developed with a specific aim, and one particular type of architecture is notorious for its capability of handling images, the convolutional neural network (CNN).

Convolutional neural networks

CNNs are one of the most popular deep neural networks, showing excellent performance on some complex visual tasks, such as object recognition, image classification, and semantic segmentation. However, CNNs are not restricted to visual perception, proving to be successful in voice recognition or natural language processing applications also [45].

CNNs, much like ANNs, are derived from a biological functioning, the brain's visual cortex. David H. Hubel and Torsten Wiesel's experiments demonstrated that many of the neurons of the visual cortex had small receptive fields. These neurons would only react to stimuli in a limited section of the visual field [32]. The receptive fields of multiple neurons could overlap, and when merged defined the entire visual field. Some differences in the focus of the neurons were also discovered, since two neurons may have the same receptive field but react to different aspects of the image; for example, to different line orientations. These studies eventually evolved into CNNs [32, 45]. The success of CNNs is due to the fewer connections they required when compared to fully connected networks with an equal number of neurons, facilitating the training and, by reducing the number of connections, obtaining a smaller model.

The convolutional networks are assembled by two primary blocks, the convolutional layer and the pooling layer. Figure 19 is a representation of a CNN, which comprises convolutional layers, followed by a ReLU activation, and then a pooling layer. This assembly of layers can be repeated various times, making the network deeper. After this block, a feedforward network is often added, consisting of one, or more, fully connected layer(s) and, in the end, a layer to estimate the output (e.g., a Softmax). While the former blocks are employed to extract features (feature learning phase), the latter is responsible for the prediction (classification phase).

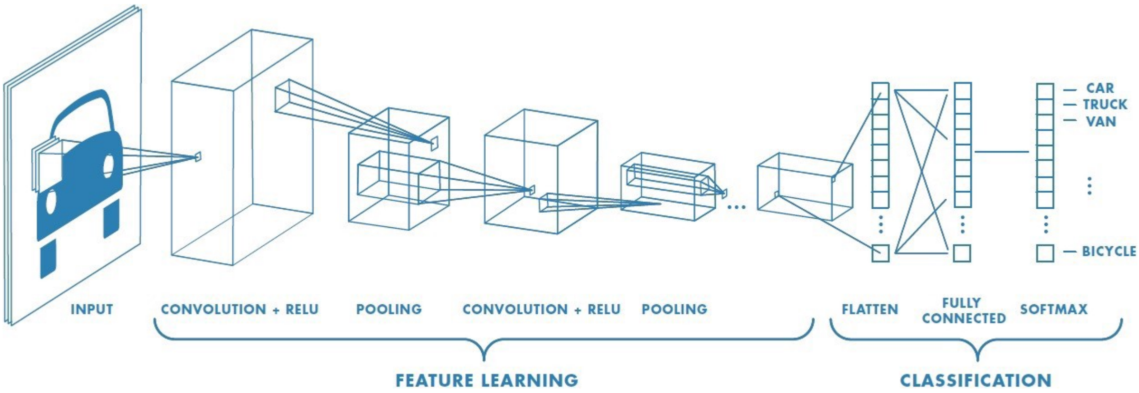


Figure 19: Representation of a basic CNN.
Adapted from [46].

In a convolutional layer, as mentioned, the neurons are not connected to each of the pixels in the input image; instead, they connect to pixels in their receptive field. The first convolutional layer is usually responsible for the extraction of low-level features, such as edges and colours. By the repetition of these layers, and by consecutively converting pixels within the receptive field into a single value, a hierarchy is created, where, by the combination of lower-level neurons, neurons in deeper layers discover more complex, abstract patterns, the so-called high-level features [32]. The convolutional layer is based on the dot product operation between two matrices: the kernel and the respective portion of the receptive field. The kernel, or filter, is a set of learnable parameters, a representation of a neuron's weights as a small image of the same size as the receptive field. Although being spatially smaller than the image, the filter extends in depth to the number of channels of the input data. During the forward pass of any training step, the filter slides through the image, producing an image representation known as a feature map (or activation map). During training, and thanks to the backpropagation algorithm, the convolutional layer learns which filters are better for the task at hand and which combinations to perform in the next layers to obtain more complex patterns, meaning that the CNN will be able to detect various features. The convolutional kernel has a set of hyperparameters that include size, depth and stride. The size of the filter, as a rule, is smaller than the image and is a square, while the depth is equal to the input data's depth. The stride corresponds to the number of pixels between consecutive applications of the kernel while sliding through the input [32, 41].

The pooling layers are often placed after the convolutional layers, as depicted in Figure 19. Neurons in the pooling layer receive the values of multiple feature maps from a certain location in the previous layer and output a single value representing a summary statistic of the neighbour values it receives. Its aim is to reduce computational load and memory usage. Despite having the same hyperparameters as the convolution layer, pooling neurons do not comprehend weights, as it simply aggregates the information received [32, 41]. Two of the most common pooling layers are the Max Pooling and the Average Pooling, in which the output will be the maximum or average value within the filter region, respectively [47].

Over the years, many new CNN architectures have been developed to solve real-world problems. In medical imaging, both 2D and three-dimensional (3D) CNNs have been used to create prediction or segmentation models, for example. Most of these networks are based on some well-known architectures, like AlexNet [48], VGG [49], ResNet [50] and derivatives centred on dealing with 3D data such as R2+1D and R3D [51], Inception [52], or even some more specialised like Models Genesis [53].

2.2 LUS interpretation

With the aim of building an algorithm that automatically interprets LUS clips, the obvious lane is to focus on the development of deep learning classifiers. Although classifiers are not novel in medicine, video-based analysis is still an under-explored domain, particularly due to the necessity of a larger amount of training data compared to frame-based classifiers. Combining that with the fact that LUS is not the standard imaging modality for lung assessment, not a lot of research has been made on this topic. Notwithstanding, there has been a *crescendo* interest in this topic with the COVID-19 pandemic creating the need for a rapid and safe assessment. Since the SARS-CoV-2 lung infection can be detected in ultrasound images, automatic interpretation algorithms for the detection of COVID-19 have emerged recently [20, 23–27]

There are two main deep learning approaches for medical video analysis found in the literature: one separates the video frame per frame and aggregates the classification result to obtain a video-level outcome, and the other uses the video itself as input to the network, i.e. as a 3D data matrix. Most of the proposed networks are CNNs, often concentrating the innovative aspect on the pre- or post-processing phase rather than altering the network architecture itself. This concept of focusing on introducing domain knowledge has gained particular attention over the last few years.

Although clinical ultrasound is a time-dependent function, as a conclusion can usually be reached only by visualising the anatomy across time to assess movement, some studies opt to concentrate on the interpretation of still images, *i.e.* a frame-oriented classification approach [20, 21, 27]. Either in the attribution of COVID-19 severity scores [27], the detection of pathologies (COVID-19, pneumonia and healthy) [20] or in the identification of image findings (B-lines) [21], most authors use a CNN as a base classifier. VGG-16, or an adaptation of it, is a recurrent option, as seen in Born *et al.* [20] and Sloun *et al.* [21]. Frank *et al.* [27], however, although performing tests with VGG-16, obtained the best results with ResNet-18 when developing a severity assessment network. Only the convolutional part of VGG-16 was utilised in [20], with an additional hidden layer of 64 neurons, while in [21] the six convolutional blocks were maintained, with the difference of each having two convolutional layers only. An alternative approach was followed in [27], where the authors did not change the VGG architecture but focused on its input. With the intent of creating a disease severity classifier, the network was designed to receive three channels as input, making it possible to pre-train the network on a natural Red-Green-Blue (RGB) colour model, 3-channel, images (Figure 20). Hereto, the raw image, pleural line and vertical artifacts masks were automatically created using traditional image processing techniques and concatenated to create a 3-channel input, targeting a more domain-specific model.

Still, within frame-based classifiers, investigations have been made concerning video classifiers based on frame classification aggregation [22, 25, 26]. The work presented in Roy *et al.* [25] proposed an algorithm that automatically predicts the pathology score of each frame of a LUS clip and aggregates the scores to generate a disease severity score at video level. Taking advantage of a frame-level and video-level labelled dataset of LUS clips from patients with COVID-19, the network of the classifier, besides predicting the score, also identifies the regions containing the LUS artifacts. This is possible by leveraging a spatial transformer network, which is used to make two different crops

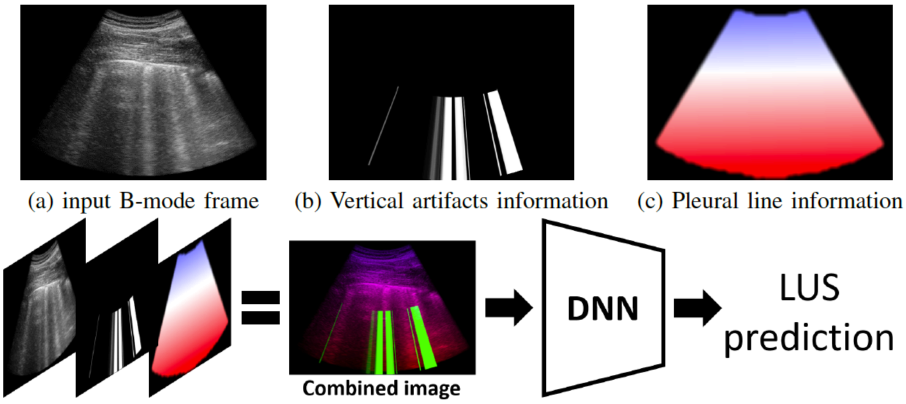

(a) input B-mode frame
(b) Vertical artifacts information
(c) Pleural line information
Combined image
DNN
LUS prediction

Figure 20: Framework for LUS interpretation based on domain knowledge integration. Proposed in Frank *et al.* [27].

of the same frame and enforce the network, a CNN, to give similar predictions of the two. The convolutional network is similar to the one described in [21]. For the aggregation section, a uninorm was utilised. Resorting to the same network, Tsai *et al.* [26] developed a pleural effusion binary classifier for the detection of COVID-19. The results obtained with a frame-level or video-level labelled dataset were compared and it was demonstrated that they were similar. This is viewed as an advantage, especially for clinicians, since the video-level annotation is much simpler and less time-consuming. In [22], Arntfield *et al.* implemented a frame-based classifier of A-lines and B-lines, which was tested both at frame- and video-level. The classifier comprised a CNN network, composed of the first three blocks of VGG-16. If all frames of a LUS clip were classified homogeneously as either A-lines or B-lines, the classifier was applied and the test would be made at frame level. Otherwise, the classifier would be combined with a two-threshold-based algorithm, consisting of a classification threshold and a contiguity threshold applied over the network's prediction, and it would be performed as a clip-based test (Figure 21). With a different approach for temporal aggregation, Barros *et al.* [54] used a pre-trained CNN network as a feature extractor for each of the 20 frames of a video and connected the multiple feature maps temporally through a long short-term memory (LSTM) layer followed by two dense layers that predict the intended output (COVID-19, pneumonia or healthy subject). LSTMs are a type of recurrent neural network and, as such, have the ability to learn long-term dependencies, due to which their use in video processing has been increasing. Although having a small dataset (185 videos), the authors in [54] concluded that a hybrid network consisting of a CNN and a LSTM provided better results than a spatial-only approach, and with less trainable parameters.

On what concerns video-based techniques, most of the works use a 3D CNN as backbone for handling the raw data, i.e the LUS clip [23, 28]. Baloescu *et al.* [28] used this model for a B-line quantification classifier, while Edabi *et al.* [23] incorporated the convolutional network in a two-stream architecture. Each stream was composed of a 3D CNN (I3D), with one receiving the RGB image while the other expected the image's optical flow. The two outputs were combined to classify the image as having either A-lines, B-lines or consolidation/pleural effusion. To compare the two

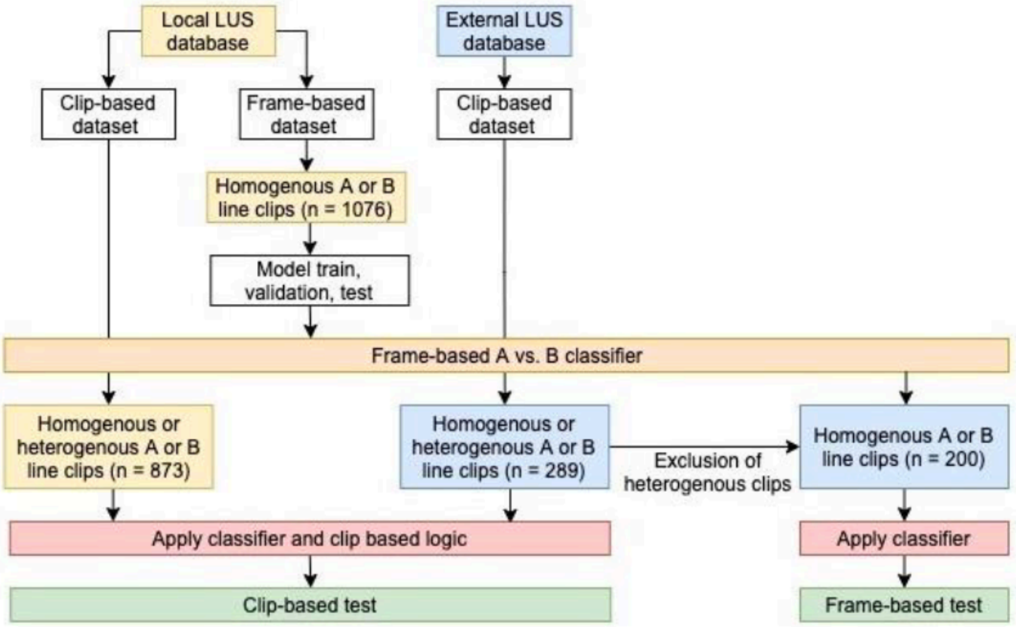

Local LUS database
External LUS database
Clip-based dataset
Frame-based dataset
Clip-based dataset
Homogenous A or B line clips (n = 1076)
Model train, validation, test
Frame-based A vs. B classifier
Homogenous or heterogenous A or B line clips (n = 873)
Homogenous or heterogenous A or B line clips (n = 289)
Exclusion of heterogenous clips
Homogenous A or B line clips (n = 200)
Apply classifier and clip based logic
Apply classifier
Clip-based test
Frame-based test

Figure 21: Flowchart of the dataset creation, labelling, data allocation and training used in Artnfield *et al.*

last mentioned approaches, Born*et al.* [24], based on his previous work [20], presented a side-by-side comparison of two networks, a frame-based model with VGG-16 as the backbone and a video-based model using a 3D CNN (Models Genesis), for a LUS pathology classifier (COVID-19, bacterial pneumonia, viral pneumonia and healthy). To achieve a video result in the frame-based classifier, and differently from [25], the authors proposed a confidence estimate selector that discards frames with low confidence before summarising the frame result. The frame classifier outperformed the video one. However, the authors stated that 3234 images were used for the first while only 770 videos were fed as training data in the latter, raising the hypothesis that the conclusions may have been constrained by the dataset.

Regarding data augmentation, most of the mentioned works applied the standard transformations, i.e flips, small rotations, translations, crops, contrast and brightness variations. Notwithstanding, Baloescu *et al.* [28], although applying those augmentations, added a time reversal transform in their algorithm. Since the lung movement does not have a relevant sequence, this alteration would not compromise the B-line quantification, which was the purpose of their work.

In terms of adaptability to the domain, some authors have focused part of their work and research on preparing the data according to the topic of analysis. Both [27] and [28] dealt with the B-lines artifact. Although not giving the appearance due to the use of convex probes, this artifact is orthogonal to the pleural line and in the direction of the ultrasound beam. As a result, these authors decided to transform the LUS clip to a consistent rectilinear format (converting from Cartesian coordinates to polar ones), so that the lines would be vertically aligned with the image axes. This allows for a simpler detection, since their algorithm searches for bright columns in the lower half of the frame,

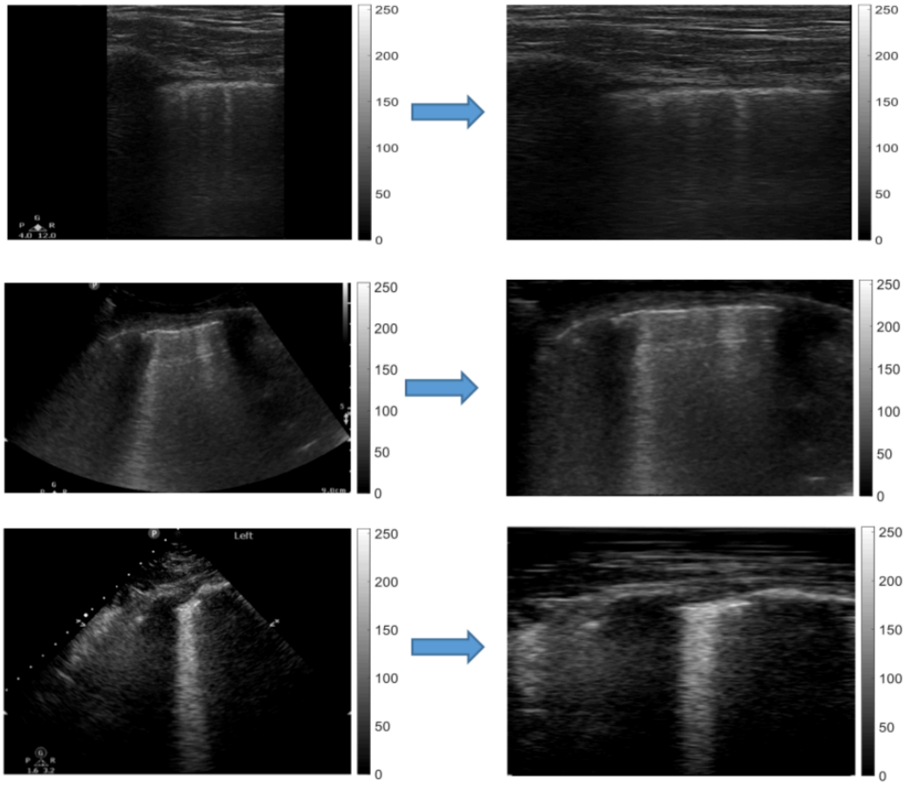


Figure 22: Examples illustrating the pre-processing of frames to rectilinear format in Baloescu *et al.*

where there is less noise present. For [28], this transformation permitted the standardization of the data across different formats.

2.3 Common challenges with CNNs

One of the most important tasks to solve when facing a deep learning problem is the understanding of the data. There are a set of characteristics that the dataset must comprehend to be able to train a successful algorithm. Most networks require a substantial amount of data in order to work properly, but a big dataset is not enough. It is also crucial that the data is representative of the cases that we intend for the model to generalise for; in other words, it must be a representation of the population. Conjointly, the fewer errors, noise and outliers a dataset has, the better its chance of performing as desired [32, 55].

However, a perfect dataset is not commonly achievable and, as such, there are certain methods to handle these challenges. One of the most frequent problems one encounters in deep learning is overfitting. This is when a model fits too perfectly the training data and lacks the ability to generalise to unseen data, due to either a small dataset or a noisy one, or due to the high number of neurons in the neural network. Regularisation is often applied to tackle this. Early stopping is one regularisation technique, stopping the training when the loss function stops decreasing. Another method highly used is dropout, where some hidden neurons are randomly ignored during training. In short, at every training step, a few neurons are excluded by setting their outputs to zero. The probability of this occurrence is the dropout rate. With this technique, the performance of the algorithm improves as it becomes less sensitive to small changes and, therefore, generalises better. L1 and L2 regularisation are other options, and they take action by constraining the networks' weights. Data augmentation is the generation of new data by performing transformations to the existing data, increasing the number of instances. A bigger dataset presents the network with varied information and, therefore, is more probable that the algorithm learns to generalise, decreasing the overfit. Lastly, batch normalisation (BN) consists of adding an operation to the model, before or after the activation function, zero-centring and normalising the inputs, followed by scaling and shifting. The optimal values for these parameters are computed by estimating the mean and standard deviation over the training set [32,41,55]. Across the DL-based LUS interpretation literature, researches have often opted to use dropout [25,26], while others employ data augmentation [24], or batch normalisation [54]. Others, like in [22], chose to design a smaller architecture, making it less prone to overfitting.

Datasets are usually labelled by humans, which although often being experts in the topic are susceptible to errors, especially since many of the labelling tasks are monotonous and tedious. These wrongly classified and unreliable labels are named noisy labels and, usually, their presence leads to a poorly performing model, due to the inability to generalise. Although the aforementioned techniques to deal with overfitting may be helpful in certain scenarios, in others they might not be sufficient. To this end, several studies have surfaced on learning under noisy conditions, however, most of the applications are focused on categorical (binary or multi-class) problems, neglecting multi-label problems [32, 55, 56].

For categorical settings, a plenitude of techniques has been proposed, from adjusting the network by adding a noise adaptation layer, to sample selection or the implementation of robust loss functions. As detailed in Song *et al.* [56], robust training can be depicted into four main groups: robust architectures, robust regularisation, robust loss design and robust sample selection. While, as the name implies, a robust architecture approach requires altering the network itself, the remaining can be used in most of the already existing algorithms. Regarding loss functions, Zhang *et al.* [57] combined the advantages of both categorical cross entropy (CCE), which performs well for difficult data, and mean absolute error (MAE), which has the ability to generalise better than CCE, by creating a generalised cross entropy (GCE). Symmetric cross entropy learning (SL) [58] is an enhancement of the vanilla CCE by combining it with its symmetric, the reverse cross entropy. In a more generic approach, Ma *et al.* [59] proposed a framework to build a robust loss, denominated active passive loss (APL), which combines two loss functions: one active, that only maximises the network's output probability for the correct class, and one passive loss, that also minimises the probability of the remaining classes [56,60]. For multi-label classification, the research is scarce, with label smoothing being the main approach. Label smoothing is a regularisation technique that prevents the model from being overly confident by smoothing the one-hot vectors, instead of relying on them solely [56, 60]. Especially for LUS interpretation, not many authors have directly tackled this problem. However, when dealing with frame inputs, some researchers, such as Frank *et al.* [27] and Roy *et al.* [25], opted to remove what they considered ambiguous frames to minimise the noise in the dataset.

One other common problem encountered in deep learning classifiers is class imbalance, and that becomes even more evident in a multi-label setting. This complication takes effect when some classes have a higher number of samples than the other classes in the dataset. Under the influence of an imbalanced dataset, the models will often overclassify the majority group, with the model showing a higher probability to wrongly classify the minority group samples. This happened because the update of the model's weights is more influenced by the gradients resulting from the majority class samples, translating in a decrease in the error of these samples but an increase in the remaining ones, ultimately affecting the model's convergence and generalisation. The most straightforward approach to mitigate this issue relies on sampling methods, either by over- or undersampling. Oversampling consists of, in a random manner, replicating samples from the minority class, which increases the size of the dataset and may be prone to overfitting. Undersampling is the exact opposite, in which data from the majority class are discarded, balancing the number of samples per class but usually resulting in a lower amount of total samples to train the model [61, 62]. Some loss functions have been developed to allay this issue, like the focal loss [63]. This loss focuses on learning hard misclassified samples and consists in a scaled cross entropy loss, in which said scaling factor diminishes to zero with the increase of the confidence in the true class.

In conclusion, a big dataset with careful and precise annotations would solve many of the most commonly found challenges in deep learning. However, the shortage is not, primarily, due to the data itself, but to the labels required. Since many developers do not have the expertise to curate the datasets themselves, and since it is a time-consuming task, one approach to enhance the datasets has been to leverage semi- or self-supervised learning techniques. By taking advantage of the unlabelled data, these algorithms work as a path to either learn the general features of the dataset or even produce said labels.

2.4 Semi-supervised learning

Semi-supervised learning (SSL), as mentioned earlier, is a meet-in-the-middle approach between supervised and non-supervised learning. It provides an opportunity for the models to benefit from unlabelled data, while simultaneously maximising the useful information provided by the, usually, smaller labelled dataset.

Semi-supervised learning algorithms were constructed as a tool to improve the performance of either supervised or self-supervised tasks by taking advantage of the insight granted by the other. Considering a dataset D that contains two subgroups, the labelled dataset, D_l, and the unlabelled one, D_u, SSL main intention is to leverage D_u to train a model that would give the most similar performance possible to the one that would be obtained if the whole dataset, $D = D_l \cup D_u$, was labelled. This is possible to be accomplished since D_u may be able to provide information about the structure of the dataset, or even about some features, that may lead to a better decision boundary between the different classes that, otherwise, would not be known [64–67].

In the past few year, a plethora of SSL methods have emerged as a result of the growing interest in leveraging DL algorithms in fields where labelled datasets are scarce or hard to obtain. Ouali *et al.* [65] divided the semi-supervised algorithms into four subgroups: consistency regularisation, proxy-label methods, generative models and graph-based methods. Broadly, **consistency regularisation** is based on the belief that the model prediction should not change if a set of realistic augmentations are applied to the data. Thus, the network is trained to give a consistent result on either D_u or the augmented D_u version. **Proxy-label methods** are algorithms that aim to label D_u based on heuristic rules or using a model trained on a smaller labelled set. Analogously to a supervised approach, **generative models** use D_u to learn features that will later be transferred to the supervised task with the intended targets. **Graph-based methods** are singular in the aspect that their base concept is that both labelled and unlabelled data points can be viewed as the nodes of a graph and one can propagate the D_l nodes to D_u nodes through some measure of similarity.

Given their relevance in the context of the present thesis, the rest of this section describes in more detail the literature on proxy-label methods. Proxy labels are approximated labels, originally not present in the dataset, obtained by using a prediction function or a variant in a non-supervised setting. In proxy-label algorithms, these created labels are used during training alongside the labelled data. One example of this category of algorithms is pseudo-labelling. Pseudo-labelling is a method in which the intended network is trained in a supervised manner with the entire dataset, D. As one may wonder, this approach may seem only conceivable if D_u is also labelled. The idea is thus to attribute pseudo-labels to D_u. To do so, one uses a model pre-trained on a smaller labelled dataset (D_l) and assumes the class with the highest probability (if binary/multi-class) or those above a given threshold (if multi-label) as ground truth label(s). The pseudo-label phase is understood as a fine-tuning of the previously pre-trained, using the entire dataset and respective true or pseudo-labels. Meta pseudo labels are another approach that proposes a student-teacher setting. Pham *et al.* [68] presented meta pseudo labelling as a derivative of pseudo labelling, that consists of two phases: the first step in which the student learns from the teacher, with the teacher producing a target class distribution and the student updating its parameters subsequently; and a second step during which the teacher is updated based on the performance of the student on D_l. This differs from pseudo-labelling since, in the latter, the teacher is a fixed pre-trained model.

However, pseudo-labelling has the downside of often overfitting to incorrectly attributed pseudo-labels, which is known as confirmation bias. Holistic methods have arisen as a useful resource to deal with this bias alongside label smoothing [69]. They are a branch of proxy labels that incorporate core ideas of the dominant paradigms mentioned above, with the exception of graph-based methods. One example is MixMatch [70]. Its idea is to produce for each batch of labelled and unlabelled data, K augmented labelled (A_l) and unlabelled (A_u) samples. Then, the proxy labels are generated for the augmented data versions and an average of the K predictions is calculated to obtain a pseudo-label for each example of A_l and A_u, which are then sharpened having in consideration the categorical distribution of the classes. To finish, the two batches of labelled and unlabelled data are merged and combined with data that was not included in this fusion, the so-called MixUp, and trained with a cross-entropy loss for supervised samples and a consistency loss for the unsupervised fraction. In its turn, FixMatch [71] uses two types of augmentations, one weak and one strong. The model is first trained to predict a pseudo-label for a weakly augmented version of the unlabelled data, and from these only the ones with the highest confidence are maintained. Afterwards, the model is trained to predict the same label for strongly augmented versions of said images.

It is relevant to reference that most of these algorithms use a weighted strategy across the mini-batches and, therefore, assume large training batches. Consequently, these algorithms may not be applicable in all settings and depend a great deal on the type of computation available. In addition, although not much research on LUS interpretation has emerged with a focus on semi-supervised learning, one can deduce that, given the nature of some of these algorithms, not all may work for this type of medical images. Models such as MixUp and FixMatch may not be the best choice since certain (strong) augmentations change the ground truth label or, at least, change the appearance of the artifacts (or diminish their presence) to an extent that the image is no longer interpretable.

3

METHODOLOGY

3.1 General overview

In this project, a novel, flexible DL-based framework for the automatic interpretation of lung POCUS video is proposed (Figure 23). The proposed framework rests on four core blocks: (1) a pre-processing block; (2) a supervised learning block; (3) a semi-supervised learning block; and (4) a model ensemble block.

In brief, the pre-processing block (Section 3.3) is responsible for standardising the input data, masking out any information outside the LUS FOV and splitting the video into multiple (smaller) overlapping clips. The supervised learning block (Section 3.4) utilises the available labelled data to train a 3D CNN model (Section 3.4.1), while employing data augmentation techniques (Section 3.4.2) and label smoothing regularisation (Section 3.4.3). The trained classifier provides per-clip outputs that are then aggregated into a video-level prediction (Section 3.4.4). In the semi-supervised learning module (Section 3.5), one employs the previously trained classifier to predict pseudo-labels for the unlabelled dataset and select those with high-confidence and low-uncertainty using the Uncertainty-aware Pseudo-labelling Selection (UPS) method [72]. Taking advantage of both labelled and pseudo-labelled samples, one trains the proposed 3D model once more from scratch. This step may be repeated multiple times, updating the pseudo-labels with the most recently trained classifier. The performance of the resulting classifier (or of that obtained through supervised training if an unlabelled set is unavailable or its use is ineffective) is further boosted through model ensemble techniques (Section 3.6), including a novel ensemble strategy that leverages of the hierarchy inherent to the LUS evaluation. The latter may also be enforced through a (optional) post-processing routine that applies conditional inference through a set of ad-hoc rules (Section 3.7).

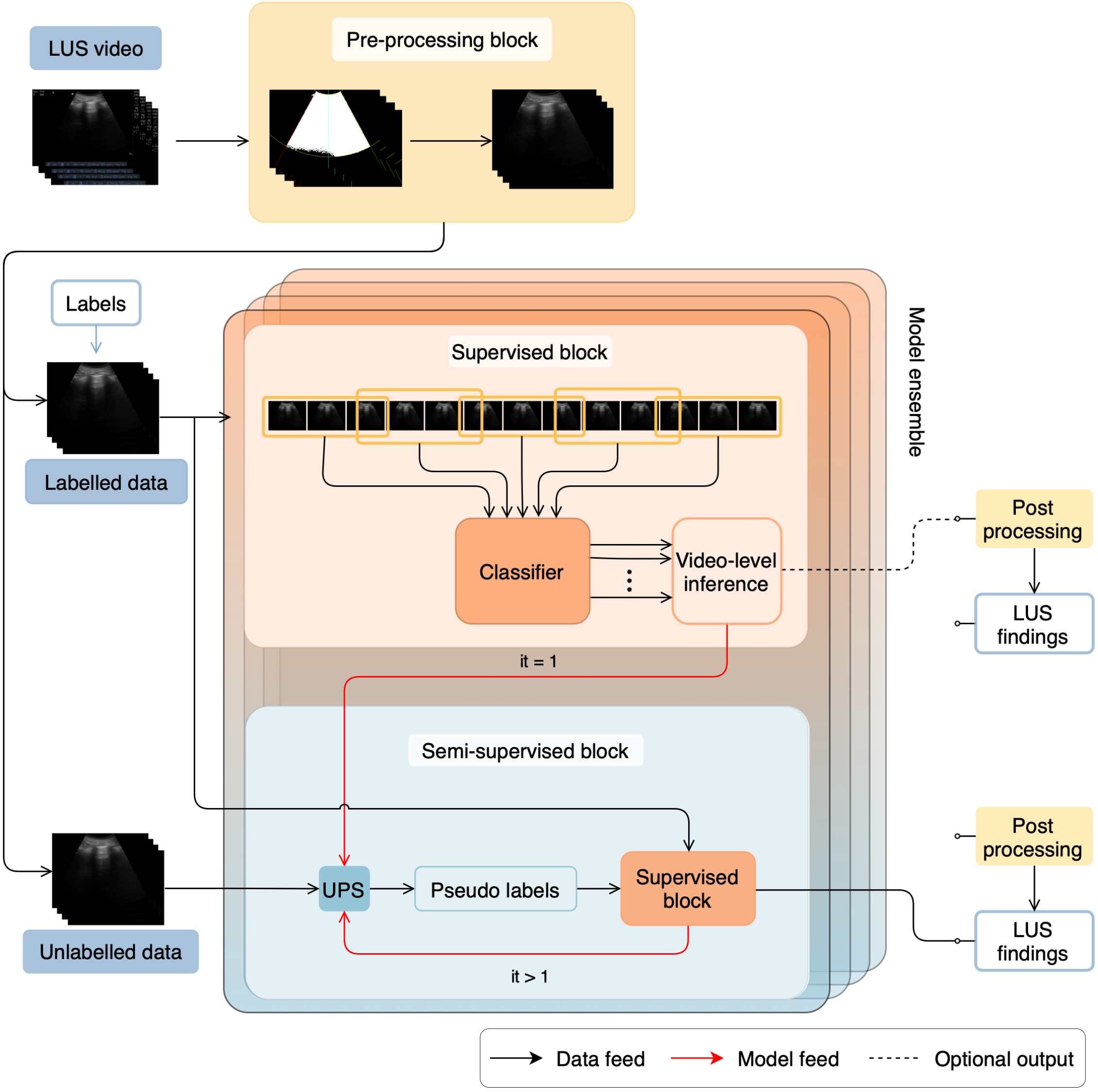


Figure 23: General overview of the proposed method for LUS video interpretation.

3.2 Dataset

The dataset is composed of videos of point-of-care ultrasound acquired at *Hospital de Clínicas de Porto Alegre* (HCAP) from 2020 onward, with the ethical approval of the Ethics Committee for Research (ECR) in Life and Health Sciences of the University of Minho (CEICVS 039/202) and the ECR of HCAP (5.334.879). The footage was obtained with two different transducers (convex and linear), being discarded all but the B-mode videos performed with a convex probe. Over this period, 990 patients were evaluated, presenting symptoms or not, by multiple physicians, which translates into 8160 LUS videos in total. The frame rate varies from 25 to 60 frames per second, with most of the videos having 6 seconds each. All lung ultrasound videos were available in MP4 format and were de-identified using *Clip Deidentifier* [73].

Four medical doctors with expertise in the POCUS technique annotated the dataset. For this labelling process, an online platform was used [74], where one would upload the data and the expert would label it, by identifying the relevant finding(s) present in each video. To guarantee a consensus between experts on what should be contemplated as relevant findings and how to classify them, meetings were arranged to reach an agreement. Of the 8160 videos, 3649 were manually annotated and therefore used in the supervised and semi-supervised settings (Section 3.4 and Section 3.5). The remnant 4511 were left without annotations and were utilised only in the semi-supervised approach (Section 3.5).

The dataset consisted of lung POCUS videos, of any of the fields described in Section 1.3.4. These videos are divided by exam, and an exam does not have to necessarily contain all fields mentioned. Additionally, there may be repetitions of the same field, resulting in a variable number of videos per exam. Since different fields may highlight distinct features and be associated with a different set of findings, in this work, one focused only on a sub-set of them which share the same list of relevant findings. Table 1 summarises the included fields and associated findings.

Figure 24 represents the labelled dataset distribution, used throughout Sections 3.4 and 3.5.

The LUS findings can be grouped hierarchically, as represented in Figure 25. First, they can be divided into two big groups: those considered to be of a normal lung, and those indicative of an underlying pathology. The presence of any of these indicative findings overshadows the presence of a normal finding. In the normal findings, one finds the scattering artifact and the A-lines pattern. Within the indicative findings, there are those that, alone, are not sufficient to conclude the presence of a pathology and those that are proof of such. In the first category, we can find pleural irregularity and the presence of less than 3 B-lines. In the latter, one considered two sub-categories: the positive

Table 1: LUS fields and respective findings

Fields	Findings
RL - AS	
LL - AS	Scattering artifact only
RL - AI	A-lines only
LL - AI	Up to 3 B-lines
RL - LS	More than 3 B-lines
LL - LS	Coalescent B-lines
RL - LI	Consolidation
LL - LI	Pleural effusion
RL - PS	Pleural irregularity
LL - PS	
RL - PI	
LL - PI	

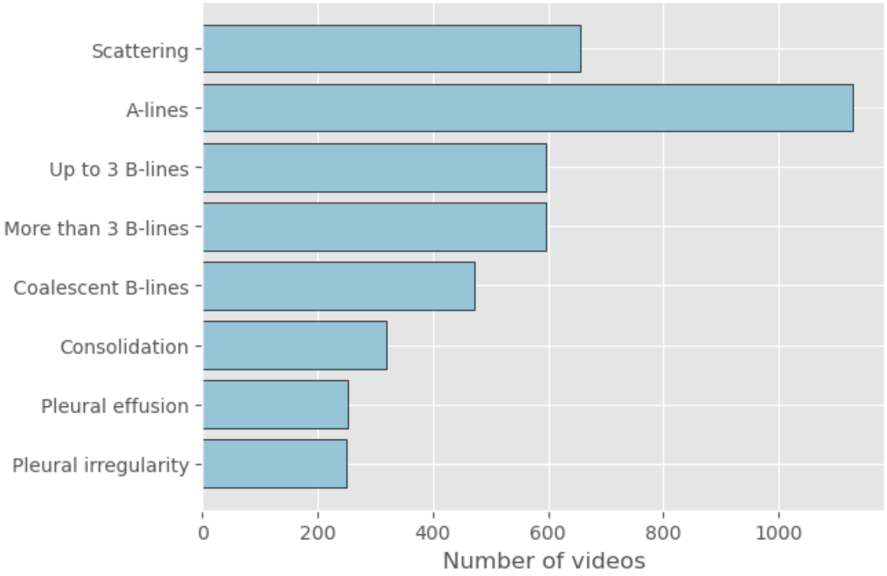


Figure 24: Distribution of the labelled dataset.

B-line pattern, which includes the presence of more than 3 B-lines or coalescent B-lines; and other significant pathologies, such as consolidation and pleural effusion.

This hierarchy and categorisation will be considered throughout the present work. For more clarity, possible labels are divided into two categories: leaf-labels and high-level labels. Leaf-labels are the labels that were annotated by the experts (scattering artifact, A-lines pattern, pleural irregularity, up

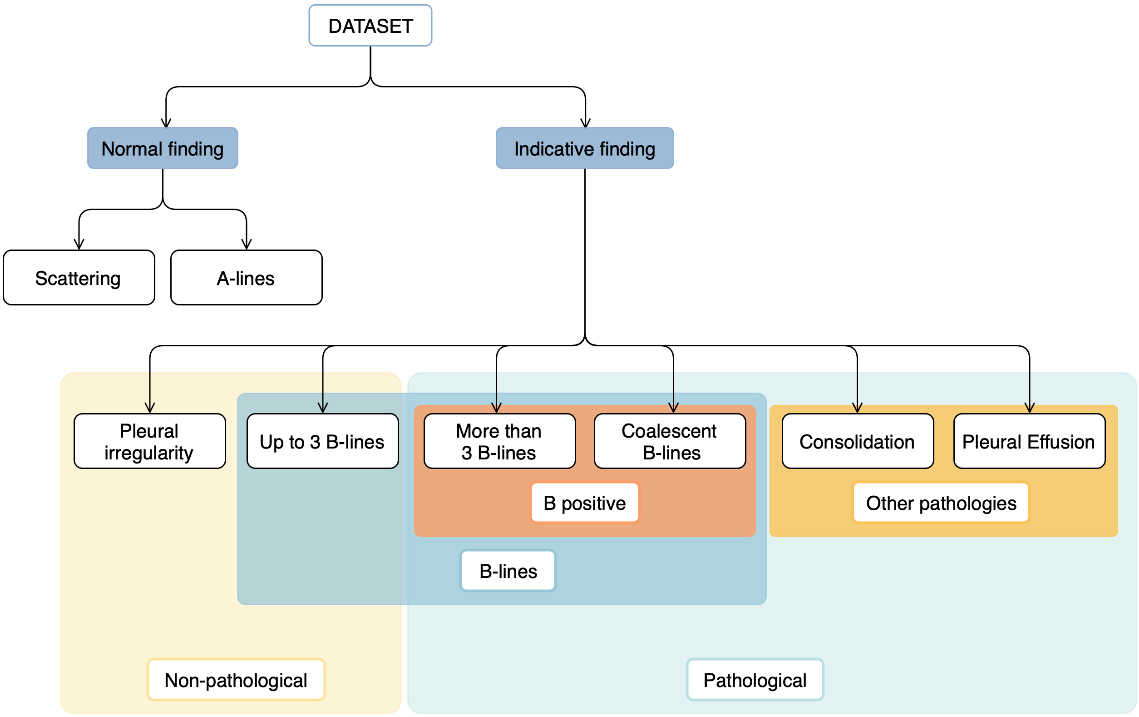


Figure 25: Dataset annotation categorisation.

to 3 B-lines, more than 3 B-lines, coalescent B-lines, consolidation and pleural effusion). The parent labels (normal and indicative artifact) and the aggregation labels (non-pathological, pathological, B-lines, B positive and other pathologies) will be referenced as the high-level labels.

As explained in Section 1.3.3, some findings are more subjective than others. Specifically, the number of B-lines present may be a ground of discussion between physicians, as there is a fine line between the definition of 'More than 3 B-lines' and 'Coalescent B-lines'. 'Pleural irregularity' is also a misleading finding since it is a very small artifact and its presence alone is many often unnoticed. The same may occur with both 'Consolidation' and 'Pleural effusion' depending on their degree of severity and size. This user-dependent interpretation makes the dataset susceptible to the presence of noisy labels.

3.3 Data pre-processing

Surrounding the ultrasound sector scan, LUS videos have additional information and marks, such as the indication of the ultrasound depth and details regarding the ultrasound machine itself. This superfluous information may have a negative influence when training a deep learning model, as the algorithm may focus on it during its learning process. For this reason, all videos must be pre-processed to mask out any information outside the FOV. However, when analysing these videos, one aspect that is evident and potentially unfavourable for such pre-processing is the presence of pixels inside the ultrasound FOV with the same intensity value as those outside it (very low values). Since the LUS videos are, for the most part, very dark due to the presence of air, they often have a deteriorated image that complicates the estimation of the FOV mask, especially considering that, ideally, the mask should have a shape equal to the ultrasound sector scan. These facts preclude the use of simple thresholding operations, and thus a novel pre-processing algorithm was created, whose sole input is the video itself. Figure 26 illustrates the simplified flowchart of the proposed algorithm, which was implemented in MATLAB (MathWorks Inc, USA).

In short, the method builds the mask by finding certain points, fitting lines/circles and attributing confidence values to each. Figure 27 is an example of the original ultrasound image.

Binary thresholding

First, a thresholding operation was applied to create a a raw mask of the LUS FOV. Although the outside of the FOV appears to be black, its intensities are not always zero. An intensity value of 2 was thus used as threshold and applied to the video. The resulting binary map is summed across all frames and the result is once again thresholded to identify those locations in which valid information appears at least once during the video (i.e. value in summed map above 0).

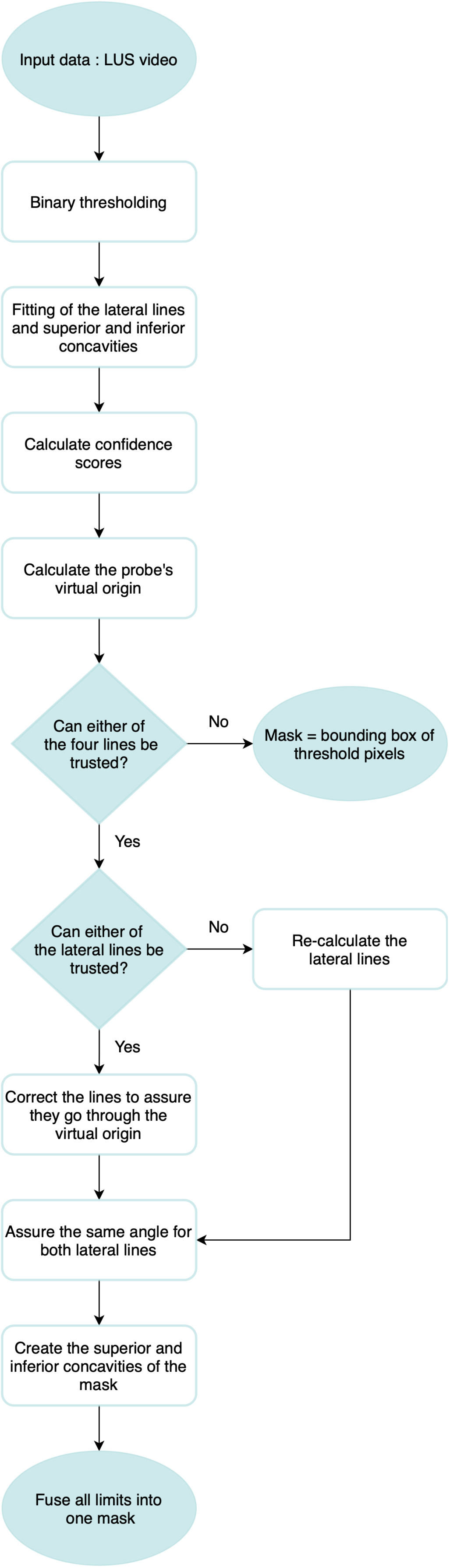

Input data : LUS video
Binary thresholding
Fitting of the lateral lines and superior and inferior concavities
Calculate confidence scores
Calculate the probe's virtual origin
Can either of the four lines be trusted?
No
Mask = bounding box of threshold pixels
Yes
Can either of the lateral lines be trusted?
No
Re-calculate the lateral lines
Yes
Correct the lines to assure they go through the virtual origin
Assure the same angle for both lateral lines
Create the superior and inferior concavities of the mask
Fuse all limits into one mask

Figure 26: Flowchart of the pre-preprocessing algorithm.

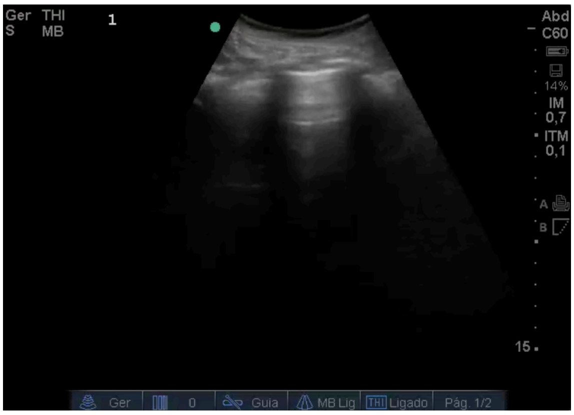


Figure 27: Example of a LUS frame before pre-processing.

Although the resulting mask contains the relevant pixels inside the sector scan, is also contains all non-zero pixels belonging to the superfluous information. To filter these unwanted details, a shape- and area-based filter operation was applied through connected components analysis. In simple terms, only non-squared components with an area above 20,000 pixels were maintained (smaller regions represent text and a large squared region represents the bottom menu present in most videos; see example in Figure 27). An example of the resulting map is shown in white in Figure 28 and Figure 29.

At this stage, a bounding box containing all selected pixels is also created.

Fitting of the lateral lines and superior/inferior arcs

Based on the binary mask created, an approximation of the lateral lines and superior/inferior arcs of the sector scan is made.

To identify the optimal lateral lines (Figure 28), it is necessary to find the most outward pixels of the binary mask, for both left and right sides. This is obtained by two search procedures, each searching for the first non-zero pixels (i.e. boundary pixels) from the respective bounding box limit (left or right) to the middle of it. For either side, a regression for the line that best fits the identified set of pixels is calculated.

The same idea is followed for the superior and inferior arcs (Figure 29). For the superior arc, by running a search procedure on the image columns, one searches for the boundary pixels with the lower y-coordinate on either half of the bounding box. All top boundary pixels located between these two identified pixels are considered as probable points of the superior arc. The superior arc is then estimated as the circumference that best fits these points. For the inferior concavity, the process is similar, with the difference that the search is applied from the bottom to all image columns. The inferior arc is then estimated by fitting a circumference to the detected boundary pixels.

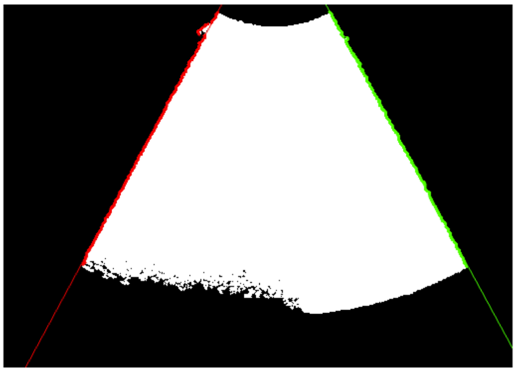


Figure 28: Fit of the lateral lines.

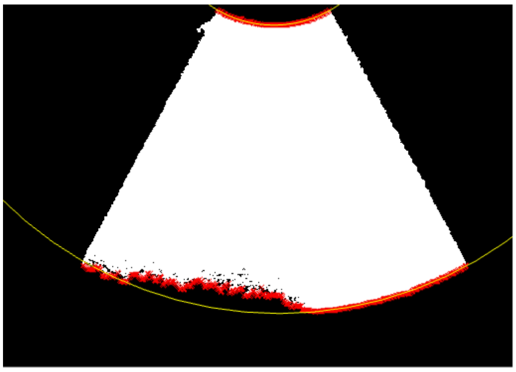


Figure 29: Fit of the superior and inferior concavities.

Line's confidences estimation

To validate if the fitted lines are accurate and can, indeed, be used to create the sector scan mask, a confidence value is calculated for each line.

For the lateral lines, this confidence is defined by two parameters: the number of pixels from the mask that are either left of the left line or right of the right line; and the angle of the line in relation to the vertical axis. If the number of points outward of the lines is greater than 500 and the angle is higher than 45 degrees, the line cannot be trusted, and its confidence is set to zero. This can occur for just one line or both. In opposition, if the line can be trusted, the confidence takes the value of 1.

For the arcs, to assure a satisfactory fitting, one compares it with the fitting of a line to the same set of pixels. In simple terms, one calculates the percentage of inliers (defined as those points whose distance to the closest point in the regressed arc/line is equal to or less than 2 pixels) when fitting the circumference and the line. If the percentage of inliers for the circumference is higher, then this value is used as confidence. Otherwise, the confidence is set to zero.

Probe's virtual origin estimation

With all the confidences calculated, the algorithm has all the information to choose which approximation will be the reference for the creation of the mask.

At this stage, one has two lateral lines and two arcs identified. Independently of being able to trust them or not, a problem exists with these calculated regressions. They were all performed independently of each other and, as such, do not have a common vanishing point, which occurs in a real LUS FOV. This is defined as the probe's virtual origin and it is calculated by a weighted average that takes into consideration the created lines' intersections and respective confidences. Specifically, five intersections are estimated: the intersection of the left line with the right line, and the intersections of each lateral line and the vertical line that passes through each circumference center. The weighted average is obtained by averaging the calculated intersection points' coordinates, weighted by the confidences implied in that intersection, and each of the arcs' centres, multiplied by their respective confidence.

Once the virtual origin is estimated, to complete the LUS sector scan, both lateral lines and arcs must be adapted to the origin. The virtual origin of the probe will be the mask's centre. This information, along with the opening angles (defined by the lateral lines) and the arcs' radii, must be provided by the algorithm, since they are needed for the data augmentation routine proposed in Section 3.4.2.

Lines and concavities re-adjustments

To adapt each of the four lines, their reliability must be taken into consideration. If none of the lines can be trusted, the mask will be the bounding box of the initial binary map. Consequently, the probe's virtual origin will be the middle top point of the image, the opening angles will be 0 and 180 degrees, the smaller radii will be 0 and the larger will be equal to the height of the image.

If the two lateral lines cannot be trusted, they are re-calculated. The only requirement is that the angle defined by the lateral lines (in relation to the vertical axis) creates a sector that includes all pixels from the initial binary map.

If either one of the lateral lines or even both can be trusted, these will also be recomputed, but in a different manner. A new line is fitted to the set of points originally used for line regression, but adding the constraint that the regressed line must pass through the probe's virtual origin. Since the LUS sector scan is known to be symmetrical, one must further guarantee that these lines are mirrored with respect to the vertical axis. Thus, if both lines were recomputed, their angle with the vertical axis is computed, with the wider angle kept and the other line adjusted. The final angles are stored as opening angles.

Afterwards, the bounding box is centred in relation to the probe's virtual origin. For the superior arc, the radius is obtained by computing the intersection of the left lateral line with the superior bounding box limit and calculating the distance of this intersection to the virtual origin. The approach is more straightforward for the inferior concavity since the radius is the distance from the bounding box's bottom limit to the virtual origin.

The identified probe's virtual origin, arcs' radii and lines (Figure 30) are ultimately used to create a binary mask of the sector scan (Figure 31), which is used to mask and crop the input video. The coordinates of the virtual origin, the arcs' radii and the opening angles are also stored in a JavaScript Object Notation (JSON) file.

Data standardisation

The input video is then scaled to 128x128 pixels by nearest neighbour interpolation (to minimise smoothing effects on the data) with padding being used when needed. Additionally, the pixel values are converted to grey-scale colour space and are normalised between 0 and 1.

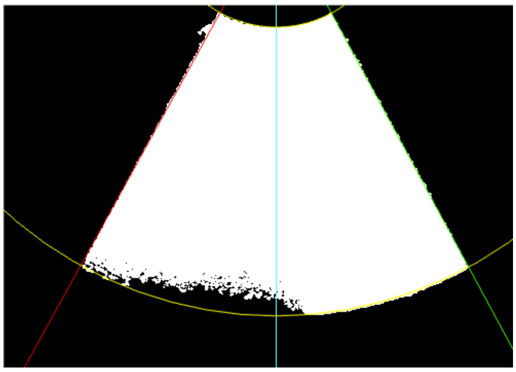


Figure 30: Example of the final lines and arcs estimated.

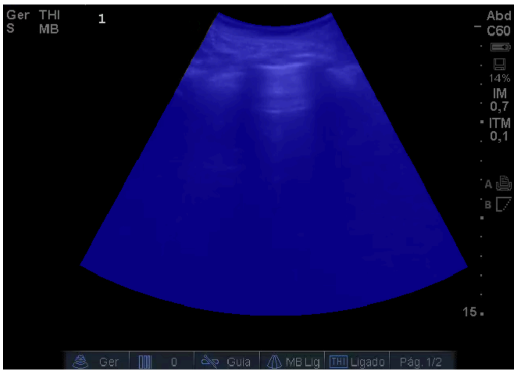


Figure 31: Example of the final mask (in blue), overlaid on the original image.

3.4 Supervised setting

Since video clips of, at least, 4 seconds are necessary to perform an adequate LUS assessment (given the average period of a respiratory cycle), the network's input was set to 32 frames, at a frame rate of 8 Hz. These video clips are extracted from the labelled LUS videos, by randomly selecting the first frame of the clip (always guaranteeing a set of 32 frames). If the input video has less than 32 frames, empty frames are added.

Regarding the training details, the Adam optimiser [75] was employed, running 100 epochs with a batch size of 4 and with weights initialised by a normal distribution (as presented in [76]). The learning rate was initialised at 1x10^{-3}, and decreased using a cosine decay learning schedule [77]. For the loss function, categorical/binary cross entropy was used, depending if the outputs were multi-class or multi-label, respectively. The labelled dataset was split into 80% for training and 20% for testing, with the training portion used in a 5-fold cross-validation for algorithm development. The training was performed in a mixed precision setting, which reduces memory usage and accelerates training time, without jeopardising convergence.

The pipeline was developed in the Keras framework [78] with TensorFlow backend [79], resorting to a workstation with a Nvidia RTX A6000, 64 GB of RAM and an Intel(R) Core(TM) i9-12900F CPU.

3.4.1 *Network architecture*

For the development of the video-based classifier, one focused on networks that dealt with 3D data as input. When analysing which architectures could have the best performance, the work of Tran *et al.* [51] stood out due to the good results granted without the computational overload that other 3D techniques presented. Consequently, a R2+1D network with 18 layers is proposed as the baseline architecture for the classifier, whose visual representation is presented in Figure 32.

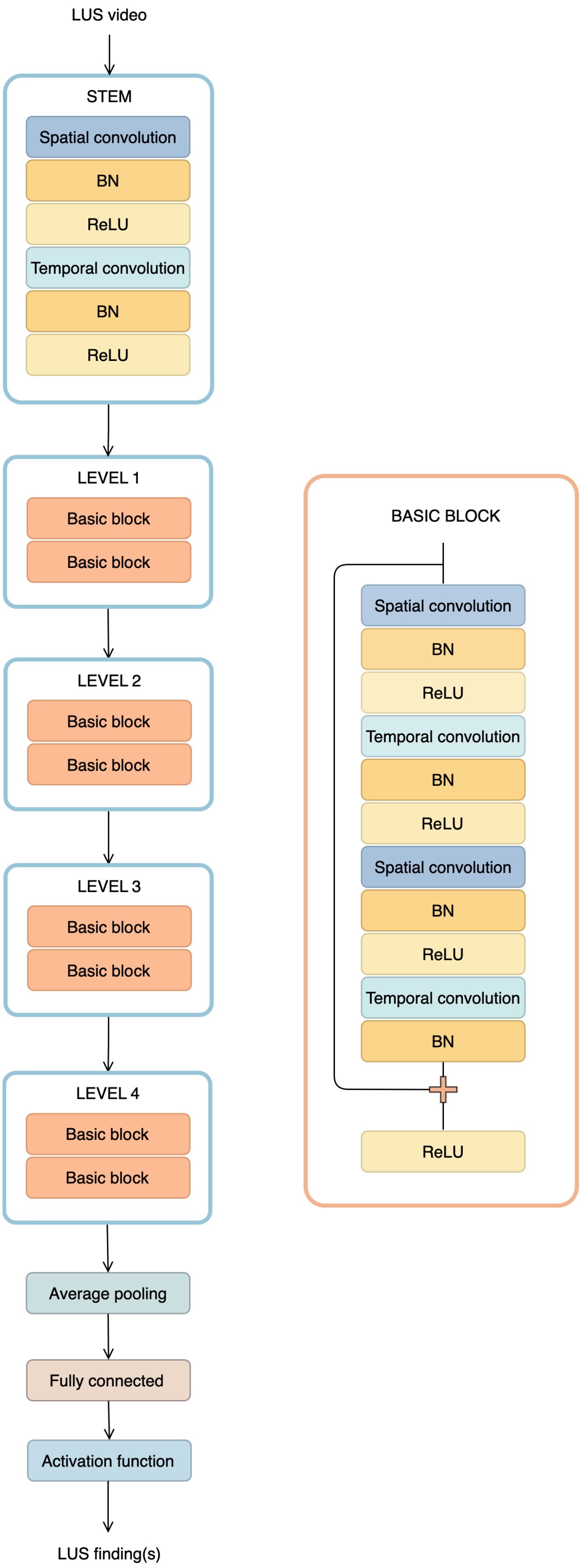


Figure 32: Representation of the R2+1D architecture.

R2+1D is a ResNet-based architecture and gets its name due to the factorisation of the 3D convolutional filters into spatial (2D) and temporal (one-dimensional (1D)) blocks. The aim is that, by having an additional non-linearity between these operations, the number of nonlinear rectifications doubles compared to the non-factorised version, R3D. This translates into a model capable of performing more complex functions. Additionally, differently from 3D filters where the dynamics are intertwined, R2+1D provides an easier model to optimise, resulting in an overall lower testing and training error, as demonstrated by Tran *et al.* [51]. Differently from [51], the proposed network does not consider an increased number of filters per convolutional layer, which reduces the number of parameters compared to a R3D network and, consequently, its memory usage and training time.

The network receives as input a LUS clip of 128x128 pixels. This size was chosen to be the smallest possible while still being interpretive. With 32 frames per clip, the standardised LUS clip enters the stem block, a convolutional block, followed by four identical levels of residual convolutions. Each of the levels can be divided into two basic blocks, as seen in Figure 32, composed of consecutive spatial-temporal-spatial-temporal stages, each accompanied by BN and ReLU. Each convolutional layer is composed of 3x3x3 kernels with the exception of the stem which relies on a 7x7x3 kernel. One should notice that each of these kernels, just like the convolutions themselves, is factorised into spatial and temporal kernels. In the residual blocks, the skip connection is added to the result of the last BN, and is proceeded by the final ReLU layer. One spatial downsampling is performed in the stem with a stride of 2, and three spatiotemporal downsamplings are applied at the first basic block of levels 2, 3 and 4 (stride of 2 also). The number of filters starts at 45 and 64 for the spatial and temporal convolutional layers in the stem, is set to 64 for the first level and is increased by a power of 2 in subsequent levels. After the fourth convolutional level, a global spatiotemporal average pooling is applied, followed by a fully connected layer and the activation function (softmax or sigmoid whether multi-class or multi-label, respectively). These details are summarised in Table 3. One may

Table 2: Detailed parameterization of the proposed R2+1D architecture

Layer	Block	Input size	Channels	Kernel	Stride
Stem	Spatial block	128 x 128 x 32	45	7 x 7 x 1	2 x 2 x 1
	Temporal block	64 x 64 x 32	64	1 x 1 x 3	1 x 1 x 1
Level 1	Basic block 1	64 x 64 x 32	64	3 x 3 x 3	1 x 1 x 1
	Basic block 2	64 x 64 x 32	64	3 x 3 x 3	1 x 1 x 1
Level 2	Basic block 1	64 x 64 x 32	128	3 x 3 x 3	2 x 2 x 2
	Basic block 2	32 x 32 x 16	128	3 x 3 x 3	1 x 1 x 1
Level 3	Basic block 1	32 x 32 x 16	256	3 x 3 x 3	2 x 2 x 2
	Basic block 2	16 x 16 x 8	256	3 x 3 x 3	1 x 1 x 1
Level 4	Basic block 1	16 x 16 x 8	512	3 x 3 x 3	2 x 2 x 2
	Basic block 2	8 x 8 x 4	512	3 x 3 x 3	1 x 1 x 1

notice that, when the downsample exists in the basic blocks, the dimensions of the output of the BN and the output of the previous stage are not compatible. In the levels in which this occurs, a 3D convolution with a 1x1x1 kernel followed by BN is applied to the output of the preceding stage.

Overall, the network has 15,382,033 parameters.

3.4.2 *Data augmentation*

3D networks have the particularity of requiring an even larger amount of data than 2D architectures. Thus, it is useful to artificially generate more training data when possible. Hence the need for the on-the-fly data augmentation step.

Although all US videos have the same beam direction (which always originates from the probe's position), in LUS exams, it produces particular findings (such as A-lines and B-lines) with particular features that are impossible to occur in any other direction than the beam one. Hence, there is a need for extra caution when choosing the augmentations to apply, since some could create unrealistic LUS videos and reduce the clinical significance of the resulting classifier.

Although a few vanilla augmentations were applied to the dataset, to avoid unrealistically looking images, some of them were performed in the polar space instead of the Cartesian one. To be able to perform these transformations, the probe's virtual origin is needed (which in polar coordinates will be the pole), as well as the opening angles for the angular coordinate and the radii (one referring to the small arc and the other to the maximum US depth) of the LUS beam. These values, as previously mentioned, were saved when the sector scan mask was created.

Specifically, the video clip is first converted to the polar space. In this space, a scaling transform, with a factor between 0.7 (Figure 33b) and 1.3 (Figure 33c), is applied over the radial axis, using the pole (represented by the first row of the polar image) as origin. In other words, one varies the axial resolution of the LUS beam (± 30% of the original resolution). A rotation transform, of up to 10° (Figure 33d and 33e), is then applied, being implemented as a translation along the angular axis. Again, the probe's virtual origin is used as a fixed origin, which in the Cartesian space represents the rocking of the probe. Still in the polar space, one applies a contrast augmentation with a gain range of 0.25. Afterwards, the clip is converted back into Cartesian coordinates using the original pole, opening angles and radii, which guarantees that the content is still restricted to the US FOV. Finally, a horizontal flip transform is applied. All these augmentations have a 50% probability of being applied, with the exception of the contrast transform which has a 15% probability. To implement these transformations, the Solt [80] package was used.

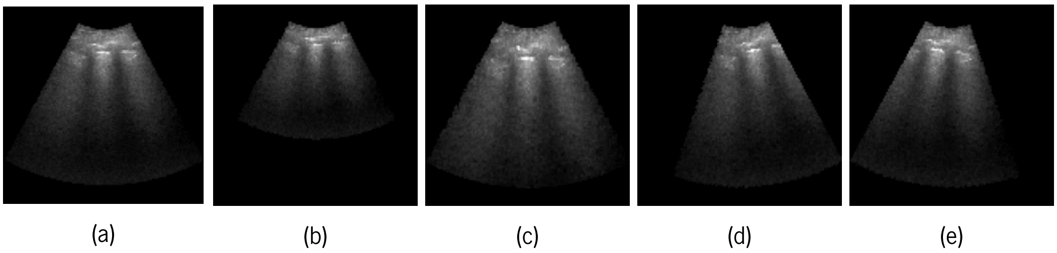


Figure 33: An examples of LUS frame and associated augmented versions.
(a) Original LUS image. (b)-(e) Augmented versions resulting from scaling with a factor of 0.7 or 1.3, or upon rotation by -10° or +10°, respectively.

3.4.3 *Label smoothing regularisation*

To increase the framework's robustness against label noise, one decided to use label smoothing regularisation (LSR). This choice was prompted by the fact that this method is one of the few that can be applied either in multi-class or multi-label problems. Furthermore, it can be jointly used with most variations of the cross entropy loss, which makes it easy to embed in the loss function.

As formerly mentioned, label smoothing alters the true one-hot labels, loosening the confidence of these values. How these values are altered depends on the type of problem. In the categorical case, label smoothing is responsible for assuring that the gap between the biggest and smallest values is decreased. This happens by reducing the value of the target labels, which were 1, and increasing the non-target ones, which had a value of 0 (Equation 3). For binary scenarios, these changes are meant to approximate the true values to 0.5, the threshold value (Equation 4). In the following equations, α represents the label smoothing factor, C is the number of output classes and y_{true} is the one-hot vector.

$$y_{true} = y_{true} \times (1.0 - \alpha) + \alpha / C \tag{3}$$

$$y_{true} = y_{true} \times (1.0 - \alpha) + 0.5 \times \alpha \tag{4}$$

Throughout the work, a label smoothing factor of 0.1 is applied.

3.4.4 *Video-level inference*

Videos have the characteristic of mutability when compared to images. In an image, the findings may or may not be present, while in a video their presence and absence may coexist. Thus, although the dataset annotation was performed on a video level, that does not mean that the labelled finding is present throughout the whole video, which may reduce the algorithm's sensitivity if only one random clip is assessed. The inference process was thus altered with that in mind.

Instead of extracting a single clip from the video and obtaining a single prediction, one divides the LUS video into multiple overlapping clips. As described in Section 3.3, all clips have 32 frames with a frame rate of 8 Hz. Contrarily to the training stage, the first frame of the first clip corresponds to the first frame of the video, and subsequent clips are extracted with a delay of 0.5 seconds. For this dataset, each video results in, approximately, 5 clips. To obtain a video-level prediction, an average of the resulting scores of the multiple clips is computed.

Besides increasing the method's sensitivity when compared to a one-clip prediction, it also increases its robustness when compared to a whole-video prediction (i.e. inputting the full video at the inference stage). Indeed, although the latter would allow the assessment of the full video and therefore allow the detection of any visible finding, by relying on multiple predictions, one decreases the method's uncertainty and ultimately increase its accuracy.

3.5 Semi-supervised setting

In this work, the intention is to benefit from the large number of data available. Since a big portion of the dataset is unlabelled, we focused on semi-supervised techniques.

To this end, one proposes to use the UPS method [72]. This is a pseudo-labelling method that, through a more precise label selection, aims to diminish the amount of noise present in the pseudo-labels. Additionally, UPS offers an approach for the use of pseudo-labelling in multi-label scenarios and performs well with video data (note however that no results were originally shown for a video-oriented multi-label dataset).

The UPS method consists of three steps: pseudo-label generation, pseudo-label selection, and model training. For the generation of pseudo-labels, the hard labels provided are the direct predictions of the network. However, to create more accurate labels and reduce the noise that will be injected into the semi-supervised training, only high-confidence predictions are selected. This is applied in a positive and in a negative manner, meaning that the network can be confident that the output is of a certain class, attributing a positive label, or confident that the output does not belong to that class, giving a negative label. In other words, it is not only the presence of a class that is verified but also its absence. For this, two thresholds are defined, τ_p the positive threshold and τ_n the negative threshold. Besides the confidence thresholds, the uncertainty of the predictions is also integrated into the pseudo-label selection. Rizve *et al.* proved that when labels are selected with more certainty, the calibration error is reduced. As such, two new limits are set, the uncertainty thresholds for positive and negative labels, k_p and k_n, respectively.

Since one intends to use this method in a multi-label scenario, τ_p was set to 0.5 and τ_n to 0.05. For the uncertainty limits, $\text{k}_\text{p} = 0.05$ and $\text{k}_\text{n} = 0.005$ were utilised.

To summarise, this approach, in the multi-label setting, has three types of pseudo labels: positive, negative, and unreliable labels. If the prediction value is higher than τ_p and the uncertainty is lower than $\mathrm{k_p}$, the label is positive. If the prediction has a value and an uncertainty lower than τ_{n} and $\mathrm{k_n}$, respectively, the label is negative. If the prediction values do not fall in any of the above conditions, the label is an uncertain one. These last labels do not contribute to the loss function. To know which labels are to be used for the loss function calculation and which are to be ignored, along with the one-hot vector, a new vector is introduced, g. For each class of each sample, g has either the value of 1 (reliable label) or 0 (unreliable label). LSR was also integrated into the new loss function. The modified binary cross entropy is presented in Equation 5, where c represents the class, C the number of classes, and i the sample. Note that, for labelled samples, all classes in g are set to 1. The true/pseudo label vector, after label smoothing, is represented as y, and the prediction output as $\hat{y}$.

$$L_{BCE} = \frac{1}{\sum_{c=1}^{C} g_c^i} \sum_{c=1}^{C} g_c^i \left[y_c^i \, log\left(\hat{y}_c^i \right) + \left(1 - y_c^i \right) log\left(1 - \hat{y}_c^i \right) \right] \tag{5}$$

Multiple iterations of the UPS method can be applied. First, the framework presented in Section 3.5 is trained on the labelled dataset. Once trained, the predictions for the unlabelled dataset are calculated. Here, in order to apply the UPS method, predictions are made to the original unlabelled clip and to nine additional augmented versions of said clip (applying with 100% probability both scaling and rotation transforms, by up to 7.5% and 2.5° respectively, and employing a horizontal flip transform with 50% probability). This is needed for the uncertainty estimation, which is computed as the standard deviation of the ten predictions. The prediction for the original clip is used for the confidence thresholding. After selecting the unlabelled samples and respective pseudo-labels, these are combined to the labelled dataset and used to re-train the proposed network model from scratch. After each iteration, new pseudo-labels are generated for all unlabelled samples, using the new classifier and the proposed video-level inference technique. The aim is that, at each iteration, more data is included and the pseudo-labels attributed are more curated, injecting a smaller amount of wrongly classified samples in the subsequent network training.

In this work, it is proposed to train the semi-supervised framework for one iteration.

3.6 Ensemble modelling

The integration of domain knowledge in DL approaches has been receiving acclaim. By leveraging the known characteristics of the data, one may adapt the framework and tailor it accordingly, often leading to performance gains. With that in mind, a novel ensemble modelling technique is proposed, based on the hierarchy of LUS findings.

The proposal consists of an ensemble of models trained to predict a different set of output labels, according to the dataset hierarchy present in Figure 25. In this sense, the multiple models' outputs comprise both leaf and high-level labels. All of the models included must comprehend the same label categories. In other words, a model may either have the high-level category as a label or all of the leaf-labels of the said category (for example, a model can either have the "Other pathologies" label, or both 'Consolidation' and "Pleural effusion" labels). It cannot, however, have labels from a data group that is not included in all models.

The intuition is that the classification of the high-level nodes can be improved by considering information from the leaf-labels. The idea of focusing on high-level nodes is that these often have high clinical relevance, namely for patient screening and management. However, high-level labels present a higher intra-class variability, which may increase the classifier's uncertainty and, thus, decrease its performance. Since leaf-labels present a lower intra-class variability, a classifier trained to predict them may focus on more refined features, and present better results for them separately. Hence, models with leaf-labels can be used to improve models with high-level labels, by passing knowledge that, otherwise, the high-level model may fail to capture.

The proposed ensemble is obtained by combining the predictions of leaf- and high-level labels from the multiple models (Figure 34). For each label, an average is computed. If the label is a leaf one (and the corresponding high-level label is not comprised in any model), the average is straightforward, considering the values inferred by all models. If the label is high-level, a mean is computed with the high-level labels from the models that contain them, and the result of an aggregation function for the leaf-labels (that derive from the high-level label under evaluation). This aggregation varies whether the problem is categorical or multi-label. In the categorical scenario, since the network's outputs went through a softmax function, the sum of all labels' values equals 1. Consequently, the probability of a high-level label is the sum of each of its leaf-labels' scores. For multi-label, the probability of a high-level label will be the score of the associated leaf-label with the highest predicted value.

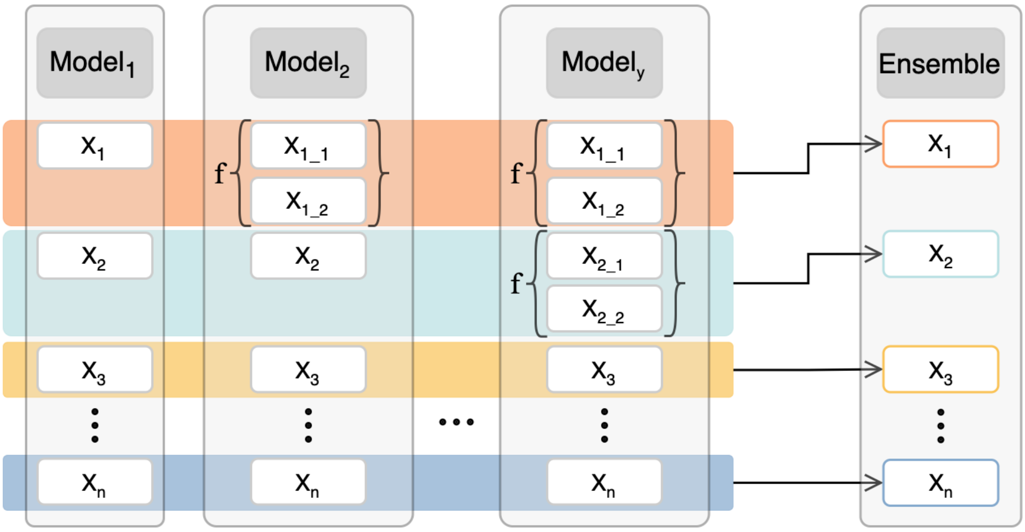


Figure 34: Illustration of the proposed multi-output to high-level ensemble method.
The aggregation function is represented by f (sum/maximum function for categorical/multi-label) and the coloured blocks represent an average operation.

However, if the purpose is to train a classifier able to detect leaf-labels only, a traditional ensemble modelling approach is recommended. Instead of computing an ensemble of multiple models with different output labels, the same model (the one with the intended label set) is trained multiple times and the results are averaged.

Independently of the scenario, the training of each one of the models is done independently, following the aforementioned implementation details.

3.7 Post-processing

With the intent of, once more, leverage from the data hierarchy (Figure 25), a set of ad-hoc rules are proposed to be applied to obtain the video-level predictions when employing a multi-label setting for LUS assessment. With the prediction vectors, instead of choosing all classes with a prediction value above 0.5, the result is modified based on the following rules:

- Only one normal finding can be selected;
- Normal findings cannot coexist with indicative findings;
- Only one of the B-lines leaf-labels can be chosen;
- At least one class has to be chosen in every LUS clip.

Based on these rules, a post-processing pipeline was elaborated. First, the maximum value of the predicted outputs is identified. If it belongs to either ”Scattering” or ”A-lines”, the one with the highest value is selected as the output class (independently of its value being above 0.5 or not). If neither of the aforementioned class has the highest value, the next step is to verify if any of the ’B-lines’ leaf-labels as a value higher than 0.5. If true, only the label with the highest score among those targeting B-lines assessment is selected. This also applies if the high-level ’B positive’ label is used instead of its corresponding leaf-labels. Next, the labels ”Consolidation” and ”Pleural effusion” are verified for the same condition and any with a value higher than 0.5 is selected. The same occurs for the respective high-level prediction. In the case that the highest value belongs to one of the findings indicative of pathologies, but the value itself is not higher than 0.5, the one with the highest score will be the sole attributed label (since all clips must have at least one label).

This post-processing block can only be applied in the multi-label scenario.

4

RESULTS AND DISCUSSION

This chapter is dedicated to the validation of the framework proposed in Chapter 3. First, the metrics utilised to evaluate said framework are presented. Then, two sections describe the experiments performed to evaluate each one of the studied training regimes: the supervised and the semi-supervised approaches. In the first, the experiments are divided into two types of problems: categorical and multi-label classification. In the latter, the research focused on multi-label classifications. In these sections, all results were obtained from a 5-fold cross-validation. Finally, results for the test set, which were obtained using a mean average of the five trained models, are presented and discussed. Unless otherwise indicated, all results include the video-level inference routine proposed in Section 3.4.4.

4.1 Evaluation metrics

To validate the proposed framework and the blocks that compose it, a set of metrics was used: balanced accuracy (BA), average precision (AP), Matthews correlation coefficient (MCC) and F1-score. All of these metrics are presented in all experiments targeting a categorical classification. In the multi-label case, only AP and F1-score are reported. Additionally, when appropriate, the expected calibration error (ECE) was also computed.

Most of the mentioned metrics are a compound of performance metrics, such as precision, recall and specificity. All of these are calculated based on the four possible outcome of the model's output. These can be either: true positive (TP) (positive samples correctly predicted as such), true negative (TN) (negative samples correctly predicted as such), false positive (FP) (negative samples predicted as positive), and false negative (FN) (positive samples predicted as negative). These values are usually presented in the form of a confusion matrix. In multi-class scenarios, these are computed in a one-vs-all manner, meaning that each class is evaluated independently by considering itself as ”positive” and all other as ”negative”.

Precision (Equation 6) is the positive predictive value, which is the ratio of TP among all positive predicted samples (i.e considering TP and FP). Recall (Equation 7), also known as sensitivity, is the fraction of TP among all positives (TP plus TN samples). Similarly, specificity (Equation 8), also termed true negative rate, is the fraction of the TN by all negatives [81].

$$P = \frac{TP}{TP + FP} \tag{6}$$

$$R = \frac{TP}{TP + FN} \tag{7}$$

$$S = \frac{TN}{TN + FP} \tag{8}$$

BA is calculated based on recall and specificity (Equation 9) [81] and is particularly useful to assess the performance of imbalanced classification problems. Derived from the basic accuracy score (a statistical measure of the correct predictions of a classifier among all predictions), it takes into consideration the imbalance ratio between the different classes, giving a better perspective of the network's real performance.

$$BA = \frac{P + R}{2} \tag{9}$$

AP is a multi-threshold metric, obtained from the precision-recall curve (Equation 10) [81]. It provides a value that has into consideration both false positives and false negatives.

$$AP = \sum_{n} (R_n - R_{n-1}) \times Pn \tag{10}$$

F1-score is also obtained from precision and recall (Equation 11) [81], and is considered the harmonic mean of these metrics. It is the selected metric to evaluate the model's performance for each class independently.

$$F1\,score = 2 \times \frac{P \times R}{P + R} \tag{11}$$

MCC performs a correlation between the true and predicted labels (Equation 12) [81]. It can assume values between -1 and +1, being -1 an inverse prediction, 0 an absolutely random prediction and +1 a perfect one.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{12}$$

ECE is a metric to verify a network's calibration [82]. Based on the comparison of the model's accuracy and the model's confidence, it can be used to adjust the model so that the probabilities become more similar to the probabilities of a correct prediction.

4.2 Supervised training

This topic will present the experiments that have led to the methods and associated algorithmic decisions described in Section 3.4. For this setting, two types of problems were considered: categorical and multi-label. All models were trained in the same conditions, with only the last activation function and loss function being altered according to the type of task.

4.2.1 *Influence of network architecture*

To evaluate the network's performance alone, it was decided to train the network with a categorical problem. To this end, samples were categorised as either "Normal findings" or "Indicative findings" (following the hierarchy present in Figure 25). This is the simplest possible problem to create with the present dataset so that factors like noisy labels and imbalanced classes influence the least the decision on the chosen architecture.

To validate the selection of the proposed network (a variant of R2+1D), Table 3 compares its performance against four other: the original R2+1D, R3D, C2+1D and C3D [51]. For all architectures, parameters like regularisation weight and learning rate were optimised to guarantee a fair comparison between networks. These architectures were chosen given their similarities with the proposed model and their ability to deal with 3D inputs. Specifically, both R2+1D and R3D were tested as proposed in Tran *et. al.* [51]. In their version, R2+1D is built in order to have the same number of total parameters as R3D, which is done by adjusting the number of filters in the convolutional layers of the network. For simplification, this operation will be named "boost". In the variant proposed in this work, the number of filters is not adapted (no boost), resulting in a network with approximately half of the number of parameters of R3D. Additionally, to verify the advantage of the residual connections, a comparison against traditional CNNs was also considered. Indeed, C3D and C2+1D differ from R3D and R2+1D,

Table 3: Performance of the proposed architecture and comparison against four other similar architectures

Network	Parameters	BA	MCC	AP	F1-score	
					Normal	Indicative
R2+1D (proposed)	15.4M	**0.9210**	**0.8417**	**0.9704**	**0.9192**	**0.9224**
R2+1D (w/ boost)	33.2M	0.9112	0.8225	0.9687	0.9086	0.9138
R3D	33.2M	0.9147	0.8293	0.9700	0.9126	0.9167
C2+1D	33.0M	0.9023	0.8044	0.9622	0.9003	0.9039
C3D	32.9M	0.9071	0.8138	0.9630	0.9051	0.9087

respectively, only in the fact that they do not have residual connections on the basic block of Figure 32.

By analysing Table 3, one concludes that the chosen network presents the best performance across all metrics. First, it is clear that ResNet variants achieve better results when compared to the traditional CNNs both in 3D or in the factorised version. This aligns with the proven advantages of shortcut connections, avoiding the problem of vanishing and exploding gradients. Another factor for consideration is the number of parameters of each architecture. The proposed R2+1D is the network with fewer parameters and the best performance. This indicates that the other networks were starting to slightly overfit the training data (despite the per-network optimisation of parameters like regularisation weight). This is one of the advantages of the use of factorised blocks compared with 3D ones. Dealing with 3D convolutions automatically increases the number of parameters of the network, while in the factorised version, a smaller number of trainable parameters can be achieved. This creates a network less prone to overfitting, as demonstrated.

4.2.2 *Influence of input pre-processing*

A study of the network's input was conducted to verify if the proposed input settings are optimal. The effect of the video length at a fixed frame rate was evaluated and the reverse also (maintaining the number of seconds and varying the frame rate).

As previously stated, a frame rate of 8 Hz and 4 seconds of duration were the chosen parameters. To have a starting point, the frame rate was decided by observing videos at different rates and choosing the one that appeared to be the minimum needed for an adequate visualisation of the relevant findings. Additionally, in clinical settings, the minimal LUS video length needed for an expert to perform a LUS assessment is 4 seconds. This is a pre-determined value due to the respiratory cycle (inhalation and exhalation). While a LUS video length of less duration may not be enough to visualise all of the present findings, a longer video may capture more than just one cycle. Notwithstanding, both these values were verified, in order to choose the most appropriate ones.

Starting with a fixed frame rate of 8 frames per second, the length of the video was varied, as observed in Table 4. The results were congruent with the initial expectations, with the best result achieved with 4-second clips. Indeed, successive improvements were seen when increasing the clip length from 2 to 4 seconds, but onward the model's performance deteriorates. For videos with less than 4 seconds, as stated, the justification for the reported results is that most of the clips do not comprehend a complete respiratory cycle. Consequently, there may be information (or even entire findings) missing. With more than 4 seconds, besides possibly covering more than one respiratory cycle, the data variability diminishes given the higher similarity between extracted clips (since their duration gets closer to the length of the original video). With less variability in the training set, the network is more prone to overfitting, thus the worst performance observed.

Table 4: Effect of the input's clip duration on the network's performance

Clip duration (seconds)	BA	MCC	AP	F1-score	
				Normal	Indicative
2	0.9065	0.8127	0.9688	0.9049	0.9075
3	0.9147	0.8293	0.9694	0.9123	0.9170
4	**0.9210**	**0.8417**	**0.9704**	**0.9192**	**0.9224**
5	0.9138	0.8279	0.9669	0.9112	0.9167
6	0.9017	0.8034	0.9619	0.8988	0.9047

Regarding frame rate, Table 5 summarises the results obtained for clips with 4 Hz to 12 Hz (and 4 seconds duration). One can observe that frame rates of 4 or 6 frames per second are not sufficient for a good performance. This is expected due to the variability of the findings, as 4 to 6 Hz may often not be sufficient to detect very sudden findings, like an isolated B-line. When increasing the frame rate to a value above 8, the performance also decreases. By increasing the frame rate, while freezing the number of seconds, one is increasing the input data depth. However, since the size of the convolutional kernels are kept constant, the associated receptive field is smaller, which means the temporal features extracted concern smaller temporal intervals. Consequently, there is less temporal correlation and more overfit, which leads to the models delivering worst results. Overall, the proposed use of a frame rate of 8 Hz, reveals to be the optimal one for the present dataset.

4.2.3 *Influence of model output*

To leverage the entire notation of the dataset and the hierarchy inherent to it, a study was conducted to demonstrate the differences in having different types of label aggregations, both for the categorical problem and the multi-label one.

Table 5: Effect of the input's frame rate on the network's performance

Frame rate	BA	MCC	AP	F1-score	
				Normal	Indicative
4	0.9117	0.8232	0.9670	0.9095	0.9137
6	0.9133	0.8266	0.9699	0.9108	0.9158
8	**0.9210**	**0.8417**	0.9704	**0.9192**	**0.9224**
10	0.9143	0.8286	0.9692	0.9119	0.9167
12	0.9144	0.8287	**0.9713**	0.9122	0.9164

Categorical models

For the categorical problem, only labels that are mutually exclusive could be considered. In that sense, besides the aforementioned categorisation, which will be named C1, three other possibilities were considered, as presented in Table 6.

Table 7 presents the aggregated results for each one of the models, with the per-class performance reported in Table 8. Note that, although direct comparison between models cannot be made from the aggregation metrics, these results, together with a per-class result, provide interesting insights. Indeed, the larger performance difference is observed between models C1 and C4, i. e. the models with the fewer and the most number of labels, respectively. However, when analysing both C2 and C3 results, one can notice that, although both have the same number of outputs, C3 presents a better performance. This is due to the difficulty of discerning between "Non-pathological" and "Pathological" findings, opposed to "Scattering" versus "A-lines". Besides being well-represented classes in the dataset, the latter (i.e. the leaf-labels of "Normal finding") are less prone to dubious interpretations. In its turn, B-lines, for example, are present in the two groups ("Non-pathological" and "Pathological"), with different degrees of severity. This can be indicative that there are more noisy labels in the "Indicative finding" data group.

In summary, by adding more categorical outputs, one gains in terms of detail of the information retrieved from the data but has to compromise on the certainty about said information.

Table 6: Categorical models and respective outputs

Categorical models			
C1	**C2**	**C3**	**C4**
Normal finding	Normal finding	Scattering A-lines	Scattering A-lines
Indicative finding	Non-pathological Pathological	Indicative finding	Non-pathological Pathological

Table 7: Comparison of categorical models with different outputs

Model	BA	MCC	AP	F1-score	
				Macro	Micro
C1	0.9210	0.8417	0.9704	0.9208	0.9208
C2	0.7363	0.6935	0.7970	0.7395	0.8128
C3	0.8097	0.7539	0.8603	0.8091	0.8502
C4	0.6974	0.6347	0.7482	0.6979	0.7360

Table 8: Per-class F1-score for the categorical models from Table 6

Model	Normal finding		Indicative finding	
	Scattering	A-lines	Non-pathological	Pathological
C1	0.9192		0.9224	
C2	0.9181		0.4940	0.8065
C3	0.6692	0.8387	0.9195	
C4	0.6647	0.8263	0.5054	0.7953

Multi-label models

For the multi-label scenario, the intention was to obtain the maximum information possible from the model, by creating models from M1 (composed of several leaf-labels and some high-level labels) to M4 (with only leaf-labels). The models and respective output labels are described in Table 9, and the respective results shown in Tables 10 and 11.

Considering Table 10 and Table 11, one observes the same phenomena seen for the categorical models. By increasing the number of outputs, one decreases the general metrics' values, as expected. Moreover, by looking at the per-class scores, it becomes clear that one of the most difficult task lies in the B-lines classification. When comparing M2 to M3, which output the high-level ”B positive” and ”Other pathologies” respectively, M2 delivers a better performance. This aligns with the statement that within the ”B-lines” category, especially ”B positive”, exists the biggest uncertainty and, possibly, label noise. Regarding the normal findings, the performance does not significantly change across the tested models. However, it seems that certain aggregations might slightly jeopardise these classes, which is the case of M3 that displays the lowest values of ”Scattering” and ”A-lines”.

In general, no model consistently outperforms the remaining ones or displays explicit connections and associations between the aggregation of classes and the model's performance. Notwithstanding,

Table 9: Multi-label models and respective outputs

Multi-label models			
M1	**M2**	**M3**	**M4**
Scattering	Scattering	Scattering	Scattering
A-lines	A-lines	A-lines	A-lines
Up to 3 B-lines	Up to 3 B-lines	Up to 3 B-lines	Up to 3 B-lines
B positive	B positive	More than 3 B-lines	More than 3 B-lines
		Coalescent	Coalescent
Other pathologies	Consolidation	Other pathologies	Consolidation
	Pleural effusion		Pleural effusion
Pleural irregularity	Pleural irregularity	Pleural irregularity	Pleural irregularity

Table 10: Comparison of multi-label models with different outputs

Model	AP	F1-score	
		Macro	Micro
M1	0.6615	0.5931	0.6924
M2	0.6356	0.5748	0.6822
M3	0.6360	0.5737	0.6463
M4	0.6196	0.5622	0.6385

Table 11: Per-class F1-score for the multi-label models from Table 9

Model	S	A	< 3 B	B +		OP		PI
				> 3 B	CB	C	PE	
M1	0.6654	0.8305	0.5228	0.7864		0.5886		0.1649
M2	0.6747	0.8358	0.4914	0.7789		0.5767	0.5161	0.1500
M3	0.6487	0.8195	0.5202	0.5644	0.6825	0.6178		0.1630
M4	0.6706	0.8294	0.5204	0.5413	0.6825	0.5446	0.5159	0.1931

S: Scattering; A: A-lines; < 3 B: Up to 3 B-lines; B+: B-line positive; > 3 B: More than 3 B-lines; CB: Coalescent B-lines; OP: Other pathologies; C: Consolidation; PE: Pleural effusion; PI: Pleural irregularity.

one still hypothesises that the leverage of all these models at once (through ensemble modelling) may deliver an overall accuracy improvement, as different models may help in classifying different labels.

4.2.4 *Effect of robust learning*

Given the complexity of interpreting LUS videos, a certain degree of variability is expected between POCUS users. Additionally, the categorisation of some findings is much more dubious than others. With that in mind, one expects the employed label smoothing regularisation to have a more impactful effect on the multi-label models than the categorical ones, where there are fewer outputs and the performance is already satisfactory.

Consequently, it was decided to validate the proposed noisy label reduction approach in the multi-label model M4, i.e. the model with the most labels. A comparison was made between the traditional loss function, binary cross-entropy (BCE), the proposed LSR-based loss function (with a factor of 0.1) and binary focal cross entropy (FCE) (with α =0.25 and γ = 2.0). Results are presented in Table 12.

Overall, the LSR approach presents the best results. Although the focal loss has a higher AP, the difference in the label smoothing AP value is not significant and, in contrast, the difference between both approaches in F1-Score is drastic. Label smoothing not only outperforms the other noise regularisation approach, but also the traditional loss function. On average, each class presents an increase

Table 12: Comparison of noisy labels regularisation strategies for model M4

Loss	AP	F1-score	
		Macro	Micro
BCE	0.6196	0.5622	0.6385
Binary FCE	**0.6210**	0.4858	0.5801
Label smoothing	0.6198	**0.5752**	**0.6505**

in F1-score of 1.3%, which clearly validates the aforementioned intuition about the presence of noisy labels in the annotated dataset, and that the proposed loss function is effective against it.

In addition, Table 13 presents the results of the application of label smoothing for the C1 model. No improvement is observed with LSR and, in fact, the metrics show a slight reduction in performance when compared to the model trained with categorical cross-entropy (CCE). Again, as anticipated, the noisy labels are mostly concentrated in the leaf-labels rather than the high-level ones. This becomes even more evident in the categorical models since they gather the simplest tasks.

4.2.5 *Influence of inference routine*

To corroborate the advantages of the video-level inference explained in Section 3.4.4, Table 14 presents a comparison between the proposed inference routine (average of multiple overlapping video segments) with two others: a whole-video inference and a one-clip inference. For the entire video prediction, a full video is fed to the trained model. For the one-clip prediction, only the first clip of each video is fed to the network. The C1 categorical model was the one chosen to perform this comparison.

From the ablation results, it is proved that the average of multiple overlapping clips from the same video is preferred over classifying the video based on a single prediction (of the entire video or one clip). Indeed, a multi-clip inference introduces variability and reduces the model's uncertainty, obtaining more accurate predictions. However, between single-prediction inference routines, the use of the full video outperforms the one-clip strategy. These results corroborate the intuition that, by considering

Table 13: Effect of label smoothing regularisation on model C1

Loss	BA	MCC	AP	F1-score	
				Normal	Indicative
CCE	**0.9210**	**0.8417**	**0.9704**	**0.9192**	**0.9224**
Label smoothing	0.9177	0.8354	0.9688	0.9154	0.9200

Table 14: Comparison of different inference routines for the C1 model

Inference method	BA	MCC	AP	F1-score	
				Normal	Indicative
One-clip inference	0.9115	0.8227	0.9637	0.9095	0.9130
Whole-video inference	0.9157	0.8313	0.9678	0.9133	0.9180
Multi-clip inference (proposed)	**0.9210**	**0.8417**	**0.9704**	**0.9192**	**0.9224**

the full video, one may identify any finding(s) that appear briefly over the full 6 seconds of the video, which may be missed if one single clip is extracted and analysed.

4.3 Semi-supervised training

Since the categorical models performed well, the semi-supervised approach focused on the multi-label models. These were the models where inputting more data could be of great benefit. Specifically, this was done with the M4 model, as it had the highest number of outputs. Upon a first evaluation, it is then applied to the remaining multi-label models.

When analysing the number of pseudo-labels fed to the network, the "Pleural Irregularity" class presents a challenge. As seen in previous results (Table 11), this is the label with the worst performance upon supervised training. Thus, when attributing pseudo-labels to the unlabelled samples, very few videos were identified as belonging to this class, contributing to a larger class imbalance in the SSL training. For these reasons, it was decided not to include this class in the evaluation of the semi-supervision approach. With that, from these results onward, the label "Pleural irregularity" will be disregarded from the multi-label models.

First, a study was conducted to verify the choice of confidence and uncertainty thresholds. To provide a fair comparison, the results of the confidence thresholds are compared to the original pseudo-label approach. Here, classes with a score above 0.5 are classified as positive (taking the value of 1), or otherwise considered negative (taking the value of 0). To verify if measuring uncertainty for pseudo-label selection was indispensable for a good outcome, an additional selection and training was conducted by removing the uncertainty-based thresholding from the UPS method. Table 15 portrays the obtained results after one iteration of the proposed semi-supervised training. All models were trained in the same conditions, following the method described in Section 3.5.

Interestingly, the classical pseudo-label method (τ_p=0.5 and τ_n = 0.5), in which all LUS unlabelled videos are added (and all labels considered reliable), already improves the model's performance when compared to the supervised model. When considering a lower negative confidence threshold (in an attempt to optimise the selection of negative labels that one can rely on), a slightly lower improvement

Table 15: Comparison of different thresholds for pseudo-label selection

	τ_p	τ_n	k_p	k_n	AP	F1-score	
						Macro	Micro
Supervised	-	-	-	-	**0.6803**	0.6414	0.6775
	0.5	0.5	-	-	0.6530	0.6632	0.6917
	0.5	0.05	-	-	0.6537	0.6605	0.6892
Proposed	0.5	0.05	0.05	0.005	0.6782	**0.6711**	**0.7007**

is observed, albeit almost negligible. Comparing the results from the use of thresholds τ_p=0.5 and τ_n = 0.05 with and without uncertainty thresholding. Addition of the uncertainty estimation allows a more refined pseudo-label selection and leads to a performance improvement, when compared to a selection dependent solely on one prediction. Indeed, a solid model should be able to deliver the same, or similar, results when observing the same clip with weak augmentations. If the results vary substantially, it means the network is not fully confident of the attributed class and, therefore, that label should not be considered reliable. The uncertainty-based selection takes this factor into consideration.

Additionally, one noticed that the choice of thresholds is essential to obtain a good performance (data not shown). If the thresholds are not ideal, for either positive or negative labels, the performance of the network falls short. In this case, more labels with lower confidence are added and the model learning is compromised, suggesting that the quality of the added videos' labels plays a bigger role than the quantity of added videos or the number of reliable labels per added video. These results corroborate the intuition that trustworthy labels are of great importance to obtain an optimal SSL-based performance.

Overall, the employed thresholds, coincidentally the same as proposed in Rizve *et al.* [72], provide the best outcomes. With this selection, an average of 3878 videos, with at least one positive or negative label, are added to each training fold.

Using the aforementioned set of thresholds, Table 16 presents the model's performance for the first 3 iterations of SSL training (with iteration 0 representing the supervised training only). As explained in Section 3.5, the UPS training occurs in iterations with the new pseudo-labels being attributed to the entire unlabelled dataset at each iteration. Besides the model's outputs, Table 16 presents the average number of added videos to each fold's training set, divided in those for which most labels are considered reliable (either positive or negative) and those with a majority of unreliable labels.

A visible improvement occurs from the supervised training to the first iteration of the semi-supervised training (the proposed one). In fact, each of the subsequent iterations also obtains better average F1 scores than the supervised version. However, there is a drop in performance from iteration 1 to iteration 2, and another significant decrease at iteration 3. By analysing the number of videos

Table 16: Number of added videos and model performance at each UPS iteration

Iteration	N° of added clips		AP	F1-score	
	With > 3 labels	With ≤ 3 labels		Macro	Micro
0	-	-	**0.6803**	0.6414	0.6775
1	911	2966	0.6782	**0.6711**	**0.7007**
2	1664	2744	0.6602	0.6650	0.6892
3	1842	2639	0.6256	0.6471	0.6738

added, one observes a significant increase in the number of videos with more than 3 reliable labels, resulting in a larger number of total added videos. However, the results are not better, which is not expected. The expected scenario would be one where at each iteration the performance increases even though the differences become successively smaller. However, this is consistent with the finding made earlier, that the quality of the video labels is more important than the number of labels per video or the total number of added videos. With the addition of 3878 videos in the first iteration, the network starts overfitting, becomes overly confident and less calibrated, which explains why the newly attributed labels, in iterations 2 and 3, are not benefiting the model. This result confirms a known issue in SSL approaches (particularly those from the proxy-label group), the confirmation bias, i.e. incorrect pseudo-labels guide subsequent training iterations and lead to a loop of self-reinforcing errors [69].

To further investigate what was causing this, the network's calibration and ECE were verified and compared to the model obtained by supervised training. The ECE and the reliability plots obtained for each class, for the supervised and first iteration of semi-supervised settings, are shown in Figures 35 to 48.

Reliability plots are a representation of the percentage of correct samples for each range of confidences. For example, considering a confidence of 0.9, in a perfectly calibrated network, it would be expected that 90% of the attributed labels would be correct. Figure 37, for instance, represents the most calibrated class of the supervised model. It has the lowest ECE value and the plot is the closest to what would be ideal.

For each class, the semi-supervised model calibration is further from optimal when compared to the supervised one. This is particularly noticeable for classes "Up to 3 B-lines" and "More than 3 B-lines", which present the highest ECE, and the largest difference between ECEs (supervised vs. semi-supervised). One notices that the conclusions align with the intuition that these are within the hardest labels to classify, and probably the classes for which the pseudo-labels inject further noise in the training.

By analysing all graphics, it is clear that the model is less calibrated in every class. This indicates that, although being able to deliver better results than the supervised version, the network is overly confident. Due to that, when producing the new set of pseudo-labels, a larger number of labels is selected, however, they are not as accurate as in iteration 0. This explains why there is a decline in the model's performance after the first iteration. At each iteration, there is a snowball effect. The network becomes more confident, more uncalibrated and, although being able to (confidently) select more labels, its performance deteriorates. This is also perceivable from the AP value reported in Table 16, which despite the increase in F1-score over the supervised model, shows a noticeable decrease after iterations 2 and 3 of the SSL training. This shows that on average at multiple thresholds, the number of false positives is larger in the semi-supervised setting.

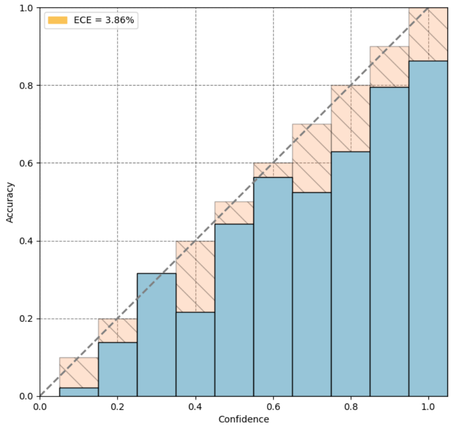


Figure 35: Reliability plot for class "Scattering" for iteration 0.

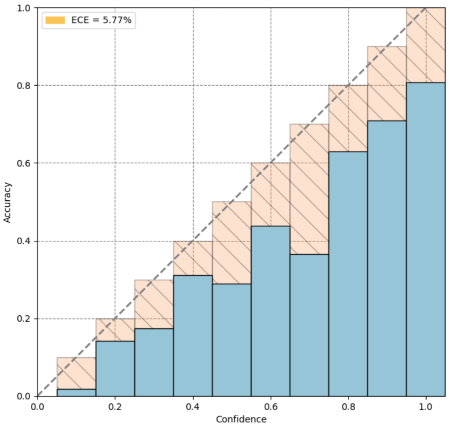


Figure 36: Reliability plot for class "Scattering" for iteration 1.

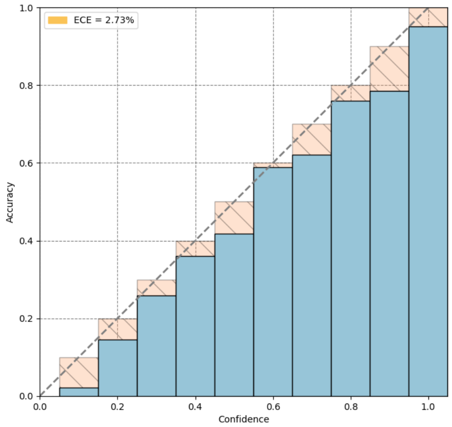


Figure 37: Reliability plot for class "A-lines" for iteration 0.

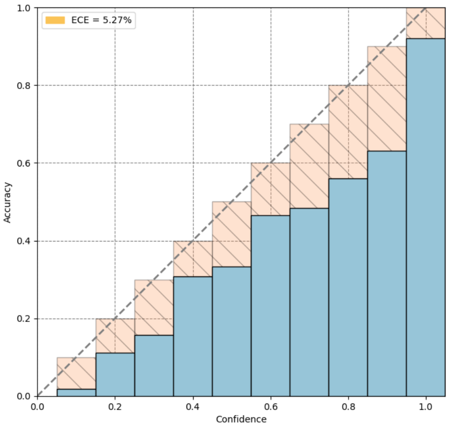


Figure 38: Reliability plot for class "A-lines" for iteration 1.

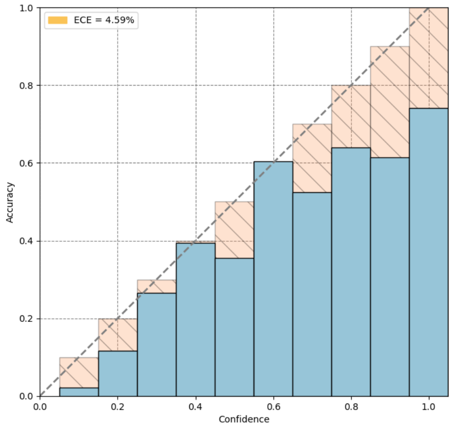


Figure 39: Reliability plot for class "Up to 3 B-lines" for iteration 0.

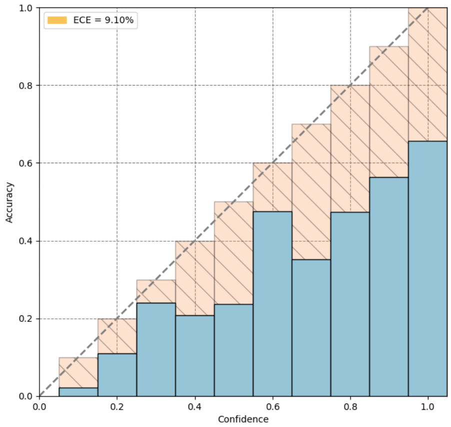


Figure 40: Reliability plot for class "Up to 3 B-lines" for iteration 1.

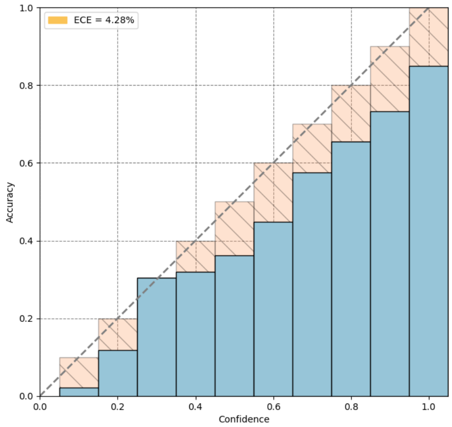


Figure 41: Reliability plot for class "More than 3 B-lines" for iteration 0.

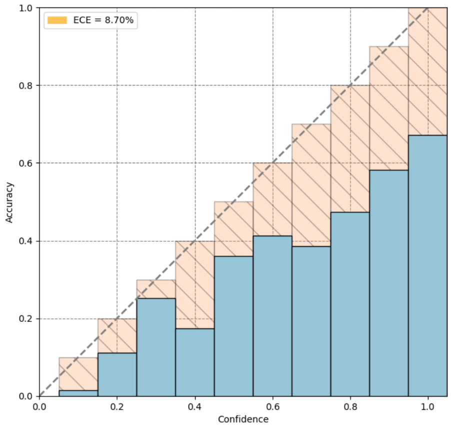


Figure 42: Reliability plot for class "More than 3 B-lines" for iteration 1.

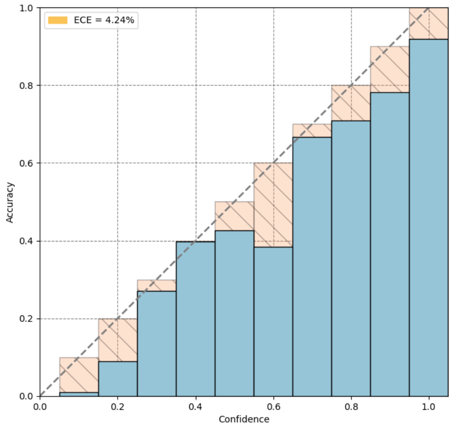


Figure 43: Reliability plot for class "Coalescent B-lines" for iteration 0.

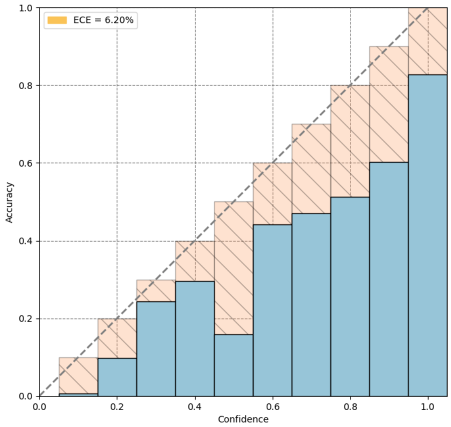


Figure 44: Reliability plot for class "Coalescent B-lines" for iteration 1.

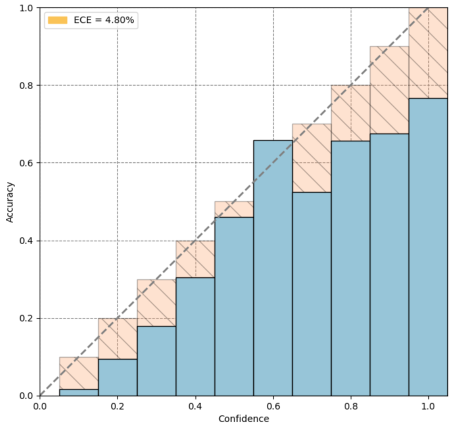


Figure 45: Reliability plot for class "Consolidation" for iteration 0.

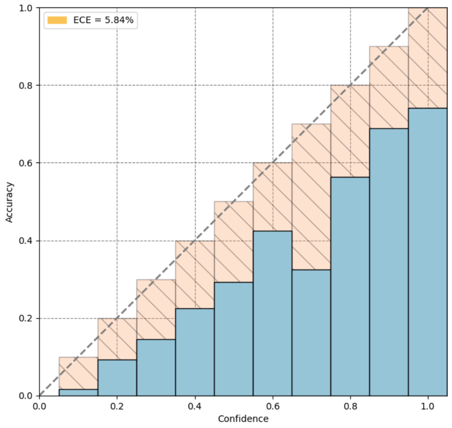


Figure 46: Reliability plot for class "Consolidation" for iteration 1.

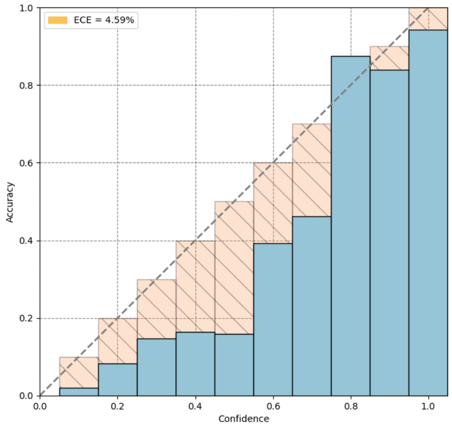


Figure 47: Reliability plot for class "Pleural effusion" for iteration 0.

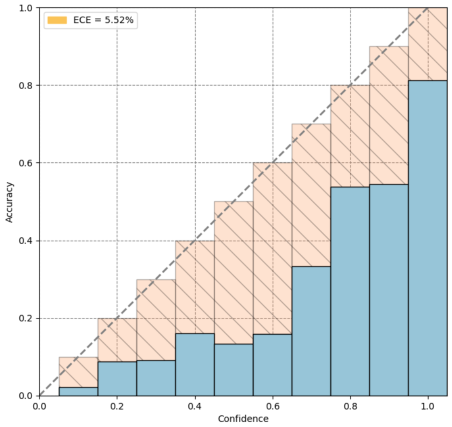


Figure 48: Reliability plot for class "Pleural effusion" for iteration 1.

4.4 Effect of model ensemble

To further improve the performance of our classifier (in either supervised or semi-supervised setting), one has proposed the use of ensemble modelling (Section 3.6). Specifically, one proposes a novel ensemble technique based on models with distinct output label sets when targeting the classification of high-level labels (like seen for model M1), and the use of a traditional model repetition technique when focusing of models with leaf-labels only (like model M4). Here, one compares the proposed strategies against others found in the literature for test-time improvements.

Starting with the high-level labels ensemble, the proposal is to leverage the four multi-label models, M1, M2, M3 and M4, to improve the performance of the M1 model. This will be named multi-output to high-level ensemble. This ensemble is compared with three others: test-time augmentation (repeating the inference for another 9 augmented versions of the input video, following the augmentation scheme described in Section 3.5), model repetition ensemble (by repeating the training of model M1 four times), and leaf to high-level ensemble (a variant of ours in which model M4 is repeated four times and the leaf-labels aggregated to obtain the high-level labels of model M1). These results are presented in Tables 17 and 18. Note that, except for the test-time augmentation, all strategies consider the training of four models

Table 17: Comparison of distinct model ensemble strategies for model M1

Method	Ensemble type	AP	F1-score	
			Macro	Micro
SL	-	0.7408	0.6855	0.7225
	Test-time augmentation	0.7516	0.6890	0.7254
	Model repetition	0.7657	0.6983	0.7346
	Leaf to high-level	0.7711	0.6984	0.7331
	Multi-output to high-level (proposed)	**0.7721**	**0.7057**	**0.7397**
SSL	-	0.7421	0.7146	0.7409
	Test-time augmentation	0.7502	0.7152	0.7408
	Model repetition	0.7595	0.7221	0.7487
	Leaf to high-level	0.7577	**0.7275**	**0.7542**
	Multi-output to high-level (proposed)	**0.7697**	0.7264	0.7515

Table 18: Per-class F1-scores obtained for the ensemble strategies in Table 17

Method	Ensemble type	S	A	< 3 B	B +	OP
SL	-	0.6632	0.8357	0.4978	0.7940	0.6369
	Test-time augmentation	0.6760	0.8310	0.5017	0.8007	0.6355
	Model repetition	0.6910	0.8411	0.5133	**0.8007**	0.6455
	Leaf to high-level	**0.6937**	**0.8482**	0.5332	0.7767	0.6403
	Multi-output to high-level (proposed)	0.6842	0.8420	**0.5525**	0.7980	**0.6520**
SSL	-	0.6832	0.8392	0.5886	0.8065	0.6554
	Test-time augmentation	0.6861	0.8362	0.5885	0.8097	0.6554
	Model repetition	0.6876	0.8509	0.5983	0.8107	0.6631
	Leaf to high-level	**0.7041**	**0.8497**	0.5929	0.8143	**0.6763**
	Multi-output to high-level (proposed)	0.6966	0.8446	**0.6005**	**0.8145**	0.6758

S: Scattering; A: A-lines; < 3 B: Up to 3 B-lines; B+: B positive; OP: Other pathologies.

In agreement with the intuition that models with a higher number of leaf-labels are more refined, it appears that ensembles that leverage from leaf-labels to provide the corresponding high-level labels outperform models relying solely on those high-level labels. In opposition, test-time augmentation delivers the worst outcome, despite still surpassing a single inference at test-time. This is corroborated for both supervised and semi-supervised settings.

Comparing both leaf and multi-output to high-level ensembles, the latter outperforms the first in the supervised setting, with the difference between the average F1-scores being minimal in favour of the first in the semi-supervised one. However, the AP score is greater in the multi-output ensemble. This leads to the conclusion that, while leveraging from one model of just leaf-labels to obtain high-level labels is advantageous, relying on models able to classify distinct output label sets, seems to be more promising. A possible reasoning for such result is the fact that models with distinct label sets focus on distinct data distributions, differing in the existing intra- and inter- class variability, but also in things like class imbalance and label noise.

For the leaf-labels model ensemble, the proposed approach is to use the model repetition ensemble described above, with four repetitions. Once again, one compared this technique against test-time augmentation to validate its effectiveness, with the results summarised in Tables 19 and 20. Both test-time inference techniques outperformed a single inference routine, with the model repetition ensemble providing a better performance, both globally and for nearly all classes individually. This is an expected outcome since, being M4 the most refined model, it is expected that the repetition of that same model multiple times outperforms the model when alone.

Finally, for the categorical models, since the semi-supervised approach was not applied, the proposed ensembles were evaluated for the supervised setting only. Again, the multiple outputs to high-level ensemble is applied to C1 (Table 21) and the model repetition ensemble to C4 (Table 22). As expected, although with a smaller improvement (mainly due to the fact that the models were already very robust), the results do increase with the proposed ensemble implementations.

Table 19: Comparison of distinct model ensemble strategies for model M4

Method	Ensemble type	AP	F1-score	
			Macro	Micro
SL	-	0.6803	0.6414	0.6775
	Test-time augmentation	0.6849	0.6411	0.6783
	Model repetition (proposed)	**0.7093**	**0.6503**	**0.6898**
SSL	-	0.6782	0.6711	0.7007
	Test-time augmentation	0.6902	0.6766	0.7036
	Model repetition (proposed)	**0.6980**	**0.6868**	**0.7118**

Table 20: Per-class F1-scores obtained for the ensemble strategies in Table 19

Method	Ensemble type	S	A	< 3 B	> 3 B	CB	C	PE
SL	-	0.6667	0.8346	**0.5356**	0.5773	0.7192	0.5733	**0.5831**
	Test-time augmentation	0.6828	0.8329	0.5190	**0.5863**	0.7275	0.5577	0.5816
	Model repetition (proposed)	**0.6937**	**0.8482**	0.5332	0.5858	**0.7373**	**0.5808**	0.5731
SSL	-	0.6961	0.8433	**0.5938**	0.6258	0.7478	0.6026	0.5882
	Test-time augmentation	0.6981	0.8445	0.5888	0.6307	0.7500	0.6136	0.6103
	Model repetition (proposed)	**0.7041**	**0.8497**	0.5929	**0.6334**	**0.7566**	**0.6381**	**0.6330**

S: Scattering; A: A-lines; < 3 B: Up to 3 B-lines; > 3 B: More than 3 B-lines; CB: Coalescent B-lines; C: Consolidation; PE: Pleural effusion.

Table 21: Original C1 model and proposed ensemble

Method	Ensemble type	BA	MCC	AP	F1-score	
					Normal	Indicative
SL	-	0.9210	0.8417	0.9704	0.9192	0.9224
	Multi-output to high-level	**0.9268**	**0.8533**	**0.9758**	**0.9252**	**0.9280**

Table 22: Original C4 model and proposed ensemble

Method	Ensemble type	BA	MCC	AP	F1-score	
					Macro	Micro
SL	-	0.6974	0.6347	0.7482	0.6979	0.7360
	Model repetition	**0.7046**	**0.6498**	**0.7723**	**0.7055**	**0.7472**

4.5 Influence of post-processing

At last, the influence of the implemented post-processing routine was also evaluated. To do so, it was gradually added in both supervised and semi-supervised settings, with and without the proposed ensemble for either M1 and M4 (Tables 23 and 24, respectively). It is important to bear in mind that this algorithm is only applicable in multi-label scenarios.

Table 23: Comparison of different combinations of blocks for model M1

Method	Ensemble	Post-processing	F1-score	
			Macro	Micro
SL			0.6855	0.7225
		x	0.6953	0.7285
	x		0.7057	0.7397
	x	x	**0.7138**	**0.7441**
SSL			0.7146	0.7409
		x	0.7073	0.7366
	x		**0.7264**	**0.7515**
	x	x	0.7212	0.7493

The results for both models are similar. In the supervised setting, the application of the post-processing block improves performance, irrespective of the usage or not of ensemble modelling. The post-processing block, as described in Section 3.7, relies on the dataset hierarchy, and is thus expected to improve the models' performance since it restricts the possible outputs to viable ones. This is not verified in the SSL approach, however. Actually, in this case, the values obtained with the post-processing block are inferior for both M1 and M4 models. One explanation for this incident is, once again, the calibration of the semi-supervised model not being ideal. The post-processing is based on the output vectors. If the network is overly confident, and therefore outputting very high or very low values, the confidence-based rules become a fault. Labels that, in the supervised version, were doubtful, in the semi-supervised become certain. And, while in the usual multi-label classification, every class with a confidence above 0.5 is selected, in this post-processing routine that does not happen. Considering, for example, a model that had a score above 0.5 for two (incompatible)

Table 24: Comparison of different combinations of blocks for model M4

Method	Ensemble	Post-processing	F1-score	
			Macro	Micro
SL			0.6414	0.6775
		x	0.6481	0.6827
	x		0.6503	0.6898
	x	x	**0.6631**	**0.7003**
SSL			0.6711	0.7007
		x	0.6611	0.6918
	x		**0.6868**	**0.7118**
	x	x	0.6755	0.7030

classes, in the vanilla classification both are selected and one of them is probably correct (unless a third incompatible class exists). However, in this scenario, the post-processing ad-hoc rules will force the choice of one. This may become more frequent in the semi-supervised model since it is overly confident.

The applicability of the domain knowledge is backed for the supervised setting. However, its objective is lost in the semi-supervised approach. Note that different results may be achieved if one calibrates the SSL-based models (following, for example, the strategies in [83]), but such experiment is left for future studies.

4.6 Test set results

Considering the multitude of experiments performed and their respective results, a final assessment of the proposed strategy (summarily illustrated in Figure 23) was performed on the held-out test set. The test results are presented for both categorical C1 and C4 models, and multi-label M1 and M4 models, for both supervised and semi-supervised settings, each compared with the corresponding baselines (Tables 25 to 28). Each baseline considers the model in question without any of the proposed strategies,with the exception of the proposed video-level inference. Additionally, for the multi-label scenarios, the considered baseline does not include the ”Pleural irregularity” class to permit the direct comparison against the results upon semi-supervised training. For the multi-label approaches, two proposals are investigated, a supervised and a semi-supervised one, while one single proposal is evaluated for the categorical classification (supervised setting only). For the multi-label models, M1 and M4, both proposals (supervised and semi-supervised) contain label smoothing regularisation and either multi-output to high-level ensemble or model repetition ensemble, respectively. The difference lies in the post-processing routine which is only applied in the supervised setting. For the categorical tasks, C1 and C4, one considers multi-output to high-level and model repetition ensembles, respectively. Once again, note that the proposed post-processing routine is not applicable for categorical tasks.

By observing Tables 25 to 28, one can note that the results decrease slightly from those reported for the validation set, which may underline a bit of overfitting due to hyperparameters’ tuning and empirical algorithmic choices. Note that this slight drop in performance may also be caused by other factors like distinct class imbalance ratios (jeopardising the performance measured for certain low-frequency classes) or higher label noise on the test set. Another reason may be linked to the quality of the test set videos, or the fact that there are fewer videos considered in this set, increasing the representativeness of each video on the final results. Among the evaluated classes, the ”Up to 3 B-lines” presents the largest decrease. This may be due to the fact that this class presents the highest amount of label noise, but also because its interpretation is difficult and hampered by the fact that a similar artifact, the so-called Z-lines (in every aspect similar but for the fact that they do not occur until the bottom of the sector scan), if present should be disregarded.

Notwithstanding, the conclusions remain the same, with all proposals surpassing the baseline models and, when applicable, corroborating the interest in employing the semi-supervised methods.

Table 25: Test set performance for the baseline and final proposal for the C1 model

Model	BA	MCC	F1-score			
			Macro	Micro	Normal	Indicative
Baseline	0.9121	0.8238	0.9118	0.9131	0.9012	0.9225
Proposal	**0.9261**	**0.8522**	**0.9261**	**0.9272**	**0.9170**	**0.9351**

Table 26: Test set performance for the baseline and final proposal for the C4 model

Model	BA	MCC	F1-score					
			Macro	Micro	S	A	NP	P
Baseline	0.6750	0.6507	0.6768	0.7614	0.6423	0.8077	0.3911	0.8662
Proposal	**0.6900**	**0.6657**	**0.6919**	**0.7716**	**0.6561**	**0.8121**	**0.4264**	**0.8729**

S: Scattering; A: A-lines; NP: Non-pathological; P: Pathological

Table 27: F1-scores obtained in the test set by the baseline and final proposals for the M1 model

Model	Macro	Micro	S	A	< 3 B	B +	OP
Baseline	0.6664	0.7243	0.6652	0.8192	0.4793	0.8356	0.5327
SL Proposal	0.6826	0.7403	0.6841	0.8313	0.4952	0.8486	0.5540
SSL Proposal	**0.7045**	**0.7540**	**0.6925**	**0.8475**	**0.5529**	**0.8572**	**0.5725**

S: Scattering; A: A-lines; < 3 B: Up to 3 B-lines; B +: B positive; OP: Other pathologies.

Table 28: F1-scores obtained in the test set by the baseline and final proposals for the M4 model

Model	Macro	Micro	S	A	< 3 B	> 3 B	CB	C	PE
Baseline	0.6095	0.6628	0.6559	0.8124	0.4758	0.6017	0.7310	0.5889	0.4006
SL Proposal	0.6406	0.6879	**0.6838**	0.8291	0.5287	0.6273	0.7495	**0.6155**	0.4502
SSL Proposal	**0.6509**	**0.6957**	0.6702	**0.8433**	**0.5414**	**0.6530**	**0.7567**	0.6139	**0.4780**

S: Scattering; A: A-lines; < 3 B: Up to 3 B-lines; > 3 B: More than 3 B-lines; CB: Coalescent B-lines; C: Consolidation; PE: Pleural effusion.

Comparing the categorical and the multi-label models, the latter (being more challenging tasks) are the models that benefit the most from the proposed blocks, showing the highest improvement. Furthermore, besides the calibration problem encountered with the semi-supervised approach, its utilisation is still an asset as it is able to produce better outcomes than the supervised setting. The final semi-supervised proposals deliver an average improvement in F1-score of 3.81% and 4.14% for models M1 and M4, respectively.

To demonstrate the potential utility of the proposed LUS interpretation method in clinical practice, one used the model C1 to create a visual representation of the classification result over the multiple fields acquired in a LUS exam. This representation shows the lungs divided according to the imaged fields and coloured (per field) following the predicted class. Figure 49 presents the obtained representation for a representative patient from the test set, considering the categorical classification between normal and indicative findings.

As one may notice, very similar representations were obtained, with the proposed algorithm wrongly classifying only two of the twelve fields. The RL - LI field was mistaken as having an underlying pathology, and the LL - AI field was misclassified as healthy. In the case of the latter, the true label is ”Up to 3 B-lines” (hence the ”Indicative finding” label), which is one of the hardest and most ambiguous findings to interpret. Overall, the representation is very similar and the global outcome regarding each lung remains the same.

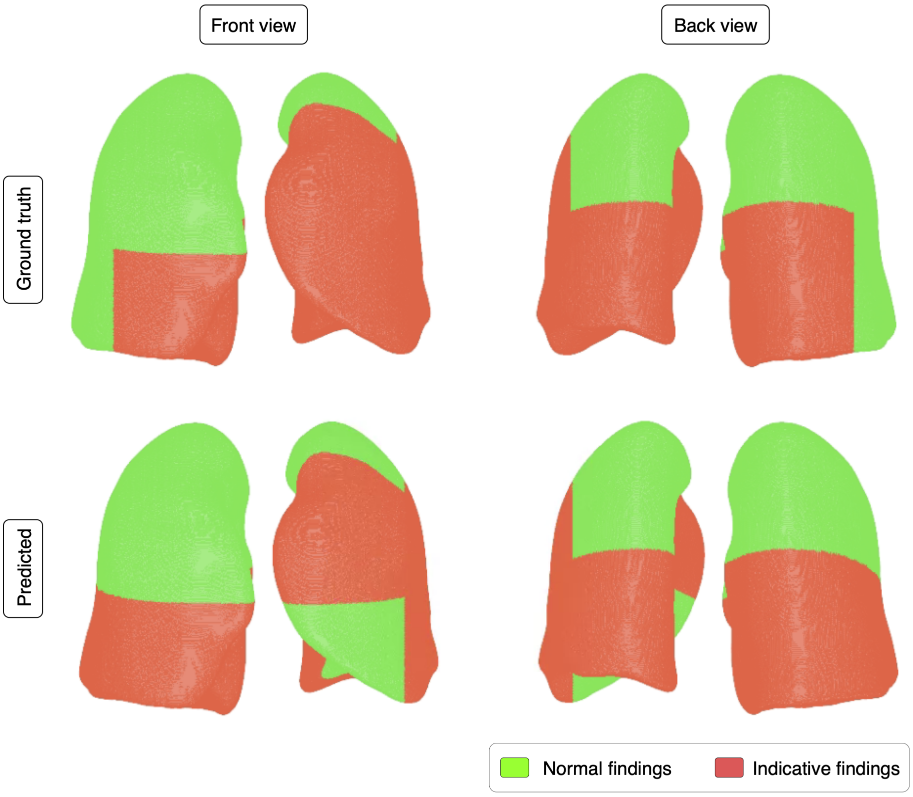


Figure 49: Visual representation of the lungs of a patient from the test set, coloured per field according to the (top) ground truth and (bottom) predicted labels.

5

CONCLUSION AND FUTURE WORK

In this work, it was proposed to develop a generic deep learning framework for automatic interpretation of lung POCUS videos. Thus, the main objective was to develop a network that, upon receiving a LUS video, identifies the finding(s) present in that video.

A framework based on a 3D ResNet was created that incorporates a noise-robust approach (in the form of label smoothing regularisation) and integrates domain knowledge (in the form of model ensembling, and optionally, a post-processing routine). The implemented framework is also flexible, being applicable for either categorical or multi-label classification, and can be successfully transferred to a semi-supervised setting (through an uncertainty-aware pseudo-label selection method) to leverage of unlabelled data, when available.

The framework's performance was validated with several models, both categorical and multi-label. For each classification, two different outputs are proposed, depending on the information one desires to retrieve from the LUS videos. By combining the different blocks proposed and the different classifications types, one can create models that are able to respond to different tasks and objectives. For example, the binary categorical model, which has excellent performance in distinguishing between normal-looking lungs and lungs with an indication of underlying pathology, can be of great use for patient screening in emergency scenarios. Triage could be performed based on the algorithm's output, giving experts a rapid forewarning on urgent cases. On the other hand, the more complex multi-label models could be used to get a sense of the progression and extension of the pathology.

These methods, however, are not without flaws. First, in order to improve the performance of the semi-supervised method, calibration strategies (such as temperature scaling and isotonic regression [83]) would need to be investigated and implemented to solve the observed limitations. Moreover, the presented work focused exclusively on lung parenchymal fields. For an increased applicability in clinical practice, the work would need to be extended to the diaphragm insertion and lung sliding fields. In addition, to simplify the adoption of the framework in clinics, experts must trust it. This can be achieved by resorting to explainability methods, allowing the expert to understand the reasons for the network's output (e.g. by identifying where it is looking at, either spatially and/or temporally). These would all be indispensable next steps to enhance the proposal.

BIBLIOGRAPHY

[1] S. Mader, *Understanding Human Anatomy Physiology*. McGraw-Hill Science Engineering, 2004.

[2] G. Tortora, *Principles of anatomy and physiology*. Hoboken, NJ: John Wiley & Sons, 2010.

[3] K. Graaff, *Human anatomy*. Boston: McGraw-Hill, 2002.

[4] *The global impact of respiratory disease*. Sheffield: European Respiratory Society, on behalf of the Forum of International Respiratory Societies, 2017.

[5] "Lung diseases." https://www.niehs.nih.gov/health/topics/conditions/lung-disease/index.cfm. (Accessed on 23 dec. 2021).

[6] "Principles of respiratory investigation - ers." https://www.erswhitebook.org/chapters/principles-of-respiratory-investigation/. (Accessed on 23 dec. 2021).

[7] L. E. Vanfleteren and D.-J. Slebos, "Emerging techniques in the world of respiratory imaging," *Respiration*, vol. 99, pp. 97–98, Dec. 2019.

[8] L. Gargani and G. Volpicelli, "How i do it: Lung ultrasound," *Cardiovascular Ultrasound*, vol. 12, July 2014.

[9] M. Peck and P. MacNaughton, eds., *Focused Intensive Care Ultrasound*. Oxford University Press, Mar. 2019.

[10] F. M. Abu-Zidan, A. F. Hefny, and P. Corr, "Clinical ultrasound physics," *J. Emerg. Trauma Shock*, vol. 4, pp. 501–503, Oct. 2011.

[11] N. Tole, *Basic physics of ultrasonographic imaging*. Geneva: World Health Organization, 2005.

[12] P. Suetens, "Ultrasound imaging," in *Fundamentals of Medical Imaging*, pp. 147–183, Cambridge University Press.

[13] Jinlei, *Ultrasound fundamentals an evidence-based guide for medical practitioners*. Cham: Springer, 2021.

[14] A. Ng and J. Swanevelder, "Resolution in ultrasound imaging," *Continuing Education in Anaesthesia Critical Care & Pain*, vol. 11, pp. 186–192, Oct. 2011.

[15] M. K. Feldman, S. Katyal, and M. S. Blackwood, "US artifacts," *RadioGraphics*, vol. 29, pp. 1179–1189, July 2009.

[16] D. A. Lichtenstein, "BLUE-protocol and FALLS-protocol," *Chest*, vol. 147, pp. 1659–1670, June 2015.

[17] D. A. Lichtenstein, "Lung ultrasound in the critically ill," *Annals of Intensive Care*, vol. 4, no. 1, p. 1, 2014.

[18] P. H. Mayo, R. Copetti, D. Feller-Kopman, G. Mathis, E. Maury, S. Mongodi, F. Mojoli, G. Volpicelli, and M. Zanobetti, "Thoracic ultrasonography: a narrative review," *Intensive Care Medicine*, vol. 45, pp. 1200–1211, Aug. 2019.

[19] D. S. Brenner, G. Y. Liu, R. Omron, O. Tang, B. T. Garibaldi, and T. C. Fong, "Diagnostic accuracy of lung ultrasound for SARS-CoV-2: a retrospective cohort study," *The Ultrasound Journal*, vol. 13, Mar. 2021.

[20] J. Born, G. Brändle, M. Cossio, M. Disdier, J. Goulet, J. Roulin, and N. Wiedemann, "Pocovid-net: Automatic detection of covid-19 from a new lung ultrasound imaging dataset (pocus)," 2021.

[21] R. J. G. van Sloun and L. Demi, "Localizing b-lines in lung ultrasonography by weakly supervised deep learning, in-vivo results," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, pp. 957–964, Apr. 2020.

[22] R. Arntfield, D. Wu, J. Tschirhart, B. VanBerlo, A. Ford, J. Ho, J. McCauley, B. Wu, J. Deglint, R. Chaudhary, C. Dave, B. VanBerlo, J. Basmaji, and S. Millington, "Automation of lung ultrasound interpretation via deep learning for the classification of normal versus abnormal lung parenchyma: A multicenter study," *Diagnostics*, vol. 11, p. 2049, Nov. 2021.

[23] S. E. Ebadi, D. Krishnaswamy, S. E. S. Bolouri, D. Zonoobi, R. Greiner, N. Meuser-Herr, J. L. Jaremko, J. Kapur, M. Noga, and K. Punithakumar, "Automated detection of pneumonia in lung ultrasound using deep video classification for COVID-19," *Informatics in Medicine Unlocked*, vol. 25, p. 100687, 2021.

[24] J. Born, N. Wiedemann, M. Cossio, C. Buhre, G. Brändle, K. Leidermann, A. Aujayeb, M. Moor, B. Rieck, and K. Borgwardt, "Accelerating detection of lung pathologies with explainable ultrasound image analysis," *Applied Sciences*, vol. 11, p. 672, Jan. 2021.

[25] S. Roy, W. Menapace, S. Oei, B. Luijten, E. Fini, C. Saltori, I. Huijben, N. Chennakeshava, F. Mento, A. Sentelli, E. Peschiera, R. Trevisan, G. Maschietto, E. Torri, R. Inchingolo, A. Smargiassi, G. Soldati, P. Rota, A. Passerini, R. J. G. van Sloun, E. Ricci, and L. Demi, "Deep learning

for classification and localization of COVID-19 markers in point-of-care lung ultrasound," *IEEE Transactions on Medical Imaging*, vol. 39, pp. 2676–2687, Aug. 2020.

[26] C.-H. Tsai, J. van der Burgt, D. Vukovic, N. Kaur, L. Demi, D. Canty, A. Wang, A. Royse, C. Royse, K. Haji, J. Dowling, G. Chetty, and D. Fontanarosa, "Automatic deep learning-based pleural effusion classification in lung ultrasound images for respiratory pathology diagnosis," *Physica Medica*, vol. 83, pp. 38–45, Mar. 2021.

[27] O. Frank, N. Schipper, M. Vaturi, G. Soldati, A. Smargiassi, R. Inchingolo, E. Torri, T. Perrone, F. Mento, L. Demi, M. Galun, Y. C. Eldar, and S. Bagon, "Integrating domain knowledge into deep networks for lung ultrasound with applications to COVID-19," *IEEE Transactions on Medical Imaging*, pp. 1–1, 2021.

[28] C. Baloescu, G. Toporek, S. Kim, K. McNamara, R. Liu, M. M. Shaw, R. L. McNamara, B. I. Raju, and C. L. Moore, "Automated lung ultrasound b-line assessment using a deep learning algorithm," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 67, pp. 2312–2320, Nov. 2020.

[29] X.-D. Zhang, "Machine learning," in *A Matrix Algebra Approach to Artificial Intelligence*, pp. 223–440, Springer Singapore, 2020.

[30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[31] K.-L. Du and M. N. S. Swamy, "Fundamentals of machine learning," in *Neural Networks and Statistical Learning*, pp. 15–65, Springer London, Dec. 2013.

[32] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O'Reilly Media, Inc, 2019.

[33] Y.-T. Cheng, "Supervised, semi-supervised, unsupervised and self-supervised learning," November 2021.

[34] M. W. Berry, A. Mohamed, and B. W. Yap, eds., *Supervised and Unsupervised Learning for Data Science*. Springer International Publishing, 2020.

[35] S. Sharma, S. Sharma, and A. Athaiya, "Activation function in neural networks," *International Journal of Engineering Applied Sciences and Technology*, vol. 4, pp. 310–316, 2020.

[36] A. Krogh, "What are artificial neural networks?," *Nature Biotechnology*, vol. 26, pp. 195–197, Feb. 2008.

[37] "Neural networks." https://medium.com/nerd-for-tech/neural-networks-68531432fb5. (Accessed on 9 jan. 2022).

[38] C. D. of Rare, M. Collections, M. L. |, . September 25, C. W. Commons, and C. I. Awards/-Provided, “Professor’s perceptron paved the way for ai – 60 years too soon,” Sep 2019.

[39] A. Dertat, “Applied deep learning - part 1: Artificial neural networks,” Oct 2017.

[40] J. Lederer, “Activation functions in artificial neural networks: A systematic overview,” 2021.

[41] B. Mehlig, *Machine Learning with Neural Networks*. Cambridge University Press, Oct. 2021.

[42] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.

[43] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, no. 61, pp. 2121–2159, 2011.

[44] M. D. Zeiler, “Adadelta: An adaptive learning rate method,” 2012.

[45] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, IEEE, Aug. 2017.

[46] “A comprehensive guide to convolutional neural networks — the eli5 way | by sumit saha | towards data science.” https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. (Accessed on 9 jan. 2022).

[47] “Cs 230 - convolutional neural networks cheatsheet.” https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks. (Accessed on 9 jan. 2022).

[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[49] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2015.

[50] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.

[51] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, June 2018.

[52] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” 2014.

[53] Z. Zhou, V. Sodha, J. Pang, M. B. Gotway, and J. Liang, "Models genesis," *Medical Image Analysis*, vol. 67, p. 101840, Jan. 2021.

[54] B. Barros, P. Lacerda, C. Albuquerque, and A. Conci, "Pulmonary COVID-19: Learning spatiotemporal features combining CNN and LSTM networks for lung ultrasound video classification," *Sensors*, vol. 21, p. 5486, Aug. 2021.

[55] "Problems in machine learning models? check your data first | by dhruv sharma | towards data science." https://towardsdatascience.com/problems-in-machine-learning-models-check-your-data-first-f6c2c88c5ec2. (Accessed on 5 sept. 2022).

[56] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–19, 2022.

[57] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," 2018.

[58] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," 2019.

[59] X. Ma, H. Huang, Y. Wang, S. Romano, S. Erfani, and J. Bailey, "Normalized loss functions for deep learning with noisy labels," 2020.

[60] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?," 2019.

[61] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, Oct. 2018.

[62] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, Mar. 2019.

[63] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2017.

[64] O. Chapelle, B. Scholkopf, and A. Zien, eds., *Semi-Supervised Learning*. The MIT Press, Sept. 2006.

[65] Y. Ouali, C. Hudelot, and M. Tami, "An overview of deep semi-supervised learning," 2020.

[66] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk, and I. J. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," 2018.

[67] J. E. van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” *Machine Learning*, vol. 109, pp. 373–440, Nov. 2019.

[68] H. Pham, Z. Dai, Q. Xie, M.-T. Luong, and Q. V. Le, “Meta pseudo labels,” 2020.

[69] E. Arazo, D. Ortego, P. Albert, N. E. O’Connor, and K. McGuinness, “Pseudo-labeling and confirmation bias in deep semi-supervised learning,” in *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, July 2020.

[70] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, “Mixmatch: A holistic approach to semi-supervised learning,” 2019.

[71] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” 2020.

[72] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah, “In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning,” 2021.

[73] “Github - uotw/clipdeidentifier: removes protected health information from ultrasound media.” https://github.com/uotw/ClipDeidentifier.

[74] “Authoring environment.” https://jacinto.harena.org/author/home/.

[75] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014.

[76] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” 2015.

[77] I. Loshchilov and F. Hutter, “Sgdr: Stochastic gradient descent with warm restarts,” 2016.

[78] F. Chollet *et al.*, “Keras,” 2015.

[79] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. Software available from tensorflow.org.

[80] A. Tiulpin, “Solt: Streaming over lightweight transformations,” July 2019.

[81] L. Maier-Hein, A. Reinke, P. Godau, M. D. Tizabi, E. Christodoulou, B. Glocker, F. Isensee, J. Kleesiek, M. Kozubek, M. Reyes, M. A. Riegler, M. Wiesenfarth, M. Baumgartner, M. Eisenmann, D. Heckmann-Nötzel, A. E. Kavur, T. Rädsch, L. Acion, M. Antonelli, T. Arbel, S. Bakas, P. Bankhead, A. Benis, M. J. Cardoso, V. Cheplygina, B. Cimini, G. S. Collins, K. Farahani, L. Ferrer, A. Galdran, B. van Ginneken, R. Haase, D. A. Hashimoto, M. M. Hoffman, M. Huisman, P. Jannin, C. E. Kahn, D. Kainmueller, B. Kainz, A. Karargyris, A. Karthikesalingam, H. Kenngott, F. Kofler, A. Kopp-Schneider, A. Kreshuk, T. Kurc, B. A. Landman, G. Litjens, A. Madani, K. Maier-Hein, A. L. Martel, P. Mattson, E. Meijering, B. Menze, D. Moher, K. G. M. Moons, H. Müller, B. Nichyporuk, F. Nickel, J. Petersen, N. Rajpoot, N. Rieke, J. Saez-Rodriguez, C. S. Gutiérrez, S. Shetty, M. van Smeden, C. H. Sudre, R. M. Summers, A. A. Taha, S. A. Tsaftaris, B. Van Calster, G. Varoquaux, and P. F. Jäger, "Metrics reloaded: Pitfalls and recommendations for image analysis validation," 2022.

[82] "Neural network calibration using pytorch | by lukas huber | towards data science." https://towardsdatascience.com/neural-network-calibration-using-pytorch-c44b7221a61.

[83] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," 2017.