Department of Information Engineering and Computer Science

Masters's Degree in
Artificial Intelligence Systems

FINAL DISSERTATION

Systematic analysis of neural networks performance and generalization capabilities with application to the automatic assessment of lung ultrasound data from COVID-19 patients

Supervisor

Prof. Libertario Demi

Co-supervisor

Federico Mento

Author

Zihadul Azam

Academic Year 2021/2022

Thanks to

In the name of Allah, the Most Gracious and the Most Merciful

Alhamdulillah. First and foremost, I would like to praise Allah S.W.T. the Almighty, the Most Gracious, and the Most Merciful for His blessing given to me during my study and in completing this thesis. May Allah's blessing goes to His final Prophet Muhammad (peace be upon him), his family and his companions.

Words cannot express my gratitude to my supervisor Prof. Libertario Demi and co-supervisor Federico Mento for their invaluable patience and feedback. I also could not have undertaken this journey without ULTRa Lab teammates, who generously provided knowledge and expertise.

Sincere thanks to all my friends, especially to Shivam K., Zuhaer A., Edvig S., Denis S., Maicol F., David K., Paolo R., Lisa T., Omar M., Davide T., Berat M., Guido T., Wamiq R. and Francesco M. for their kindness and moral support during my study. I am also grateful to my classmates for their support and friendship during this journey.

Last but not least, my deepest gratitude goes to my beloved parents; Mr Shafiul Azam and Mrs Zannatul Ferdous, and also to my little sister Ms Tanzia Azam for their endless love, prayers and encouragement. Thank you very much.

— Zihadul Azam, 19 October 2022

Contents

Abstract **2**

1 Introduction **3**
1.1 Medical Imaging . . . 4
1.2 Ultrasound . . . 5
1.2.1 Physical quantities . . . 6
1.2.2 Generation and detection of ultrasound . . . 9
1.2.3 Interaction of ultrasound with matter . . . 11
1.2.4 Ultrasound beam shape . . . 14
1.2.5 Ultrasound image generation . . . 15
1.2.6 Safety . . . 17
1.3 Lung ultrasound . . . 18
1.3.1 Covid-19 . . . 20
1.4 Deep Learning . . . 21
1.4.1 Feedforward Networks . . . 23
1.4.2 Cost functions . . . 24
1.4.3 Output units . . . 24
1.4.4 Back-Propagation . . . 25
1.4.5 Regularization . . . 28
1.4.6 Optimization . . . 29
1.4.7 Convolutional Neural Networks . . . 31

2 Dataset **35**

3 Methodologies **37**
3.1 Frame-level scoring system . . . 37
3.2 Explainable AI with Grad-CAM . . . 39
3.3 Generalization capability of the network . . . 39
3.4 Training setup . . . 40

4 Experimental results **42**
4.1 Frame-level scoring system . . . 42
4.2 Grad-CAM . . . 43
4.3 Generalization capability of the network . . . 44

5 Conclusion **46**

Bibliography **47**

Abstract

The Corona virus disease 2019 (COVID-19) pandemic has revealed the utility of Lung Ultrasound in the diagnostics field thanks to its rapidity, transportability and lack of ionizing radiation. In the past years, different studies have been conducted to propose deep learning (DL) solutions for the automatic assessment of lung ultrasound (LUS) data from Covid-19 patients. However, the majority of them suggest multi-task or novel network solutions. This work aims to show and prove that the application of state-of-the-art convolutional neural networks (CNNs) can provide more accurate frame-level classification compared to existing methods. In fact, results from experiments have shown that ResNet-18 performs better than existing models with an F1-Score of 0.659. Furthermore, this work will prove that with the same amount of data provided to the model, training from scratch results in better performance than with the pre-trained weights. Moreover, for the first time, the generalization capability of the neural network will be assessed across different medical centers, on lung ultrasound frames. Using half of the LUS data from one medical center for training shows that the pre-trained model can generalize across different medical centers in testing and comparable performance can be achieved with less representative data. In addition, with the help of the Grad-CAM, an explainable AI algorithm, the behaviour of the best model has been assessed. This algorithm can highlight areas of the image that are spatially accountable for the model's class prediction. The outputs of this algorithm have proven the learning ability of the best CNN model. The model has succeeded in learning relevant artifacts from the LUS data.

Chapter 1
Introduction

COVID-19 can cause lung complications such as pneumonia and, in the most severe cases, acute respiratory distress syndrome, (ARDS) [1]. In pneumonia, the lungs get filled with fluid and become inflamed, leading to breathing difficulties [1]. Breathing problems can become severe and the patients can need hospital care with oxygen or possibly a ventilator. COVID-19 pneumonia is a novel coronavirus that spreads across patients, both symptomatic and asymptomatic, simply through close contact and respiratory droplets [2]. The high infection rate and increasing mortality resulted in a shortage in testing capacity and supply of medical equipment [3]. Furthermore, the low sensitivity of the Reverse Transcription Polymerase Chain Reaction (RT-PCR) test and the high rate of false negatives in COVID-19 diagnosis, added uncertainty in PCR-based clinical evaluations and interpretations [4]. These circumstances developed a strong need for identifying alternate potential methods for COVID-19 diagnosis worldwide [5]. Among the imaging modalities, chest computed tomography (CT) can be considered as a potential alternative due to its high sensitivity for the diagnosis of COVID-19 [6]. However, its limited availability, need for patient mobility, and the long-term risk due to radiation exposure limits its applicability. Moreover, the cost factor cannot be ignored, most of the hospitals and diagnostic centers in developing countries cannot afford any due to its excessive cost. Overcoming these limitations, LUS has lately been started to be used by clinicians and is considered a promising diagnostic tool for assessing COVID-19 pneumonia [7]. LUS-based assessment mainly relies on the interpretation of visual artifacts, making the analysis subjective and prone to error [8]. These visual artifacts mainly appear in two types, horizontal and vertical [9]. Horizontal artifacts represent the fully aerated condition of the lung and are formed as repeated reflections of ultrasound waves from the pleural line. In contrast, vertical artifacts are formed because of the formation of acoustic traps due to partial de-aeration of lung from different areas [9]. Several lung disorders can be connected to these artifacts [9]. But not always easy to detect these patterns through LUS imaging. It depends on the operator's knowledge and experience with the lung anatomy, and on the imaging parameters used during the acquisition process as well. For this purpose, a standardized acquisition protocol was proposed to ensure consistent evaluation across all of the data. Alongside, a semi-quantitative approach was proposed using a 4 level-based scoring system to represent the severity of lung damages suffering from COVID-19 [10]. Deep learning (DL) techniques have been employed to automate the evaluation process in an effort to aid the clinician performing this analysis. Work has been done to perform DL-based identification for the presence and absence of vertical artifacts in a LUS scan [11]. Another method utilized DL-based models to localize and classify LUS frames from patients based on the severity of COVID-19 indicated by the 4 level scoring system [12]. However, this dissertation aims to present a detailed analysis of state-of-the-art deep neural networks applied over LUS frames. In addition, experiments have been conducted to identify the minimum amount of good-quality data required to obtain state-of-the-art performance. Moreover, for the first time model's generalization capability is assessed across different medical centers using LUS data.

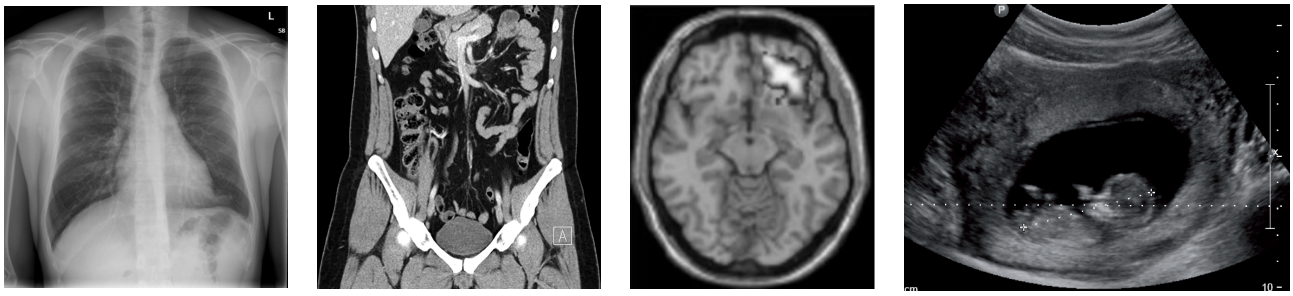
In Section 1.1, a short description of different medical imaging techniques is presented. In Section 1.2 and in Section 1.3, the fundamentals of Ultrasound and Lung Ultra Sound are described, respectively. Then in Section 1.4, the fundamentals of Deep Learning and modern Convolutional Neural Networks are discussed. Chapter 2 provides a detailed description of data acquisition, data labelling, data cleaning and dataset building. Chapter 3 deals with the methodologies and experimental setups. Chapter 4 illustrates numerical and graphical representations of the results obtained from various experiments. It also contains a qualitative and quantitative analysis of each experimental result. In the end, Chapter 5 is devoted to the final conclusions. After a quick summary, some discussions about achieved results, open issues and future developments are given.

1.1 Medical Imaging

Nowadays, disease diagnosis and treatment strategies depend heavily on medical imaging technologies. Modern imaging techniques allow clinicians to explore the interior of the human body and collect important information through image processing and analysis. These imaging techniques can provide a visual representation of some organ or tissue. Medical imaging diagnostics aims at the formation of images describing the internal structures of the body that skin and bones conceal. However, it also establishes a database of normal anatomy and physiology to make it possible to identify abnormalities.

According to the World Health Organization (WHO), approximately 3.6 billion diagnostic procedures are performed worldwide every year [13]. This number is tending to grow year by year, caused by the rising prevalence of chronic diseases such as cardiovascular, cancer, orthopedic and diabetes. Moreover, about 350 million of these are performed on children under 15 years of age [13].

X-ray, magnetic resonance imaging (MRI), computed tomography (CT) and ultrasound imaging are the most used techniques in hospitals and diagnostic centers (Fig. 1.1 illustrates some examples). However, in this final dissertation, only ultrasound technology will be discussed deeply. Since this final dissertation is about deep learning methodologies applied to ultrasound images, it is very important to start from the basics and have sufficient knowledge about the fundamentals of ultrasound technology. For that reason, the next section (1.2) of this chapter will be dedicated entirely to it. However, other imaging technologies will be covered shortly in the next three paragraphs.



(a) An example of chest X-ray image [14]

(b) An example of abdominal CT scan image [15]

(c) An example of brain FMRI image [16]

(d) An example of ultrasound image [17]

Figure 1.1: Most used medical imaging techniques

X-ray or radiography technology is one of the oldest and most common forms of medical imaging technique used by clinicians. It uses ionizing radiation to generate pictures of the body's internal structures. It sends waves with short wavelengths (from $0.01\,nm$ to $0.1\,nm$) through the body, which are then absorbed in different amounts depending on the materials they pass through. Dense materials like bone or metal will appear white on x-rays because of their high attenuation nature, while the lower attenuation levels in our lungs would cause the air to appear black. Materials like fat and muscle often appear as shades of grey. As was previously said, X-rays are ionizing radiations, which means they are a type of radiation with enough energy to ionize atoms or molecules by removing their electrons. This is potentially harmful to our body, and can cause skin redness, hair loss, radiation burns, also increase risk of developing cancer or acute radiation syndrome. Therefore, clinicians must use this imaging technology with attention and respect a specific radiation dose threshold for each patient.

Computed tomography (CT), sometimes called computed axial tomography (CAT), uses specialized x-ray technology and computing algorithms to create images of the inside of the body. It creates cross-sectional images with the help of computer processing. CT images are more detailed than conventional x-ray images and can reveal soft tissue and organs in addition to bones. A conventional x-ray uses a fixed tube that sends x-rays in only one direction, while a CT scanner uses the motorized x-ray source that shoots narrow beams of x-rays as it rotates around the patient. There are special digital x-ray detectors located directly opposite the x-ray source. As the x-ray passes through the patient, they are picked by the detectors and transmitted to a computer. Image sliced can eighter be displayed

individually in two-dimensional space or stack together to generate a three-dimensional image that can reveal abnormal structures for help the clinician plan and monitor treatments. A CT scanner can produce more detailed images of the internal organs, but it has some significant disadvantages. Compared to a traditional X-ray, the patient receives a higher radiation dose. Due to its size and weight, the scanner cannot be moved. It is also very expensive, thus not all the hospitals can afford one.

Magnetic Resonance Imaging (MRI) uses radio waves (radio frequency energy) and a strong magnetic field to produce accurate images of internal body structures. In hospitals and clinics, MRI is widely applied for disease staging, diagnosis, and follow-up. Compared to CT, MRI provides better contrast in images of soft-tissues, e.g. brain or abdomen. MRI uses magnetic fields and radio frequencies rather than ionizing radiations used in X-ray and CT. The magnetic field strength of an MRI machine is measured in Tesla (T). Typically, the majority of MRI systems used in the medical domain operate with magnetic fields ranging from 1.5T to 3T. These magnets are very powerful and can produce a magnetic field which is 50'000 times the earth's magnetic field. Although MRI does not emit ionizing radiations like in the x-ray and CT imaging process, it also has some disadvantages. MRI machines produce excessive noise during the test, it takes a long time for acquisition (between 30 to 60 minutes), patients with a magnetic implant or medical device cannot access it due to the presence of the giant magnet and similarly to CT, it is expensive.

1.2 Ultrasound

Ultrasound uses high frequency sound waves to create images of various organs and tissues of our body. It sends sound waves into the body and converts the returning sound echoes into an image. Although the ultrasound technology is having a great impact in the medicine field from the fifties of the last century, it was already a well-known technology in other domains, especially in naval defense.

Ultrasound had been discovered by Pierre Curie and his brother in 1880. The first practical application of ultrasounds is recorded during World War I. Big losses to the ships of the allied troops caused by the attack of the German submarines encouraged researchers to find a defensive system which can localize submarines underwater. In 1917 Professor Paul Langévin and his students invented a quartz sandwich transducer, which was able to detect submarines underwater with the help of ultrasound technology. At the end of WWI results and research ended up being archived, but started gaining attention again during WWII with the sonar. In 1942 Dr Karl Dussik, a psychiatrist, at the hospital in Bad Ischl (Austria) published a new technique with name *"Hyperphonography of the Brain"*. He was trying to locate brain tumors with a new method consisting of an ultrasound emitter at one end and an ultrasound receiver at the other. In 1947 he published the first ultrasound image, which was able to show the basic structure of the human brain and ventricles [18]. At the end of the WWII, a formal military doctor, George Döring Ludwig, started doing experiments to detect the presence and position of foreign bodies in animal tissues and in particular to localize gallstone, using reflective pulse-echo ultrasound technology similar to the sonar used for the detection of submarines. In his setup, very short pulses of ultrasound were employed using a combined transmitter/receiver transducer. In 1949 he published a 30-pages report where he declared to be able to detect gallstone in a dog's body with ultrasound waves [19]. In 1956, an English physician named Ian Donald used the one-dimensional A-mode to measure the parietal diameter of the fetal head. Two years later, in June 1958, he (with the collaboration of Tom Brown) published an article with the title "Investigation of abdominal masses by pulsed ultrasound" in the medical journal the *Lancet* [20]. This seminal article not only paved the way for ultrasound observations in obstetrics, but also the widespread use of ultrasound in medical diagnosis. They presented the ultrasound image of a female genital tumor. Brown invented the so-called *"two-dimensional compound scanner"*, which enabled the examiner to visualize the density of the tissue, which is often referred to as the turning point in the application of ultrasound in medicine [21]. The commercial use of ultrasound devices dates back to 1963 when the B-mode ("brightness mode") devices were constructed, enabling the examiner to visualize the two-dimensional image. In the mid-seventies, the "grey scale" was introduced (Kossoff, Garrett) leading to the introduction of real-time

ultrasound scanners. A decade later the Doppler effect served as the basis for the construction of the device that enabled the visualization of blood circulation (color flow Doppler ultrasound). Figure 1.2 shows an example of colour Doppler imaging.

Over the years, due to its diagnostic imaging capabilities, ultrasound has become very popular across all clinical applications, from obstetrics and gynecology, orthopedics and cardiology, to emergency medicine, prostate cancer and breast cancer detection. Its real-time imaging capability makes it unique compared to other imaging techniques (e.g. MRI). The speed, efficacy, cost-effectiveness and noninvasive nature of ultrasound imaging are some of the key features that have given this technology an edge over other imaging modalities like X-rays and Computed Tomography. Moreover, its size makes the device more accessible and portable. There are several different types of wireless probes available nowadays. By using them clinicians can carry out diagnosis anywhere: the probe and a tablet or smartphone are all that's needed (Fig. 1.3). Ultrasound has also the advantage of being safer than other modalities, as it does not involve the use of ionizing radiation or magnetic fields. In addition, ultrasound equipment is economical; even the most sophisticated ultrasound systems cost only about one-fifth of the price of a basic magnetic resonance imaging (MRI) system [22].

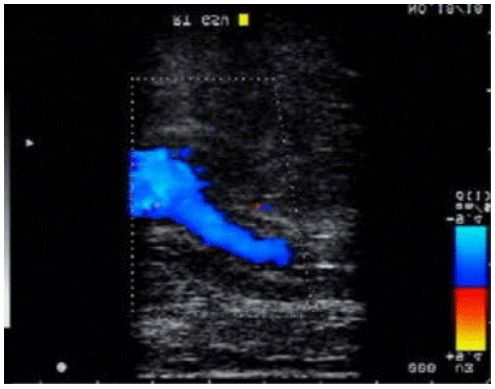


Figure 1.2: Example of ultrasound application: doppler ultrasonography shows a vein [23]

Figure 1.3: Wireless probe with tablet (Sonon 300L model) [24]

1.2.1 Physical quantities

A sound is a mechanical form of energy. The production of sound is caused by a vibrating source. The number of vibrations per second is known as the frequency, expressed in units called Hertz (Hz) and $1\,Hz$ is equal to 1 vibration per second. The frequency range can be divided into three main categories: audible sound, infrasound and ultrasound. Audible sound is the range of frequency which can be heard by the human ears. A frequency between $20\,Hz$ and $20'000\,Hz$ is considered an audible sound. Sound which has a frequency below the lower limit of human audibility is considered as infrasound. A study showed that some animals (e.g. whales, elephants, giraffes etc.) use infrasound to communicate over distance (up to a hundred miles in the case of whales). Then we have ultrasound, which has frequencies higher than the human perception (higher than $20\,kHz$). However, in diagnostic applications typically frequencies in the range from 1 to 20 MHz are utilized.

The propagation of ultrasonic energy requires a material medium: it cannot take place in empty space [25]. A source of ultrasound in contact with a medium transfers the mechanical disturbance to the medium, initiating vibrations in the "particles" of the medium [25]. A particle represents a small amount of volume present in the medium. All contained atoms of that volume respond uniformly to physical stimulation. When a particle vibrates, its vibration movements effect also its neighbouring particles, resulting direct transfer of energy. This transfer of energy continues sequentially from particle to particle. At the end of this process, we have ultrasonic energy propagated through the medium. This propagation of the ultrasonic energy does not cause any migration of particles across the medium, they only vibrate about their mean positions.

The propagation of ultrasound is known as longitudinal wave. The direction of displacement of medium particles is usually the same as the direction of the source of ultrasound. We can say that the ultrasound wave is propagated in the same direction as the direction of the disturbance caused by the source. However, it should be noted that ultrasound does not always propagate in the same direction as its source, which causes the disturbance. It may propagate in a direction that is perpendicular to the source (e.g. in bones). In that case, it is known as transverse wave. For simplicity, here ultrasound is considered as longitudinal waves only.

When the ultrasonic wave propagates through the particles it behaves like a piston moving rapidly in a confined space. In the forward direction, the piston compresses the medium particles in front of it, increasing their concentration per unit volume, hence creating increased pressure [25]. This is referred to as the compression phase, sometimes also called the condensation phase. When the piston moves in the reverse direction, the medium particles are decompressed, giving rise to a low-pressure phase, known as rarefaction [25]. The periodic movement of the piston, therefore, creates pressure waves in front of it, alternating between high and low-pressure [25]. Similarly the same happens when mechanical vibrations of a source propagate through the medium. They create alternative phases of compression and rarefaction in the particles involved during the propagation phase. For that reason, these vibrations also can be considered as pressure vibrations. Figure 1.4 illustrates the pressure variations during the ultrasound propagation through a medium.

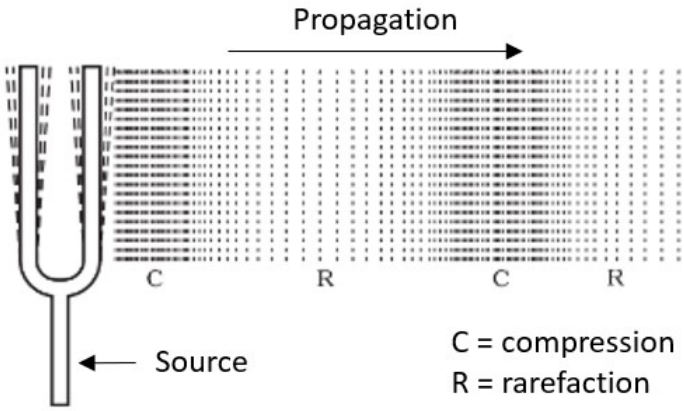


Figure 1.4: Pressure variations in the propagation of ultrasound

The periodic movements of medium particles about their mean positions, and the corresponding regular fluctuations in pressure, can be conveniently represented as a sinusoidal curve [25] (Fig. 1.5). The sinusoidal curve is characterized by amplitude and wavelength parameters. The amplitude is the magnitude of the particle displacement at a given moment during the propagation. It has a maximum and a minimum, which is related to the intensity of the ultrasound beam. On the other hand, the wavelength is the distance travelled by the pressure wave during one complete wave cycle. Each complete wave cycle is attributed to one vibration of the source [25]. The wavelength of the wave could be calculated by using the following formula:

$$\lambda = \frac{c}{f} \tag{1.1}$$

Where, c is the velocity (m/s), then f is the frequency $(s^{-1}$ or $Hz)$ and λ is the wavelength (m). Ultrasound waves travel at the speed of sound. In diagnostic ultrasound imaging, the average sound speed in soft tissues is assumed to be 1540 m/s. Note that in Figure 1.5 the amplitude is plotted against time, the period of the wave is the time for one complete cycle. The wave period is equal to the reciprocal of the frequency [25].

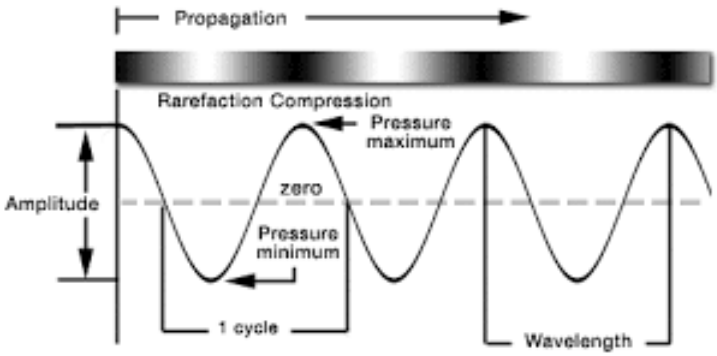


Figure 1.5: Variation of particle displacement in the direction of wave represented as a sinusoidal curve

The ultrasound propagates through a medium with a velocity. This velocity is not the same for all mediums. It varies from one medium to another, depending on the physical characteristics of the material. The density (ρ) and the compressibility (κ) are the two crucial physical properties on which the ultrasound velocity depends. As we all know, a denser material contains more massive particles than a less dense material. In case of the massive particles, a greater force is required to initiate the movement. In other words, the more massive a particle is, the greater its inertia [25]. The larger force required to overcome particle inertia in denser materials leads to the conclusion that wave velocity should be lower in materials of high-density [25]. This last statement sounds very odd because it contradicts common experience. First, let's consider the second physical characteristic of the medium, which will clear all doubts. The compressibility of a medium is the capacity of being mechanically deformed and reformed. In simple words, stiffness and rigidity. There is a direct relationship between compressibility and density. Materials with low density are easier to compress because their particles are located far from each other (e.g. Gas). On the other hand, the denser materials have low compressibility, because their particles have less space between them, so there is less room for compression. The closeness of the particles has an important role in the velocity of the ultrasound wave. It only takes small motions of the particles to transmit energy to the neighboring particles in a denser media, like bone, where the particles are relatively close to one another (low compressibility). As a result, the energy transfer is quick. That predicts a higher wave velocity in materials of low compressibility, even though the initial wave velocity was low [25]. As mentioned before, ultrasound waves travel at a speed of sound c, given by:

$$c = \sqrt{\frac{1}{\rho \cdot \kappa}} \tag{1.2}$$

Where ρ is the density and κ is the compressibility of the medium. As we can see from the equation 1.2, the velocity of ultrasound varies from medium to medium because of the combined effects of the density and compressibility. Table 1.1 shows the velocity of ultrasound in different materials. Among them, the velocity is higher in bone and lower in air. While the higher density of bone predicts reduced velocity of ultrasound, its lower compressibility increases the velocity to a greater extent. Here the compressibility is the most predominant factor. However, the velocity of ultrasound is affected by the medium temperature also. But the influence on velocity is quite minimal for temperature changes of just a few degrees. Since our body self regulate the temperature by the the homeostatic process, the effect of temperature on velocity can be considered insignificant in clinical ultrasound applications.

Material	Density, ρ $(kg \cdot m^{-3})$	Velocity, v $(m \cdot s^{-1})$
Air (0 °C)	1.2	330
Water (20 °C)	1000	1480
Blood (37 °C)	1060	1585
Muscle (37 °C)	1080	1600
Fat (37 °C)	930	1460
Liver (37 °C)	1050	1580
Bone (37 °C)	1900	3300

Table 1.1: Volumetric density ρ and velocity v of ultrasound in different materials at a given temperature

1.2.2 Generation and detection of ultrasound

The creation of mechanical vibrations with a high frequency is possible in a variety of ways. Among them, piezoelectric effect phenomenon is more natural for the wave generation. This phenomenon uses unique physical properties of the crystalline materials (Fig. 1.6), which ensure reversible conversion of two forms of energy, known as mechanical and electrical energies. When crystals of piezoelectric materials are compressed or stressed, electrical charge are produced on their surfaces. This behaviour is known as piezoelectric effect. Contrarily, when voltage is applied a piezoelectric crystal responds by expanding or contracting. This opposite behaviour is known as the reverse piezoelectric effect.

Figure 1.6: Piezoelectric Crystals [26]

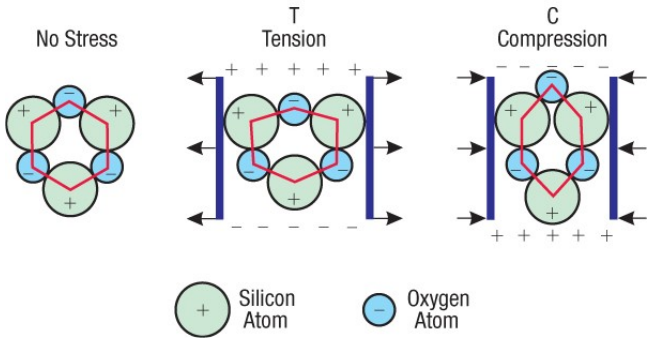


Figure 1.7: Piezoelectric effect

Ultrasound waves are produced by using the reverse piezoelectric effect, which converts electrical energy into mechanical energy (Fig. 1.7). On the other hand, the piezoelectric effect is used to detect high frequency mechanical vibrations, by converting them from mechanical energy to electrical signal. Because of the reversibility of this phenomenon, it is possible to use the same crystal to produce ultrasound, and subsequently to detect echoes returning to the crystal from a reflector at some distance away [25].

Devices which are responsible for converting one form of energy into another are called transducers. Today, modern transducers are made of many small, carefully arranged piezoelectric elements that are interconnected electronically. For simplicity, consider a transducer made with a single piezoelectric crystal. The Multi-crystal transducers will be discussed later. The essential components of the single crystal transducer are shown in Figure 1.8. The crystal element is the most important component of the transducer [25]. It is a thin disc of piezoelectric material near the front surface of the transducer [25]. The thickness of the crystal material controls the frequency of vibration. A vibrating crystal transmits ultrasound in both directions from its two surfaces: up surface and back surface (Fig. 1.8). From the up surface the ultrasound waves get transmitted toward patient's body. Whereas, the back surface transmits waves in the opposite direction, towards the transducer. The crystal thickness is chosen such that the vibrations at the two surfaces will reinforce each other every time the ultrasound makes a round trip internally from one face to the other and back to the first face [25]. Reinforcement

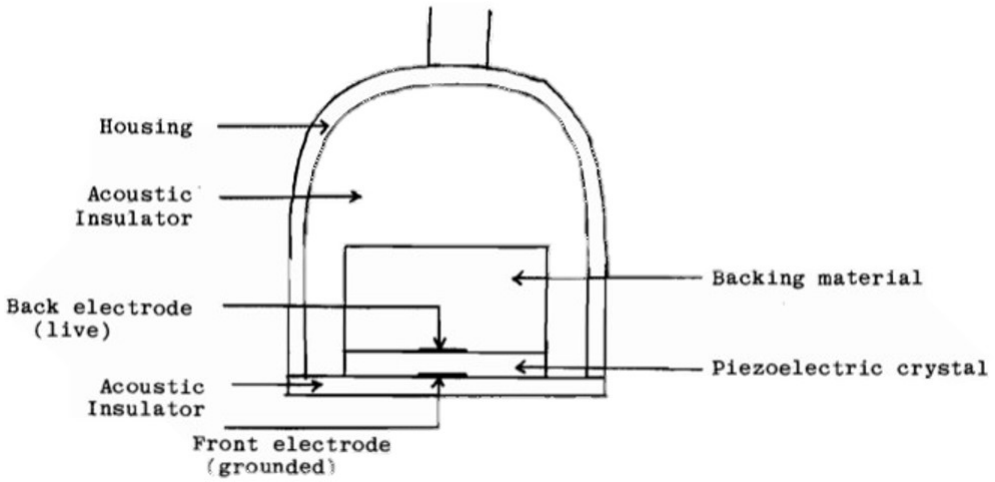


Figure 1.8: Components of a single crystal [25]

takes place if the distance covered by the ultrasound during the round trip equals a whole wavelength of the ultrasound wave [25]. The wave then arrives at a crystal face in the same phase of the wave cycle as the preceding disturbance which caused it [25]. Consecutive vibrations on the crystal faces will then reinforce each other through what is known as constructive interference in wave theory [25]. The reinforcements result prolonged self-sustenance of vibrations, a condition called resonance [25]. In the medical domain, the thickness of the crystal material varies from 0.1 mm (for higher frequency) to 1.0 mm (for lower frequency). The thinner the crystal, the higher the frequency is. The back and front surfaces of the crystal are coated with thin electrically conducting material to facilitate the connections to the electrodes, which supply the potential difference for pulsing crystal [25]. The front electrode, called also ground electrode, is used to protect the patient from electrical shock. On the other hand, the back one, called the positive electrode, is used for the live electrical connection. Moreover, the electrodes are used also during the receiving phase, they pick up the piezoelectric signals generated when the returning sound echoes hit the crystal. However, the front side of the transducer is covered with an electrical insulator, which makes direct contact with the patient during the imaging process.

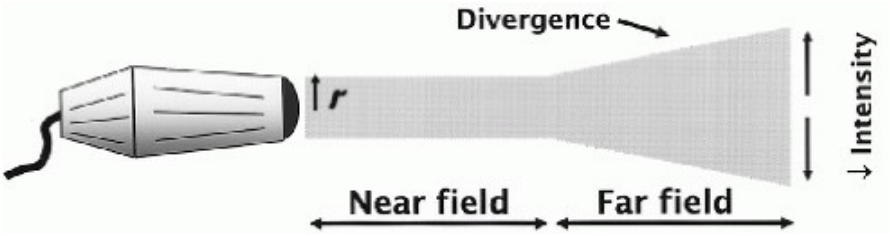


Figure 1.9: Near field and far field of a transducer [25]

Transducer design is critically important to optimal image creation. An important feature of ultrasound is the ability to direct or focus the beam as it leaves the transducer [25]. This results in a parallel and cylindrical shaped beam [25]. Eventually, however, after propagating for a certain distance, the beam will begin to diverge and become cone shaped (Fig. 1.9). The proximal or cylindrical portion of the beam is referred to as the near field or Fresnel zone [25]. When it begins to diverge, it is called the far field or Fraunhofer zone [25]. Within this portion of the beam, a decrease in intensity occurs. The length of the near field (l) is defined as:

$$l = \frac{r^2}{\lambda} \tag{1.3}$$

Where r is the radius of the transducer and λ is the wavelength of the emitted ultrasound.

The back side of the crystal has a block called backing block. It is made of a material such it can

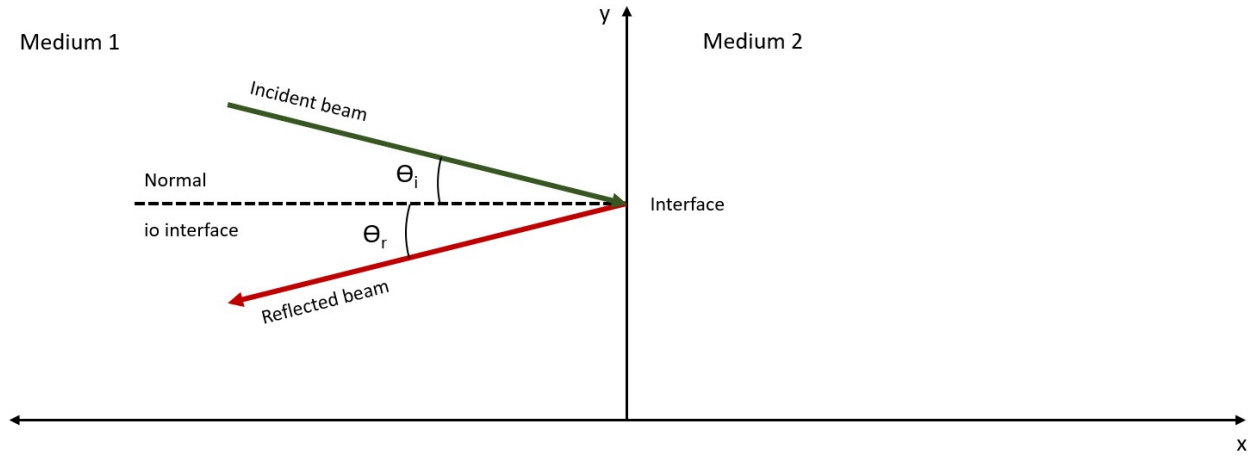


Figure 1.10: Specular reflection

absorb ultrasound waves. This block is a very important element of the transducer, it allows it to absorb ultrasonic energy transmitted back into the transducer and make sure to dump quickly the vibrations of the crystal following a pulsation. Moreover, the damping also controls the pulse length, which affects image axial resolution.

1.2.3 Interaction of ultrasound with matter

The proper usage of ultrasound imaging or therapeutic procedure depends on how ultrasonic waves interact with the components of our bodies. Over a chosen area of interest, the subject's tissues are penetrated using ultrasonic waves. The ultrasonic energy then interacts with the tissues along with its path. The interaction is different from matter to matter, influenced by the characteristics of the ultrasound wave, as well as the physical properties of the tissue through which the beam passes [25]. Different materials interact differently with the transmitted ultrasound waves, depending on the properties of the medium particles. In order to understand the interaction, the most important property to consider is the acoustic impedance (Z) of the medium. It is a measure of the resistance of the particles of the medium to mechanical vibrations [25]. The resistance increase or decreases in proportion to the density (ρ) and the velocity (c) of the ultrasound waves. The medium parameter acoustic impedance (Z) could be defined as:

$$Z = \rho \cdot c \tag{1.4}$$

Where, Z is measured in $\frac{Pa \cdot s}{m}$ or $Rayl$. Positions in tissues where the values of acoustic impedance fluctuate are crucial for ultrasound interactions. These positions are called acoustic boundaries, or tissue interfaces [25].

1.2.3.1 Reflection of ultrasound

When a beam of ultrasound makes contact with a acoustic boundary, some part of the beam energy pass through and get transmitted across the boundary, while some of the energy get redirected backwards toward the source. Two possible reflections can occur, depending on the size of the boundary relative to that of the ultrasound beam, or on irregularities of shape on the surface of the reflector. The first one is specular reflections, which occur when the surface of the boundary is smooth and larger than the beam dimensions. The other type of reflection is non-specular reflections, occurring when the interface is smaller than the beam.

Specular reflection occurs when the sound pulse encounters a large smooth boundary, such as an organ capsule. Sound is reflected from interfaces where there is a change in tissue density and compressibility (impedance – will be described more in detail later). The angle between the incident beam and the perpendicular direction to the interface is called *angle of incidence*, marked as θ_i in Figure 1.10.

The perpendicular line to the interface is also called *normal* to the reflecting surface. The reflected ultrasound will be on the opposite side of the normal with respect to the incident beam, and the angle it formed with the normal is called *angle of reflection*, marked as θ_r [25]. For the specular reflection the angle of reflection is equal to the angle of incidence, $\theta_r = \theta_i$. The probability that an echo will go back toward the transducer and be detected (after the specular reflection) increases as the angles θ_i and θ_r decrease.

The intensity of an echo due to the specular reflection depends on the angle of incidence as well as the difference in acoustic impedance values of the two media forming the boundary [25]. This difference in the acoustic impedance is known as *acoustic mismatch*, annotated with Z. The ideal specular reflection would be when the ultrasound beam strikes a reflector at 90° to the surface of the boundary and it is known as *normal incidence*. In that case, the angles θ_i and θ_r will be equal to 0° and the echos would go back to the transducer with a high probability of being received. In this special case of specular reflection, the echo intensity in relation to the intensity of the ultrasound beam incident upon the boundary is given by the relation:

$$\frac{I_{\theta_r}}{I_{\theta_i}} = \frac{(Z_1 - Z_2)^2}{(Z_1 + Z_2)^2} \tag{1.5}$$

$$R = \frac{I_{\theta_r}}{I_{\theta_i}} \tag{1.6}$$

Where I_{θ_r} is the intensity of reflected echo, I_{θ_i} is the intensity of the incident beam at the boundary, Z_1 is the acoustic impedance of the first medium and Z_2 is the acoustic impedance of the second medium. The ratio $I_{\theta_r}/I_{\theta_i}$ is called the reflection coefficient, annotated as R. It represents the proportion of beam intensity that is reflected from the interface. The reflection coefficient is determined by the difference of the Z value (from Eq. 1.5). A large difference between Z of two media at an interface produces a large reflection coefficient, which give rise to a large echo, whereas a small difference between Z values produce small echoes. As mentioned earlier, the value of Z changes from material to material. Table 1.2 contains acoustic impedance values of different materials which are part of our body.

Material	**Acoustic impedance, Z ($Rayl$)**
Air (0 °C)	396
Water (20 °C)	1.48×10^6
Fat (37 °C)	1.34×10^6
Brain (37 °C)	1.58×10^6
Kidney (37 °C)	1.62×10^6
Soft tissue (average) (37 °C)	1.63×10^6
Liver (37 °C)	1.66×10^6
Blood (37 °C)	1.68×10^6
Muscle (37 °C)	1.71×10^6
Bone (37 °C)	7.60×10^6

Table 1.2: Acoustic impedance Z in different materials at a given temperature

From the table 1.2 we can notice that the difference between soft tissues are not significant, which implies that the reflections at boundaries between soft tissues will generally give rise echoes with a reduced amplitude. We can also observe that the Z-value of the air (and for other gaseous materials) is much lower than soft tissues. Therefore, soft tissue/gas acoustic interface is formed, a significant amount of energy is reflected.

When the reflecting interface is irregular in shape, and its dimensions are smaller than the diameter of the ultrasound beam, the incident beam is reflected in many different directions [25]. This behaviour is known as non-specular reflection, or scattering.

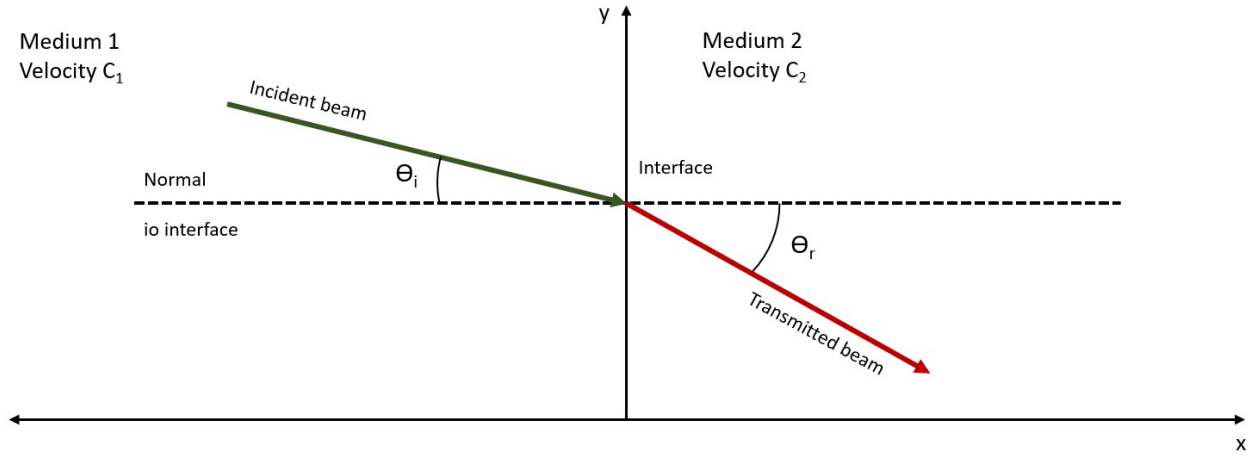


Figure 1.11: Refraction of ultrasound

1.2.3.2 Refraction of ultrasound

Refraction is a change of beam direction at a boundary between two media in which ultrasound travels at different velocities [25]. Figure 1.11 shows phenomenon of refraction. It happens when the angle incidence is not equal to zero. When a ultrasound beam strike an interface with an incident angle not equal to zero, a part of the beam energy go back toward the transducer after specular reflection, and the remaining energy pass through and get transmitted into the second medium. Depending on the relative velocities of the ultrasonic waves in the two media, the transmitted beam propagates through the second medium with a deviated angle that is either towards or away from the normal. Snell's law describes the relationship between the angles and the velocities of the waves [25]:

$$\frac{\sin(\theta_i)}{\sin(\theta_r)} = \frac{c_1}{c_2} \tag{1.7}$$

Where, θ_i is the angle of incidence, θ_r angle of refraction, c_1 velocity of ultrasound in medium 1 and c_2 velocity of ultrasound in medium 2. The deviation of ultrasonic energy into new directions contributes to the loss of beam intensity. A refraction may also be the cause of artefacts.

$$T = R + 1 \tag{1.8}$$

However, we can calculate a new coefficient called transmission coefficient from the *reflection coefficient* (Eq. 1.6). This measures the amount of energy that passes through the interface.

1.2.3.3 Absorption of ultrasound

When ultrasound waves get propagated through the medium, some of the energy gets transferred to the medium, where it gets transformed into another form of energy, mostly in heat. This process is called absorption. The capacity of the medium to absorb energy from the beam depends on the three main variables:

1. the viscosity of the medium
2. the relaxation time of the medium
3. the beam frequency.

Viscosity is a measure of the frictional forces between particles of the medium as they move past one another [25]. The greater these frictional forces the more heat generated by the vibrating particles [25]. Therefore, absorption of ultrasound increases with increasing viscosity. Relaxation time is the time taken by the medium particles to return to their initial position after the vibration caused by the ultrasound waves. When the relaxation time is short, the vibrating particles could return to their normal position before the next ultrasound pulse gets transmitted. Contrarily, when the relaxation

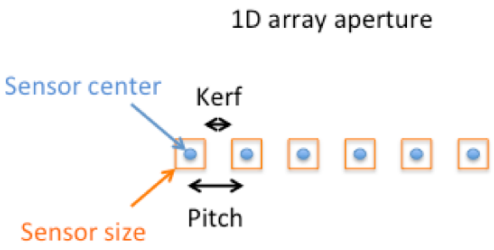


Figure 1.12: This figure shows the different sensor distributions for a one-dimensional (1D) [27]

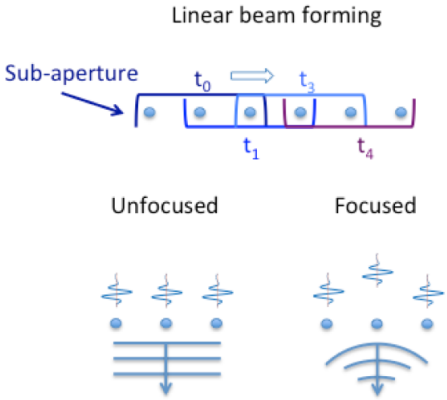


Figure 1.13: Linear array beam forming in cases with a focused and an unfocused beam. [27]

time is long, the next pulse can find the medium particles already in vibrating status. The new compression and the particles may then be moving in opposing directions, thus resulting in additional dissipation of energy from the beam [25]. Therefore, the longer the relaxation time of a medium, the higher the absorption of ultrasound [25].

High frequency means that the medium particles move each other at a high rate, which generates more frictional heat. Increased frequency also reduces the probability that, following an ultrasound pulse, the vibrating particles will return to their equilibrium position before the next pulse gets transmitted, thereby increasing energy absorption as the new wave moves in opposition to the relaxing particles. We can conclude that absorption of ultrasound increases with increasing beam frequency [25].

It is significant to remember that high-frequency ultrasound waves absorb energy quickly. That means high frequencies cannot be employed to examine long distances in tissue. High-frequency waves reduce beam penetration capacity. However, high frequency helps to get images of better resolution.

1.2.4 Ultrasound beam shape

The ultrasound imaging technique aims to illustrate the body's internal structure by using high-frequency ultrasound waves. Transducer is a fundamental component of a ultrasound system. It is capable to generate ultrasound waves and also receiving returning waves after the reflection with human internal body elements. Generally, the transducer is placed over the patient's skin in the area of interest. However, the transducer does not make contact directly with the skin. Typically, a suitable gel is applied between the transducer surface and the patient's skin in order to exclude air.

The transducers are normally made as an array of sensors, in order to shape the beam into the domain and recombine the received signals from an image [27]. This process is called beamforming. The distance between the center of two sensors is called *Pitch* and the space between two sensors is known as *Kerf* [27] (Fig. 1.12). Each sensor can be excited by a signal having its amplitude, phase, and waveform [27]. However, sensors are generally grouped into subgroups, also called sub-apertures (Fig. 1.13). Within one sub-aperture, the signals can be transmitted with the same waveform, but with different phases and amplitude.

Array types for ultrasound transducers might vary based on the needs of the clinician. The three most common ultrasound transducer types are: linear, curvilinear and phased array (Fig. 1.14). However, in this section only the linear array mechanism will be discussed. The linear array beamforming sub-aperture is used for transmitting and receiving ultrasound signals. The receiving signal represents the structure seen by the ultrasound waves over depth and in front of the sub-aperture. This signal is called an A-scan. Subsequently, this sub-aperture is linearly shifted over the entire array to obtain

Transducer Type	Linear	Curvilinear	Phased array
Frequency	5-10 Mhz	2-5 Mhz	1-5 Mhz

Figure 1.14: The three types of transducers [25]

multiple A-scans, ultimately forming an image line by line [27]. The sensors that belong to a sub-aperture could be excited by signals that share the same phase, i.e., the unfocused case, or have different phases, as in the focused case [27].

The transmitted beam can be either unfocused or focused to a specific focal point. In the first case, no phase coefficients are applied to the elements. In the second case, time delays are applied to each element. These delays are applied to allow the transmitted beam to be focused at a specific depth [25]. Focusing can improve the lateral resolution and increase the pressure amplitude, thereby the penetration depth [28].

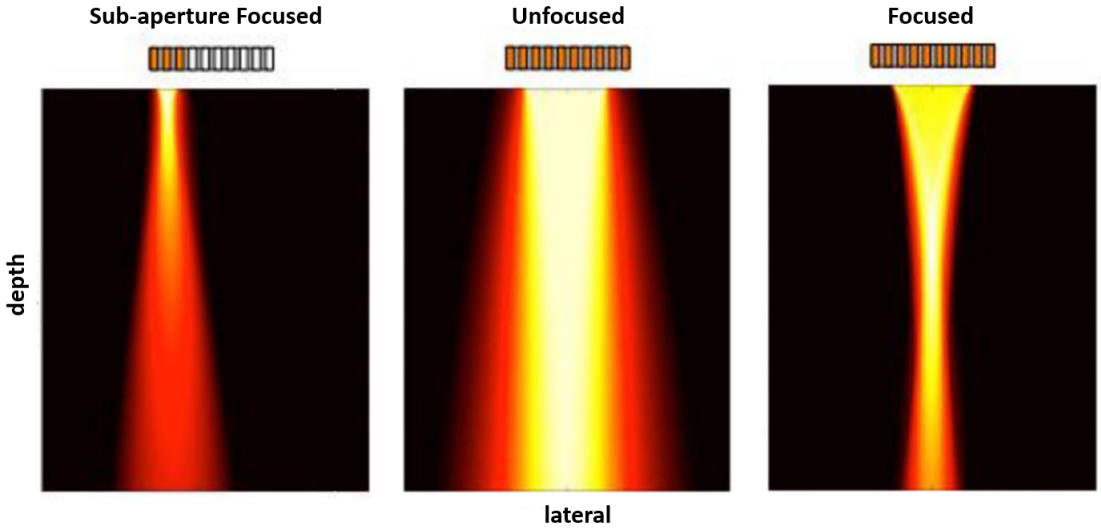


Figure 1.15: This figure shows the different spatial distribution of pressure amplitudes when sub-aperture focused, unfocused and focused. [27]

1.2.5 Ultrasound image generation

An ultrasound transducer emits high-frequency sound pulses toward the zone of interest of the patient. As was already mentioned, a portion of each pulse travels through the interface between two media, while the remainder is reflected and travels back toward the transducer. The transducer of the ultrasound equipment does not only act as a sound source but also as a receiver. In fact, after emitting sound pulses, it remains in listening mode. At the transducer, the returning echo will interact with the piezoelectric crystal and generate an electric signal [25]. The signals generated by the returning echoes at the transducer are electronically processed and organized in computer memory before being

displayed. For each position of the ultrasound beam, a set of signals will be recorded along the beam path, corresponding to reflecting boundaries lying at different distances from the transducer [25]. The set of signals produced along one beam path may be referred to as a scan line [25]. It represents single-dimensional information along the beam path. By sweeping the ultrasound beam across the subject ("scanning") in a selected direction, many other scan lines are generated to build a two-dimensional (2-D) image of a plane in the subject [25].

Determining the distance between the transducer and a reflecting interface is an essential parameter for composing ultrasound image. This is done by transforming temporal information to the spatial information. In fact, to achieve this, two events are electronically "marked", the moment the transducer is pulsed, and the moment it receives the returning echo from the tissue boundary [25]. Assume d is the distance between the transducer and a reflecting boundary, we assume also that t is measured time interval between pulsing and reception, then the distant traveled by the ultrasound beam during t time interval is $2d$. To determine the distance d the following simple relationship is being used by the ultrasound system:

$$2d = c \times t \tag{1.9}$$

Where d is the distance, c is the velocity of ultrasound in the transmitting medium and t is the time. Ultrasound systems use the average soft tissue velocity value of 1,540 m/s to calibrate distance measurements [25]. Although, as we saw before, ultrasound velocity varies from medium to medium. This constant velocity works because the variance of the velocity is very low among the soft tissues, and they are very close to their average value. Errors in distance measurement are therefore not significant.

Once the diagnostic information is recorded and transformed into temporal information for the spatial data, it has been displayed as an image. There are different display modes, and they are:

- **A-mode:** a single transducer scans a line through the body with the echoes plotted on screen as a function of depth [23].
- **B-mode:** a linear array simultaneously scans a plane through the body that can be viewed as a two-dimensional image on screen [23].
- **M-mode:** M-mode means motion-mode. In m-mode a rapid sequence of B-mode scans whose images follow each other in sequence on screen enables doctors to see and measure range of motion, as the organ boundaries that produce reflections move relative to the probe [23].
- **Doppler-mode:** This mode makes use of the Doppler effect in measuring and visualizing blood flow [23].

Moreover, there are some essential parameters which describe the imaging system:

1. **Spatial resolution:** The smallest spatial distance for which two structures can be distinguished in the final image. In ultrasound the spatial resolution can be axial, lateral or elevation. Spatial resolution is expressed in mm. Typical values are from 0.1 mm to 1 mm. [27].
2. **Temporal resolution:** Is the time interval between two consecutive images, has Hz as unit of measure. Temporal resolution can be expressed with $T_r = \frac{c_0}{N_l 2L_{max}}$, where N_l are the number of lines in the image and L_max is the maximum depth. Typical values are from 1 to 1000 Hz. [27].
3. **Penetration depth:** The penetration depth is expressed in cm, which is the larger depth for which the signal-to-noise ratio (SNR) can be maintained above the receiver sensibility. It depends on the attenuation in the investigation domain, on the beam shape, on the maximum pressure generated by the transducer and on the minimum detectable pressure [27].
4. **Array aperture:** Expressed in cm^2, namely the area of the surface encompassing all the array elements. It depends on the number of sensors [27].

5. **Field of view (FOV):** Expressed in cm^2 or cm^3, that is the size of the area represented in the final image. Factors affecting its value are the selected beamforming strategy, the penetration depth and the array aperture [27].

1.2.6 Safety

Ultrasound is considered one of the safest imaging techniques. It does not use ionizing radiation to capture diagnostic information. However, there are some indexes in which values must be monitored during the acquisition process. The first one is mechanical index:

$$MI = \frac{P^{-}_{max}}{\sqrt{f_0}} < 1.9 \tag{1.10}$$

Where:

P^{-}_{max}: is the maximum negative pressure.
f_0: is the central frequency in MHz

Moreover, there is another important index, which is thermal index. As explained earlier in Section 1.2.3.3, when ultrasound propagates through a medium, some mechanical energy gets absorbed by the medium in form of another energy (mostly as heat). The scope of the thermal index is to measure the thermal bioeffects caused by the ultrasound beam. This index is very important, because the biological effects may include tissue eating and mechanical damage resulting in haemorrhage. The thermal index should remain below 1 during the acquisition process and it is expressed as:

$$TI = \frac{W_P}{W_{deg}} < 1 \tag{1.11}$$

Where:

W_P: is attenuated acoustic power at the depth of interest.
W_{deg}: is estimated acoustic power necessary to raise the tissue equilibrium temperature by 1° celsius.

1.3 Lung ultrasound

Lung ultrasound (LUS) is the ultrasound imaging technique applied specifically to the lung, which allows to explore the organ and collect important data using output images. The lungs are a pair of spongy, air-filled organs located on either side of the chest (thorax). They have around 480 million microscopic air sacs called alveoli. In the alveoli, oxygen from the air is absorbed into the blood and carbon dioxide travels from the blood to the alveoli, where it can be exhaled [29]. Lungs are protected by a thin tissue layer called the pleura. By using LUS, it is possible to evaluate and detect thoracic diseases like pleuraleffusions, atelectasis, pneumothorax, and pneumonia [9]. Reading an ultrasound image is not an easy task. It requires expertise with the anatomy being studied and understanding of how ultrasound interacts with various types of material. However, there are two important artifacts which must be observed carefully during a lung ultrasound examination: horizontal artifact (A-line) and vertical artifacts (B-line). Figure 1.17 shows some ultrasound images with these two artifacts.

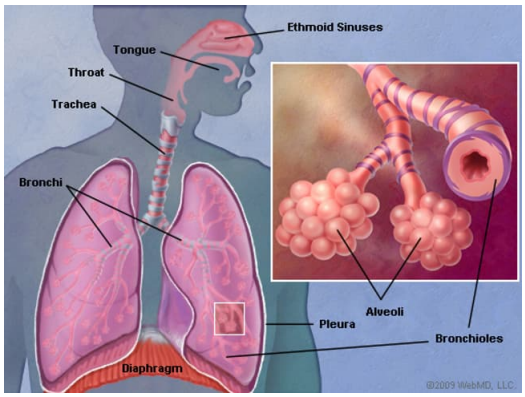


Figure 1.16: Lung anatomy [29]

In case of a normal, healthy lung, the pleural-line behaves as a perfect reflector to the acoustic waves, preventing the propagation beyond it. The lung volume mostly occupied with air that acts as a wall against ultrasound waves. In this case, horizontal artifacts appears in the ultrasound image: these are equidistant from each other, and their distance is equal to multiples of the distance between the pleural line and the probe [9]. Once the reflected wave reach the transducer, some part of that get reflected again, and has enough energy to reach the pleural line for the second time, and this create the mirror artifacts. In fact, the echo that arrives to the pleural line for the second time, get re-reflected as a second echo and reach the probe again. Therefore, there are two echoes coming from the pleural line, and consequently, the obtained spatial function highlights two pleura lines: the first is localized correctly while the second is localized at a double distance [9].

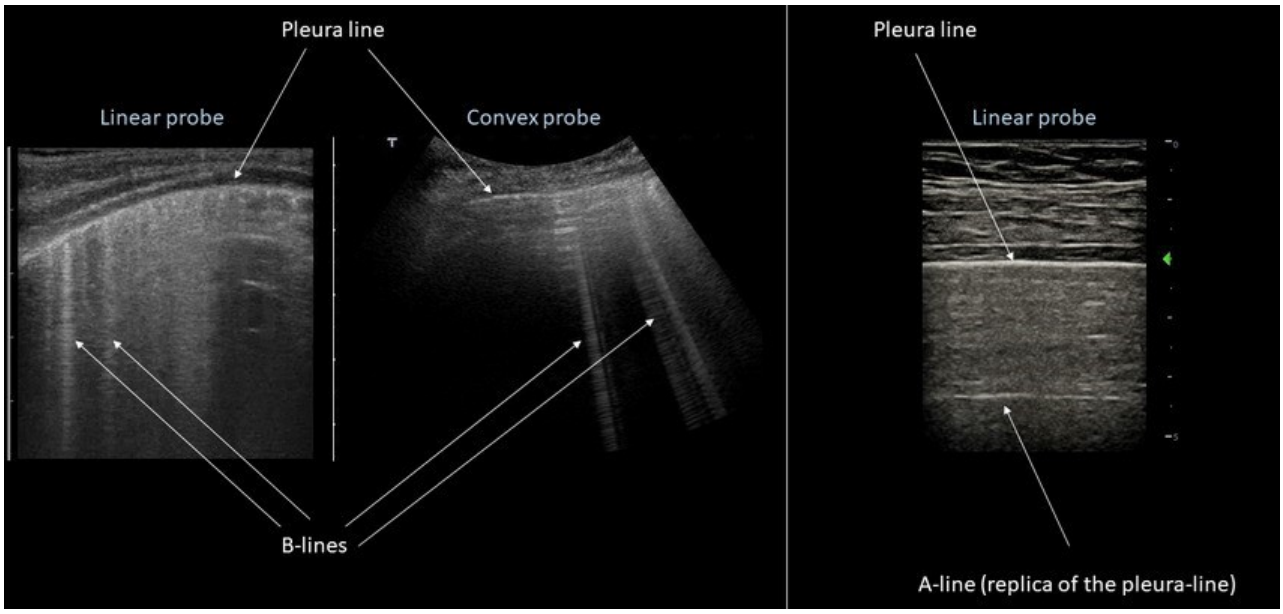


Figure 1.17: B-line artifacts as observable with a linear (left) and convex (center) probe and A-line artifact as observable with a linear probe (right). The pleura line is also indicated for the three images. [30]

While vertical artifacts are defined as hyper echoic artifacts which originate at the pleura line and lie roughly perpendicular to the latter. They appear where there is evidence of cardiogenic pulmonary edema (diffusely and homogeneously distributed), lung contusions, acute respiratory distress syndrome (ARDS), and pneumonia around consolidative cores [9]. But, vertical artifacts alone cannot differentiate the cause, even though exists evidence of a correlation between the appearance of this type of artifact and the presence of several pathologies [9]. The low specificity of vertical artifacts does not allow researchers and clinicians to adequately classify the various diseases that can give rise to these artifacts [9]. There are several conducted studies which tried to explain the cause for which vertical artifacts appear in ultrasound images. Among them so-called *"acoustic traps"* model is a reasonable work, which tries to explain this artifact with physical and mathematical evidence [31]. This study says that every vertical artifact which may appear in an LUS image is probably generated by multiple reflections between the walls of the lung aerated spaces. The wall is made of aerated alveoli, which wrap a small fluid volume. If the ultrasound wave is able to enter this trap with a sufficient amount of energy, it gets reflected multiple times between the trap's walls. Obviously, the transmitted acoustic wave must have enough energy to be able to enter the trap, get reflected multiple times and return echoes that have measurable energy which can be detected by the transducer. In this study, they considered the trap as a circle structure (Fig. 1.18), which can trap part of the energy carried by the transmitted ultrasound pulse and progressively reradiate the transducer, consequently generating vertical artifacts in the reconstructed image. vertical artifacts can appear in different shapes, the structure is dependent on the type of pathology affecting the lung [32].

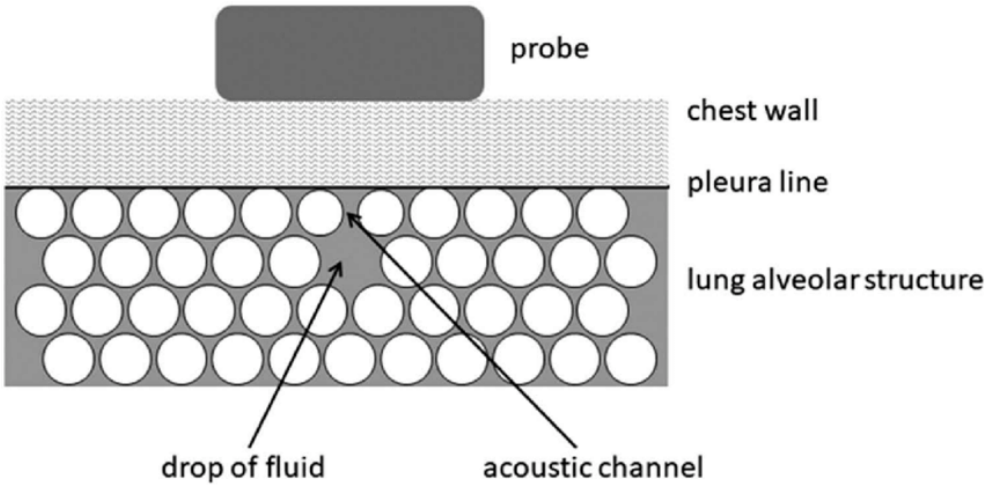


Figure 1.18: The figure illustrates an example of an acoustic trap: a drop of fluid surrounded by air spaces which are exposed to the penetration of ultrasounds through a transonic channel. [9]

However, besides horizontal and vertical artifacts there is another pattern which may appear in a LUS image, called white lung. When there is distributed damage affecting the pleural line or the alveoli, a wide area of high intensity appears in the reconstructed LUS image, which is believed to be composed of many juxtaposed B-lines [9].

1.3.1 Covid-19

During the Covid-19 pandemic, LUS was an essential and popular imaging technique in hospitals. The use of LUS became very frequent thanks to its unique characteristics. As mentioned earlier, it is non-invasive, does not emit ionizing radiation and the most important characteristic is its portability. According to Johns Hopkins Medicine, Covid-19 can cause lung complications such as pneumonia and, in the most severe cases, acute respiratory distress syndrome, or ARDS [33]. These lung diseases can decrease our respiratory capacity drastically and also can cause respiratory failure. Advance Covid-19 pneumonia determines widespread consolidations and artifactual patterns, which can be detected in a short period of time by using LUS. Moreover, LUS is very useful for monitoring lung alterations during the treatment process. However, in the earlier stage of this pandemic, there was no defined international standardized protocol for the acquisition process. There was no official indication about the use of LUS in the treatment of patients with COVID-19. Therefore, for standardizing the use of LUS on Covid-19 patients, researchers came up with a standard acquisition protocol, which marks 14 areas where LUS should be applied to have a better view of the lung condition and to monitor lung involvement during a specific treatment [10]. They developed a standardized approach regarding equipment and the acquisition protocol. The Figure 1.19 illustrates 14 areas where LUS should be applied, according the protocol mentioned above.

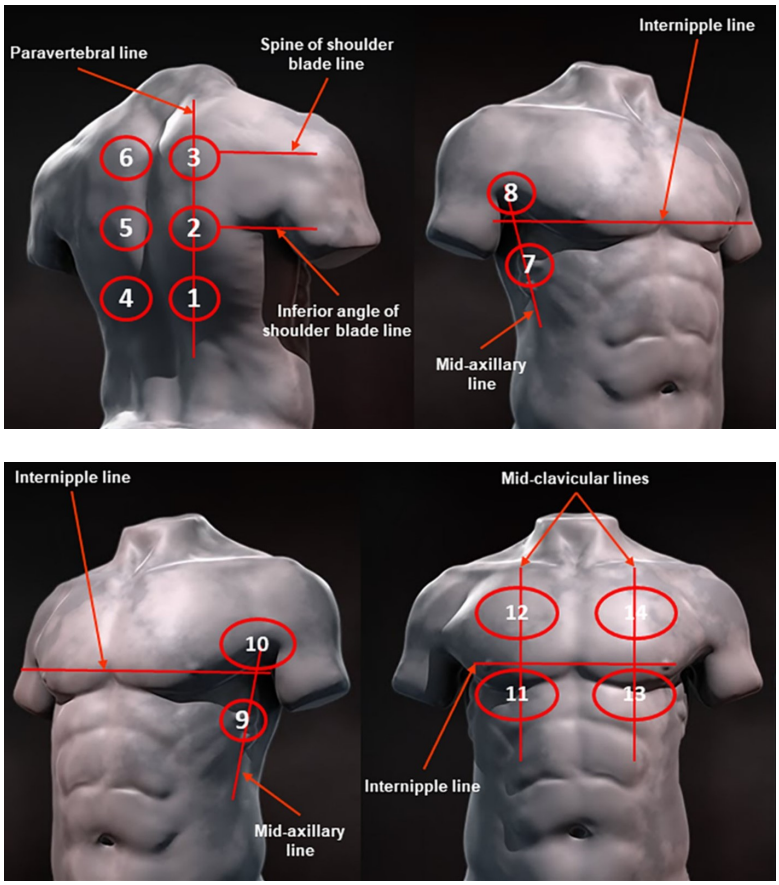


Figure 1.19: Schematic representation of the acquisition landmarks on chest anatomic lines. [10]

Fourteen areas (3 posterior, 2 lateral, and 2 anterior) should be scanned per patient for 10 seconds along the lines indicated here [10]. Scans need to be intercostal to cover the widest surface possible with a single scan [10]. In addition, they also proposed a scoring system to represent the severity of lung damage suffering from COVID-19, which will be discussed later in the Chapter 2.

1.4 Deep Learning

Deep learning (DL) is a subdomain of Machine Learning (ML). Like ML also DL aims to learn from data and previous experiences. But what is the meaning of learning? According to Mitchell (1997):

> "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" [34].

Where, a task T could be any of the following: classification task, classification with messing inputs task, regression task, machine translation task, anomaly detection task, denoising task, density estimation task, generating new data task, segmentation task, etc. Performance P is the measurement of model goodness. It could be the error rate on testing data or accuracy for the classification problem, and also the similarity measurement in case of new data generation task (e.g. novel images generation with a specific class). What about the Experience E? Machine learning algorithms can be broadly categorized as unsupervised or supervised by what kind of experience they are allowed to have during the learning process [34]. Most of the ML and DL algorithms are allowed to experience an entire dataset. A dataset is a collection of many examples with one or more features. Sometimes examples are called data points also.

For instance, the Iris dataset (Fischer, 1936) is one of the oldest datasets. It contains a collection of measurements of the different parts of 150 iris plants. Where, each plant is an example and different measurements are considered as features: the sepal length, sepal width, petal length and petal width. The dataset also records which species each plant belonged to. It contains examples from three different species. In ML and DL the data plays a key role, it is also called the "fuel"" of algorithms, without which the machine cannot learn. To build an ML or DL model usually the data are divided into multiple data sets. In particular, three data sets are commonly used in different stages of the creation of the model: training set, validation set and testing set. The model is initially fit on a training data set, which is a set of examples used to fit the parameters (e.g. weights of connections between neurons in artificial neural networks) of the model. The observations in a second data set, referred to as the validation data set, are successively predicted using the fitted model. The validation data set offers a fair assessment of a model fit on the training data set while adjusting the model's hyperparameters. Finally, the last dataset, such as testing data, is used to evaluate the goodness of the model on new and unseen data, it provides an unbiased evaluation of a final model fit on the training data set.

As mentioned earlier, ML and DL algorithms can be unsupervised or supervised. An unsupervised algorithm experience a dataset containing different features, then learn useful properties of the structure of this dataset [34]. A typical task of an unsupervised algorithm is clustering, where the algorithm learns to divide the dataset into clusters of similar examples. On the other hand, supervised algorithms experience a data set containing features, but each example is also associated with a label or target. For instance, the data points are labelled with the species in the Iris dataset. A supervised learning algorithm can study the Iris dataset and learn to classify iris plants into three different species based on their measurements.

Deep learning applications use a layered structure of algorithms called an artificial neural network (ANN). The structure of ANN is very similar to our brain. The human brain is composed of a network of neurons and each of us has approximately 86 billion of neurons. A neuron is made of:

- **Soma:** cell body, contains nucleus.
- **Dendrites:** a set of filaments departing from body.
- **Axon:** a longer filament (up to 100 times body diameter).
- **Synapses:** connection between dendrites and axons from the other neurons.

Neurons are information messengers. They use electrical impulses and chemical signals to transmit information between different areas of the brain, and between the brain and the rest of the nervous system. Everything we think and feel and do would be impossible without the work of neurons. Electromagnetic reactions allow signals to propagate along neurons via axons, synapses and dendrites. Synapse either excites or inhibits. Once a neuron's potential exceeds a certain threshold, a signal gets generated and transmitted along the exon [36].

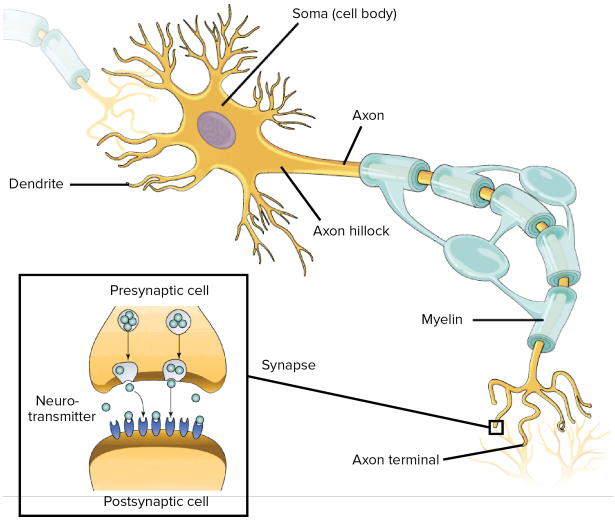


Figure 1.20: Neuron anatomy [35]

Similarly, also an artificial neural network (ANN) is made of multiple artificial neurons, which are interconnected between them and these artificial neurons are called perceptron (Fig. 1.21). A perceptron is a non-linear parameterized function with a restricted output range. It receives features $X = \{x_1, x_2, .., x_{n-1}, x_n\}$ as input and produces $\hat{y}$ as output value, where n is the number of features. Firstly, the input features X get multiplied by the weights θ, then they get summed. Successively, the weighted sum result passes through an activation function, which produces a binary output. If the input of the activation function is greater or equal to the threshold then it produces a (e.g. $+1$) as output, otherwise, it produces b (e.g. -1) as function output. Beside all input features, perceptron has another input value called bias, which allows to shift of the activation function's threshold by adding a constant. The perceptron can be expressed mathematically with the following equation:

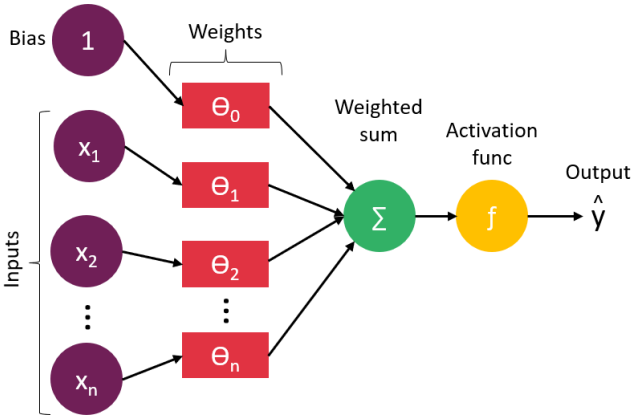


Figure 1.21: Perceptron (an artificial neuron)

$$\hat{y} = h(\theta_0 + \sum_{i=1}^{n} \theta_i \cdot x_i) = h(\theta_0 + \theta^T x) \tag{1.12}$$

Where, h is the activation function. An activation function decides whether a neuron should be activated or not. This means that it will decide whether the neuron's input to the network is important or not in the process of prediction using simpler mathematical operations. The purpose of an activation function is to add non-linearity to the neural network. Without non-linearity, all neurons will behave in the same way, which means the network will not be able to learn complex tasks. There are different types of activation functions could be used in a neural network, some of them are:

- **Binary step function:**
$$h(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$
- **Sigmoid:**
$$h(x) = \frac{1}{1 + e^{-x}}$$
- **Rectified Linear Unit (ReLU):** $h(x) = \max(0,\ x)$
- **Leaky ReLU:** $h(x) = \max(\alpha x,\ x)$ where, α is 0.1

1.4.1 Feedforward Networks

Deep Feedforward Networks, also called Feedforward Neural Networks are made of multi-layer perceptrons (Fig. 1.22). Until now we have talked about single perceptron and its mathematical theory. But, a single perceptron alone is not capable to handle complex problems. So, we need a network of perceptrons which can mimic our human brain functionality, specifically, we need multi-layer perceptrons. The goal of multi-layer perceptrons or a feedforward network is to approximate some function $f*$. For instance, for a classifier, $y = f*(x)$ maps an input x to a category y. A feedforward network defines a mapping $y = f(x;\theta)$ and learns the value of the parameters θ that result in the best function approximation. [34].

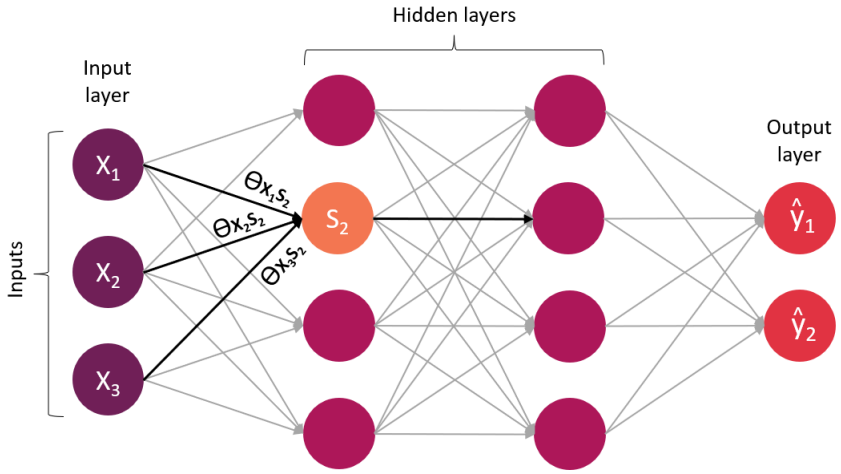


Figure 1.22: This is an example of a Deep Feedforward Network, which is made by a network of perceptrons interconnected. S_2 is a single perceptron with its parameters $\{\theta_{x_1S_2}, \theta_{x_2S_2}, \theta_{x_3S_2}\}$

These models are called feedforward because the information flows in the forward direction. From the X nodes inputs go through computations which define the function f and produce output $\hat{y}$. Feedforward networks are a very important part of machine learning, especially for computer vision. They form the basis of many important commercial vision applications. For instance, the convolutional networks used for object detection and recognition from photos or videos are a specialized kind of feedforward network.

Feedforward neural networks take inputs through the input layer. The number of input nodes in the input layer depends on the shape of the input. Feedforward neural networks are called networks because they are typically represented by composing together many different functions [34]. The model is associated with a directed acyclic graph describing how the functions are composed together [34]. For instance, let's consider three functions $f^{(1)}$, $f^{(2)}$ and $f^{(3)}$. They are connected in a chain, so we can define their relationship as $f(x) = f^{(3)}(f^{(2)}(f^{(1)}))$. These chain structures are the most commonly used structures of neural networks [34]. In this case, $f^{(1)}$ is known as the first layer of the network, $f^{(2)}$ the second and so on. The chain length defines the depth of the network. The name "deep learning" arose from this terminology [34]. The final layer of the network is known as output layer, which contains output nodes. During neural network training process, the goal is to learn the best parameters θ such $f(x)$ become capable to match $f^*(x)$. For instance, we want to train a neural network which is capable to distinguish three specie of the Iris dataset. The network takes features (different measurements of the plant, e.g. sepal length, sepal width, etc.) as input. In this case, we have to train the model such that it can find the best θ value for each directed arc such the network can predict the plant's specie correctly, as much as possible. Please note that the training examples specify directly what the output layer must do at each point x; it must produce a value that is close to y. The behaviour of the other layers is not directly specified by the training data [34]. The learning algorithm must decide how to use those layers to produce the desired output, but the training data do not say what each individual layer should do [34]. Instead, the learning algorithm must decide how to use these layers to best implement an approximation of f^* [34]. However, in the deep neural network,

each unit receives inputs from many other perceptrons and computes its own activation value.

1.4.2 Cost functions

Deep neural networks use iterative training methods to fit the model. This method feeds training data to the network again and again iteratively, such the network can learn by adjusting its parameters θ. Like us humans, also neural networks learn from past experiences. But the network cannot learn without knowing if it is doing well or doing worse with respect to the expected output. So, it is necessary to have a metric which can indicate the goodness of the current learning situation. For that reason, neural networks, especially feedforward networks need a cost function and the choice of the cost function is an important aspect of the design of a deep neural network. Most modern neural networks are trained using maximum likelihood [34]. This means that the cost function is simply the negative log-likelihood, equivalently described as the cross-entropy between the training data and the model distribution [34]. This cost function is given by:

$$J(\theta) = - \mathbb{E}_{x,y\sim\hat{P}_{data}} \log_{P_{model}}(y \mid x) \tag{1.13}$$

Where, $\mathbb{E}_{x,y\sim\hat{P}_{data}}$ is the expected value from the $\hat{P}_{data}$ distribution (real data distribution). For a single example the loss L would look like this:

$$L = - y \cdot \log(\hat{y}) \tag{1.14}$$

Where y is the ground truth and $\hat{y}$ is the estimated value by the network. The specific form of the cost function changes from model to model, depending on the specific form of $log\, _{P_{model}}$ [34]. However, the non-linearity of the neural network makes the loss non-convex. For that reason, the neural networks are usually trained by using iterative, gradient-based optimizers that merely drive the cost function to a very low value, rather than the linear equation solvers or the convex optimization algorithms with global convergence guarantees, which are usually used to train machine learning algorithms [34]. Stochastic gradient descent applied to non-convex loss functions has no such convergence guarantee and is sensitive to the values of the initial parameters. For feedforward neural networks, it is important to initialize all weights to small random values. The biases may be initialized to zero or too small positive values.

1.4.3 Output units

To apply gradient-based learning the representational choice of the output units or output nodes is very important. The choice of the cost function is tightly coupled with the choice of output unit [34]. Most of the time, the cross-entropy between the data distribution and the model distribution is used. The choice of how to represent the output then determines the form of the cross-entropy function [34].

The simplest output units are linear units, which have no non-linearity involved. These are often just called linear units. Given features h, a layer of linear output units produces a vector $\hat{y}$:

$$\hat{y} = W^T h + b \tag{1.15}$$

Predicting the value of a binary variable y is a common task of neural networks. A classification problem with 2 classes can be seen as a binary classification problem. In this case, it is better to use Sigmoid Output Units, which is defined by:

$$\hat{y} = \sigma(W^T h + b) \tag{1.16}$$

Where, σ is the sigmoid function described previously.

The Softmax Unit, which employs the softmax function, is the most suitable output unit if the neural network's goal is to estimate a value over n potential values, also known as multiclass classification. This can be seen as a generalization of the sigmoid function. When representing the probability

distribution across n separate classes as the output of a classifier, softmax functions are most frequently utilized. Formally, the softmax function is given by:

$$softmax(z)_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \tag{1.17}$$

Where, Z_i are the possible categories.

Often, in deep learning log softmax is preferred over classic softmax. Log softmax is advantageous over softmax for improved numerical performance and gradient optimization. Log softmax is the log of the softmax function, mathematically it is defined as:

$$log\ softmax(z)_i = z_i - \log \sum_j \exp(z_j) \tag{1.18}$$

1.4.4 Back-Propagation

The Feedforward network accepts input X in the input layer and produces output $\hat{Y}$ through the output layer. Information flows forward through the network: from the input layer to hidden layers, after propagating through each hidden layer information, in the end, reaches the output layer. This mechanism is called Forward Propagation. On the other hand, when the information flows backwards toward the input layer known as Back Propagation. In this case, the flow direction is the opposite, information flow from the output units to the input units. The back-propagation algorithm (Rumelhart et al., 1986a) allows the information from the cost to then flow backwards through the network to compute the gradient [34]. The purpose of the backpropagation is to compute the derivative of the error/cost function with respect to the weights [34]. Often, the term backpropagation gets misunderstood as the whole learning algorithm. But in practice, the backpropagation algorithm refers only to the method for computing the gradient, while another algorithm, such as stochastic gradient descent, is used to perform learning using this gradient. Furthermore, back-propagation is often misunderstood as being specific to multi-layer neural networks, but in principle, it can compute derivatives of any function. In learning algorithms, the most often required gradient is the gradient of the cost function with respect to the parameters, which allows to update the parameters of the network during training iterations. The idea of computing derivatives by propagating information through a network is very general and can be used to compute values such as the Jacobian of a function f with multiple outputs.

The basic idea of the back propagation is very simple. The whole algorithm can be divided into three main steps:

1. **Feedforward propagation:** accepts input X, pass through intermediate stages (hidden layers) and obtain output.
2. **Compute error:** uses output to compute a scalar cost depending on the loss function.
3. **Back propagation:** allows information to flow backward from cost to compute the gradient.

In the next page, these steps will be explained in detail with a practical example.

1.4.4.1 Step 1: Feedforward operation, from input to output

Suppose we have a feedforward network with 1 input layer, 1 hidden layer and 1 output layer. The structure is following:

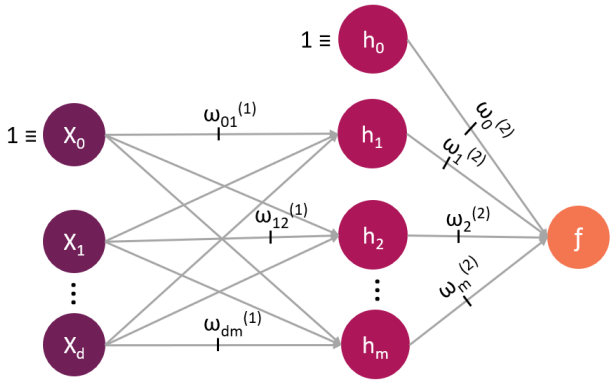


Figure 1.23: Feedforward network, x_0 and h_0 are biases with value $= 1$, f is the prediction function.

By applying the chain rule, the output function will be:

$$\hat{y}(X;W) = f\left(\sum_{j=1}^{m} w_j^{(2)} \cdot h\left(\sum_{i=1}^{d} w_{ij}^{(1)} \cdot x_i + w_{0j}^{(1)}\right) + w_0^{(2)}\right) \tag{1.19}$$

1.4.4.2 Step 2: Compute error

In neural network different error functions could be used. But here for simplicity, the root mean square is used:

$$L(X;w) = \sum_{i=1}^{N} \frac{1}{2}\left(y_i - \hat{y}(x_i;w)\right)^2 \tag{1.20}$$

The error gradients of the last layer can be found using partial differentiation.

$$\frac{\partial L(x_i)}{\partial w_j} = (\hat{y}_i - y_i)\, x_{ij} \tag{1.21}$$

1.4.4.3 Step 3: Back propagation

As anticipated earlier, each unit or node of the neural network has an activation function and suppose it is named with h. A general unit activation in a multi-layer network is made as follows:

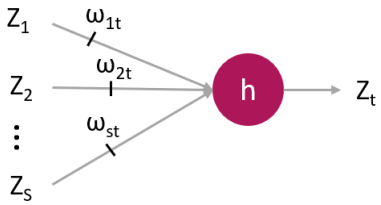


Figure 1.24: General unit activation, h is the activation function.

The output of the unit can be calculated as:

$$z_t = h\left(\sum_j w_{jt} \cdot z_j\right) \tag{1.22}$$

Forward propagation calculate for each unit:

$$a_t = \sum_j w_{jt} \cdot z_j \tag{1.23}$$

The loss L depends on w_{jt} only through a_t. The gradient for a parameter linking a particular error signal and input signal in a layer is a product of the input signal and the error signal of the layer:

$$\frac{\partial L}{\partial w_{jt}} = \frac{\partial L}{\partial a_t} \cdot \frac{\partial a_t}{\partial w_{jt}} = \frac{\partial L}{\partial a_t} \cdot z_j \tag{1.24}$$

$\frac{\partial L}{\partial w_{jt}}$ is the error gradient with respect to the weight w_jt of a single unit and this gradient is crucial for the weight adjusting/updating. Now let's see how we can calculate this gradient:

$$\frac{\partial L}{\partial w_{jt}} = \underbrace{\frac{\partial L}{\partial a_t}}_{\delta_t} \cdot z_j \tag{1.25}$$

First, lets annotate $\frac{\partial L}{\partial a_t}$ as δ. The calculation of the gradient depends on the type of the unit. If the unit is a output unit, then it is very simple: it is the error/loss:

$$\delta_t = \hat{y} - y \tag{1.26}$$

On the other hand, if the unit is a hidden unit, the gradient calculation is a little bit more complex. Suppose the hidden unit is t and it sends output to units S:

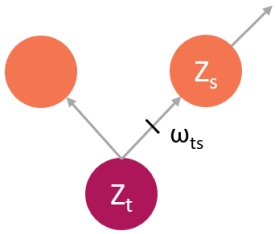


Figure 1.25: z_t hidden unit: z_s next layer unit

Then the δ_t can be calculated as following:

$$\delta_t = \sum_{s \in S} \frac{\partial L}{\partial a_s} \cdot \frac{\partial a_s}{\partial a_t} \tag{1.27}$$

$$= \underbrace{h^{'}(a_t)}_{\substack{\text{derivation of} \\ \text{activation function}}} \cdot \underbrace{\sum_{s \in S} w_{ts} \delta_s}_{\text{next layer}} \tag{1.28}$$

Note that, in this case, δ_t value depends on the derivation of the activation function and the next layer unit values. For that reason, the activation function should be chosen appropriately, such that the computational cost for the calculation of the derivative is acceptable.

This is how error gradients are calculated with respect to each parameter. Then using these valuable gradients the training algorithm updates the value of each parameter, which usually helps to decrease the training error.

1.4.5 Regularization

Neural networks (NN), also known as Artificial Neural Networks (ANN) are designed to mimic human brain functionalities and solve decision-making problems using data. A neural network should not only be able to make decisions on training data but especially on new and unseen data. Typically a neural network is considered a good network if it is able to perform well with new data, and this capacity is known as generalization. Generalization is the ability to perform well on previously unobserved inputs.

In ML and DL overfitting and underfitting are two very important terms. A neural network is called overfitted if its performance is good on training data, but it performs poorly with new data. On the other hand, a NN is called under fitted if its performance is very low on both training data and new data. In deep learning overfitting and underfitting can be controlled by altering the model capacity: the number of model parameters or weights. A model with low capacity can result in underfitting and a model with high capacity can lead to overfitting. So, finding a good balance is essential.

There is another important term in DL, known as regularization. Regularization is if any modification is made to a learning algorithm that is intended to reduce its generalization error but not its training error. In deep learning, the best model (the one that minimize the generalization error) is a large model that has been regularized appropriately. There are different type of regularization, some most used regularization techniques are:

1.4.5.1 Parameter norm penalties

This regularization method limits the model capacity by adding a parameter norm penalty to the objective function.

$$\tilde{L}(\theta) = L(\theta) + \lambda\,\Omega(\theta) = \sum_{i=1}^{N} L\left(f(x_i, \theta), y_i\right) + \lambda\,\Omega(\theta) \tag{1.29}$$

Where, λ is a hyper parameter and has value range [0, ∞]. Ω is the parameter norm penalty, which penalizes only the weights of the affine transformation at each layer and leaves the biases unregularized. However, the most two used parameter regularization are L2 and L1 regularization.

L2 regularization, also known as Weight Decay, adds "squared magnitude" of coefficient as penalty term to the loss function. Here the highlighted part represents L2 regularization element [37].

$$\tilde{L}(\theta) = \sum_{i=1}^{N} L\left(f(x_i, \theta), y_i\right) + \frac{\lambda}{2} \sum_{l} \|w_l\|^2 \tag{1.30}$$

Where, w_l are the parameters of a specific layer and l is the total number of parameters present in that layer. Here, if lambda (λ) is equal to 0, then the loss will be unchanged. However, if lambda is very large then it will add too much weight and it will lead to under-fitting. Having said that it's important how lambda is chosen. This regularization technique works very well to avoid over-fitting issue.

Instead, L1 regularization, also called Lasso, adds "absolute value of magnitude" of coefficient as penalty term to the loss function [37].

$$\tilde{L}(\theta) = \sum_{i=1}^{N} L\left(f(x_i, \theta), y_i\right) + \frac{\lambda}{2} \sum_{l} \|w_l\|_1 \tag{1.31}$$

Again, if lambda is zero then it will get back to a traditional loss L and more lambda λ grows, more weights become 0, which can cause underfitting. Lasso shrinks the less important feature's coefficient to zero thus, removing some features altogether. So, this works well for feature selection in case the training data has a huge number of features.

1.4.5.2 Data augmentation

Best way to make a DL model generalize better is to train it on more data. But, it is not always easy to have a big dataset for training. So, it is necessary to search for alternatives. One of the most used solution is synthesized data, which generate new samples (x, y) just by transforming inputs. Almost all image classification datasets undergo some form of data augmentation application, and this method is considered as an essential stage during the training process of an image classification model. In fact, data augmentation is very effective for the object recognition problem. For image data augmentation the techniques are very simple, they are some image pre-processing techniques: - flipping image, - randomly crop image, - rotate the image with a random angle, - random noise adding, - contrast changing, etc. However, this technique should be applied with attention, not should be applied transformation that would change the class.

1.4.5.3 Early stopping

In machine learning and deep learning early stopping is used to avoid overfitting when training alearner with an iterative method, such as gradient descent. The idea is very simple, the model is evaluated on a holdout validation dataset after each epoch, during training. If the performance of the model on the validation dataset starts to degrade (e.g. loss begins to increase or accuracy begins to decrease), then the training process is stopped. The model at the time that training is stopped is then used and is known to have good generalization performance.

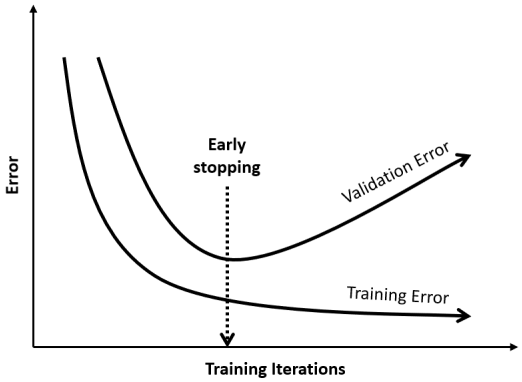


Figure 1.26: Early stopping

1.4.5.4 Dropout

Dropout is a regularization technique for reducing overfitting in neural networks. It prevents complex co-adaptations on training data. It is a very efficient way of performing model averaging with neural networks. The term "*dropout*" refers to dropping out units (both hidden and visible) in a neural network. Dropout is a technique where randomly selected neurons are ignored during training. They are "dropped-out" randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass. The effect is that the network becomes less sensitive to the specific weights of neurons. As a result, the network is better able to generalize and is less prone to overfit the training dataset. By turning off some neurons we force the network to learn better with only available neurons [38].

1.4.6 Optimization

As mentioned earlier, back propagation is a method for computing gradients with respect to weights. Stochastic gradient descent in conjunction with back propagation allow to update NN's weights in an efficient way. The gradient is the vector of partial derivatives with respect to the all coordinates of the weights. Each partial derivative measures how fast the loss changes in one direction. When the gradient is zero, i.e. all the partials derivatives are zero, the loss is not changing in any direction. However, in NN the optimization problem is non-convex, it probably has local minimum.

1.4.6.1 Vanilla gradient descent

Gradient descent is one of the most popular algorithms to perform optimization and by far the most common way to optimize neural networks. Vanilla gradient descent is the simplest version of the gradient descent algorithm. The algorithm is following:

```
while True do
    weights_grad = evaluate_gradient(loss_fun, data, weights)
    weights += − step_size * weights_grad                    // perform parameter update
end
```

Algorithm 1: Vanilla gradient descent

1.4.6.2 Batch gradient descent

In Batch Gradient Descent, all the training data is taken into consideration to take a single step. It takes the average of the gradients of all the training examples and then uses that mean gradient to update parameters. This algorithm requires a learning rate ϵ_k, where k is the iteration number. The parameter θ should be initiated with a value, typically randomly. The algorithm is following:

```
while stopping criteria not met do
    Compute gradient estimate over N examples:
    ĝ ← +(1/N) ∇_θ Σ_i L(f(x^(i); θ), y^i)
    Apply update: θ ← θ − ε ĝ
end
```

Algorithm 2: Batch gradient descent at iteration k

Here the gradient estimate is stable. But, this algorithm is very costly, in terms of computational time. Need to compute gradients over the entire training set for one update. This algorithm performs well with small dataset, like 1000 examples. But, what about if we have a data set with 5 millions entries? The training process definitely would be very slow and would require a lot of computational power. For that reason, usually stochastic gradient decent algorithm is preferred over batch gradient descent, which does not requires computing gradient over entire dataset.

1.4.6.3 Stochastic gradient descent

In Stochastic Gradient Descent (SGD), only one example is considered at a time to take a single step. This algorithm requires some parameters, which must be initialised before and they are learning rate ϵ_k, and initial network parameter θ. The algorithm is following: Since here only one example

```
while stopping criteria not met do
    Sample example (x^(i), y^(i)) from training set
    Compute gradient estimate over N examples:
    ĝ ← + ∇_θ L(f(x^(i); θ), y^i)        // no loss sum, only loss of single example
    Apply update: θ ← θ − ε ĝ
end
```

Algorithm 3: Stochastic gradient descent at iteration k

is considered at a time, the cost will fluctuate over the training examples and it will not necessarily decrease. But in the long run, the cost will decrease with fluctuations. However, there is some algorithm which is capable to optimize the SGD further, they allow to converge SGD algorithm faster. Some of modern optimization algorithms are: SGD with momentum, AdaGard and Adam. Among them, Adam is the most popular gradient-based optimization algorithm for training deep learning models.

1.4.7 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are specialized kinds of neural networks which mimic the visual cortex of the human. Our brain is structured in different sections to recognize objects and create body reactions. Each section has a specific task to solve. In the first levels of recognition (Retina - LGN - V1 - V2 - V4 - PIT - AIT) different levels of object detection can be found [34]. The first layers are used to detect edges, corners and simple shapes. Based on these detections more complex shapes can be recognized. Only at the last layers (AIT) real object representations are produced [34]. The same does also the CNNs, the first layers are responsible for recognizing simple features like edges, shapes and corners, and only in the last layer, the network is capable to recognize the whole object. As anticipated, it is very good to handle grid-like topology data, like images, which are 2-D grids of pixels. The CNNs are feedforward networks but with convolution. Convolution is a specialized kind of linear operation. *Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers* [34].

In its most general form, convolution is an operation on two functions of a real-valued argument. [34]. Let's see an example of two functions in one dimension. Suppose there is a car, which is tracked using a sensor. The sensor provides a single output $x(t)$, the position of the car at time t. Both $x$and t are both real values, so, we can get a different reading from the sensor at any instant in time. Now suppose that the sensor is somewhat noisy. To obtain a less noisy estimate of the car's position, we would like to average several measurements. Of course, more recent measurements are more relevant, so we will want this to be a weighted average that gives more weight to recent measurements. We can do this with a weighting function $w(a)$, where a is the age of measurement. If we apply such a weighted average operation at every moment, we obtain a new function s providing a smoothed estimate of the position of the car [34]:

$$s(t) = \int x(a)\, w(t-a)\, da \tag{1.32}$$

This operation is called convolution. The convolution operation is typically denoted with an *asterisk* [34]:

$$s(t) = (x * w)(t) \tag{1.33}$$

In convolutional network terminology, the first argument (in this example, the function x) to the convolution is often referred as the input, and the second argument (in this example, the function w) as the kernel. The output is sometimes referred as the feature map. Convolution is a general purpose filtering operation for images. A kernel matrix is applied to an image. It works by determining the value of a central pixel by adding the weighted values of all its neighbors together. The output of the convolution operation is a new modified filtered image $S(i, j)$, where i and j are 2-D dimension (pixel numbers) of the image. This filtering process can be explained mathematically by the following formula:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n) K(i-m, j-n) \tag{1.34}$$

Convolution operation is very useful for image processing, usually used for smoothing, sharpen or enhance image. However, different sized kernels contain different patterns of numbers and produce different results. The size of a kernel is arbitrary (Fig. 1.27 illustrates how the convolution operation is applied between a matrix and a kernel).

As said before, convolutional neural networks are multi-layer neural networks, but, with local connectivity. It means, neurons in a layer are only connected to a small region of the layer before, not with all neurons as happens with the fully connected NN. Moreover, in CNN weights are shared, the same kernel could be used by different neurons, which means fewer parameters to learn.

Besides the convolution operation, there are other two types of operations which are essential for a CNN network. The first one is non-linearity, where non-linear activation functions are applied.

Activation functions are chosen according to the needs, but among them, ReLU function is the most used. The second important operation in CNN is pooling. Pooling operation is used to reduce the spatial dimension. Convolutional layers in a CNN summarize the presence of features in an input image. A problem with the output feature maps is that they are sensitive to the location of the features in the input. For that reason, a non-linear down-sampling is needed to make it invariant to translation.

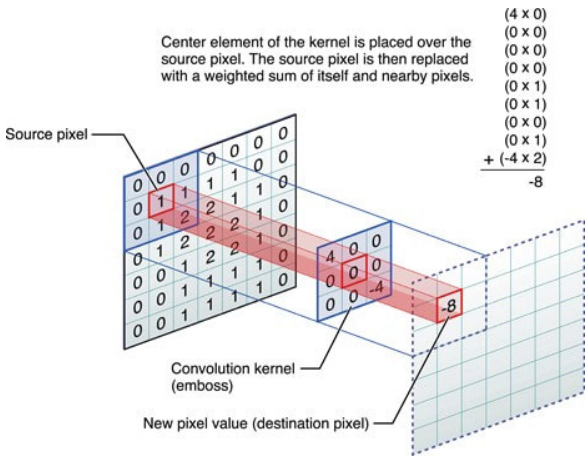


Figure 1.27: Convolution operation [39]

As a result, the down-sampled feature maps are more robust to changes in the feature's position in the image. Two common pooling methods are average pooling and max pooling which summarize the average presence of a feature and the most activated presence of a feature, respectively. Now let's see briefly some popular CNN architectures.

The very first convolutional neural network is LeNet5, which was invented in 1994 by Yann LeCun. This network was developed to recognize handwritten and machine-printed characters. It is a simple multi-layer convolution neural network for image classification. The network is known as Lenet-5 since it contains five layers with learnable parameters. It has three sets of convolution layers with a combination of average pooling. After the convolution and average pooling layers, it has two fully connected layers. At last, a Softmax classifier which classifies the images into respective classes.

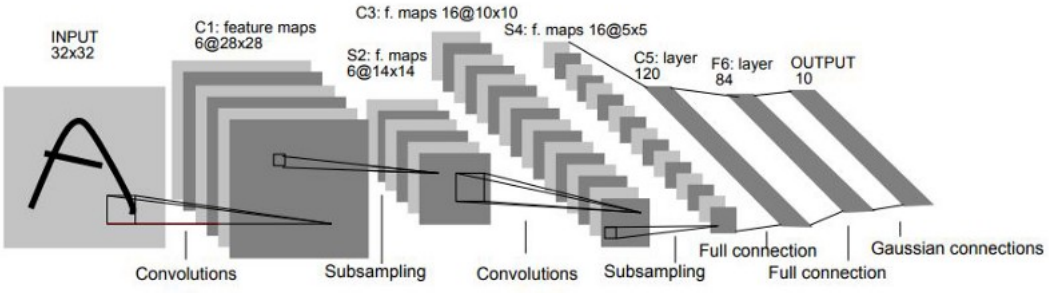


Figure 1.28: LeNet-5 Architecture [40]

In 2012, Alex Krizhevsky invented AlexNet which won the ImageNet challenge. In this challenge researchers competed with each other to achieve higher accuracy on several visual recognition tasks, using the ImageNet dataset. ImageNet dataset contains over 14 million images belonging to 1000 classes. Alex's network was deeper and at the same time much wider than LeNet, which introduced a very important concept to the deep learning community. A much larger neural network with a high number of parameters can learn and handle much more complex objects and object hierarchies. AlexNet has in total 8 layers, the first five were convolutional layers, some of them followed by max-pooling layers, and the last three were fully connected layers [41]. It used the non-saturating *ReLU* activation function, which showed improved training performance over *tanh* and *sigmoid* [41].

In 2014, two researchers from Oxford (Karen Simonyan and Andrew Zisserman) published a paper with the title *"Very Deep Convolutional Networks for Large-Scale Image Recognition"*, where they introduced a new type of CNN architecture called VGG. VGG stands for Visual Geometry Group. This architecture achieved top-5 test accuracy of 92.7% in ImageNet [42]. It is one of the famous architectures in the deep learning field. VGG uses a small-sized kernel instead of large-sized kernels like AlexNet. This network was able to improve accuracy by replacing AlexNet's large kernel-sized filters with 11 and 5 in the first and second layer respectively, with multiple 3×3 kernel-sized filters

one after another [42]. VGG has two different versions: VGG16 and VGG19, which has a different number of layers, but, among them, *VGG16* is the most used.

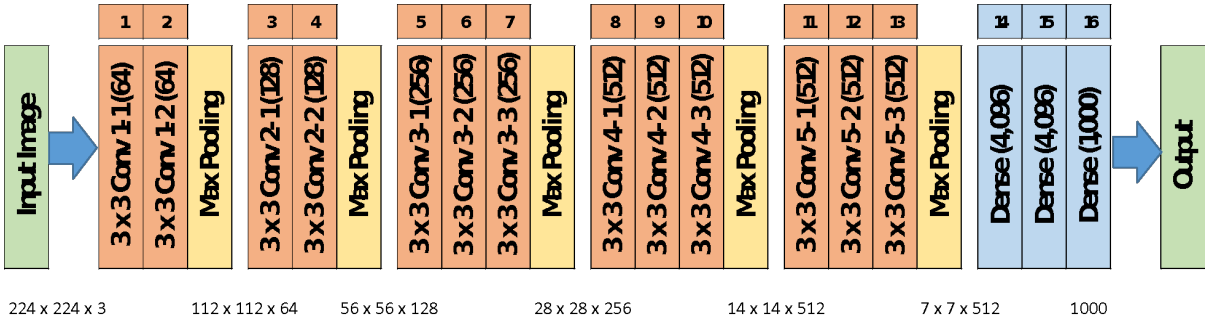


Figure 1.29: VGG-16 Architecture [42]

In 2015, there was another feedforward convolutional neural network which surprised the deep learning community by winning the ImageNet competition. The name of that network was ResNet, it was invented by Shaoqing Ren, Kaiming He, Jian Sun, and Xiangyu Zhang. ResNet stands for Residual Neural Network. This network was designed to solve the vanishing gradient problem suffered by deeper networks. During that time, the research community was following the trend of the deeper network, more large and deeper the network is more it can learn complex objects. In fact, since AlexNet, the state-of-the-art CNN architecture is going deeper and deeper. While AlexNet had only 5 convolutional layers, the VGG network and GoogleNet (CNN developed by Google in 2014) had 19 and 22 layers respectively [43]. But, increasing network depth does not work by simply stacking layers together. Deep networks are hard to train because of the notorious vanishing gradient problem — as the gradient is back-propagated to earlier layers, repeated multiplication may make the gradient infinitely small [43]. As a result, as the network goes deeper, its performance gets saturated or even starts degrading rapidly [43]. ResNet is able to tackle this problem by using *identity shortcut connections* (Fig. 1.30). These shortcut connections allow to bypass one or more layers and add extra information to the next layer, which helps to tackle the vanishing gradient problem. Also, ResNet has different versions, in which the number of the layer varies and consequently also the number of parameters. Among them *Resnet-18* and *ResNet-50* are the most popular.

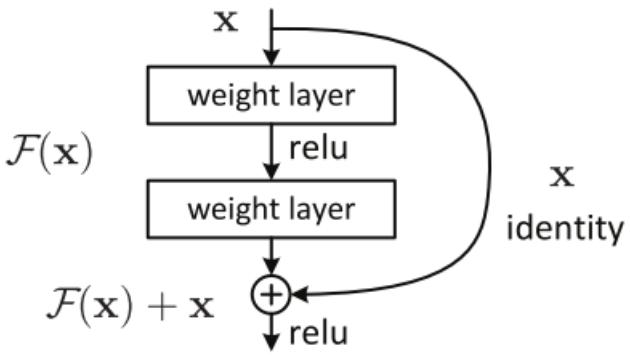


Figure 1.30: Identity shortcut connection [43]

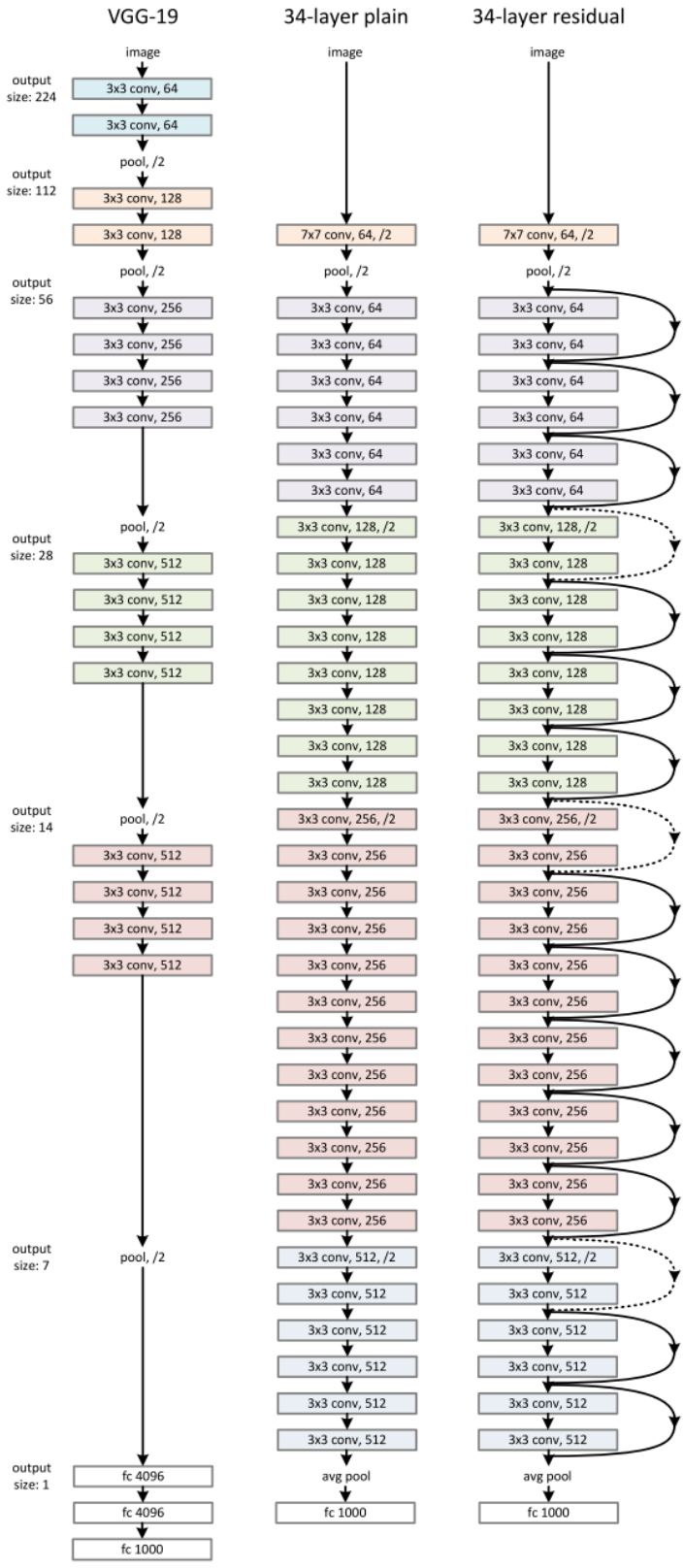


Figure 1.31: ResNet architecture vs VGG architecture [43]

Figure 1.31 illustrates the key differences between the ResNet and the VGG network. ResNet has skip connections between layers which allow information to flow directly to the next layer or the deeper layer. These skip connections are very useful when the gradient starts becoming smaller during the back-propagation, they add extra information and prevent the gradient from vanishing.

Chapter 2
Dataset

For this research study, the Italian COVID-19 Lung Ultrasound DataBase (ICLUS-DB) is used. It was introduced by Italian researchers in a scientific research paper [12] in 2020. It contains a total of 277 lung ultrasound (LUS) videos from 35 patients, corresponding to 58,924 frames. Among them, 45,560 frames were acquired with the convex probe and 13,364 frames with the linear probe. The data were acquired within different medical centers. In particular, 1,313 (1,305 convex and 8 linear) video frames were collected from Fondazione Policlinico Universitario San Matteo IRCCS, Pavia, Italy. 7,401 video frames (1,657 convex and 607 linear) were collected from Fondazione Policlinico Universitario A. Gemelli IRCCS, Rome, Italy. 5,892 (3,020 convex and 362 linear) video frames were collected from Fondazione Policlinico Universitario A Valle del Serchio General Hospital, Lucca, Italy. 42,048 video frames (33,645 convex and 8,402 linear) were collected from Brescia Med, Brescia, Italy whereas the remaining 2,271 video frames (all acquired with linear probes) were acquired at Tione General Hospital, Tione (TN), Italy. Among the 35 patients, 17 of them (49%) were confirmed COVID-19 positive, 4 patients (11%) as COVID-19 suspected and 14 patients (40%) as confirmed asymptomatic.

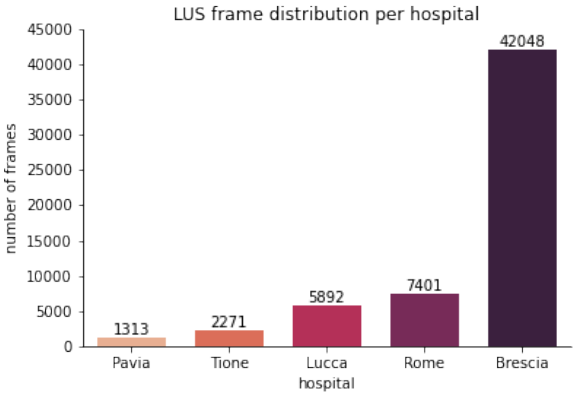


Figure 2.1: Frames distribution per hospital

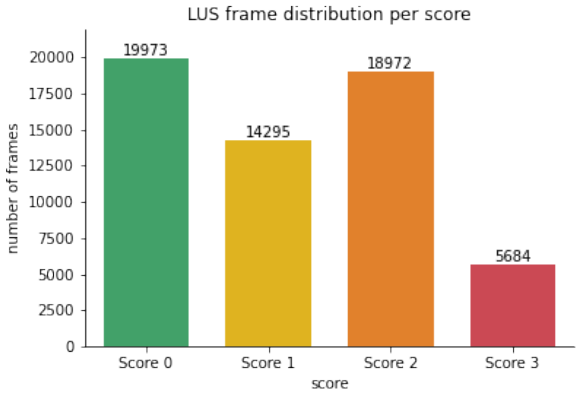
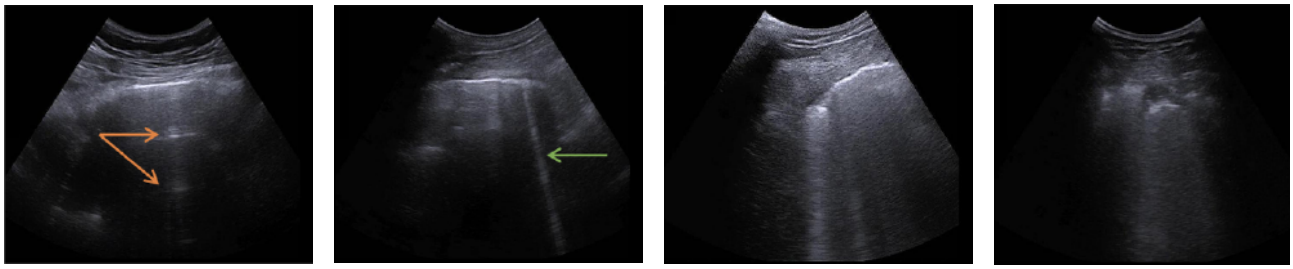


Figure 2.2: Frames distribution per score

In 2020 Soldati et al. proposed a scoring system for severity classification [10]. In this research paper, they described how specific imaging biomarkers in LUS can be used in the management of COVID-19 patients. Specifically, a 4-level scoring system was proposed to evaluate the progression of the pathology. This scoring system allows to classify LUS frames by the severity of the pathology. The proposed scoring method has a score range from 0 to 3.

- **Score 0:** indicates a continuous and regular pleural line in conjunction with the presence of horizontal artifacts. These artifacts are formed due to ultrasound waves reverberation between the (highly reflective) normally aerated lung surface and the probe. These reverberations are visualized as multiple equidistant replicas of the pleural line. These artifacts are also referred to as A-lines. Orange arrows in Figure 2.3a indicate horizontal artifacts.
- **Score 1:** marks the appearance of the first signs of abnormality. The pleural line is indented and vertical artifacts are visible. These artifacts are formed due to location deaeration along the lung surface lung, as volumes of lung previously occupied by air are replaced by media favoring the transmission of acoustic signals. Green arrow in Figure 2.3b indicates vertical artifact.
- **Score 2:** indicates a broken pleural line. Below the breaking point, consolidations (dark areas) are visible in association with white areas (white lungs) below them. The dark areas represent



(a) Score 0 (b) Score 1 (c) Score 2 (d) Score 3

Figure 2.3: LUS scoring system

deaeration followed by the replacement of air with media having similar acoustic properties as soft tissues. White areas beyond consolidations represent partial deaeration causing the multiple scattering of transmitted waves.

- **Score 3:** indicates the extension of vertical artifacts or white lung across the scan, with or without consolidations, over an area which extends over more than 50% of the pleural line.

All 58,924 frames were labelled according to the scoring system defined above. A total of 19,973 video frames (34%) were labelled as score 0, 14,295 video frames (24%) labelled as score 1, 18,972 video frames (32%) labelled as score 2 and 5,684 video frames (10%) were labelled as score 3. A plot representing the distribution of the scores is shown in Figure 2.2. To ensure accurate interpretation the labelling process was divided into four stages. In the first stage, frame-level scores were assigned by master students, which were then validated by PhD expert from LUS background in the second stage. In the third stage, the scored frames were further verified by a biomedical engineer (with more than 10-year experience in LUS). All data were acquired using a variety of ultrasound scanners (MindrayDC-70 Exp®, EsaoteMyLabAlpha®, ToshibaAplio XV®, WiFi Ultrasound Probes - ATL) with both convex and linear probes.

During the LUS data acquisition, noise in the form of metadata, scanner information, and imaging parameters is recorded as part of the image. This redundant information is required to be filtered out before feeding the data to an AI-based model to avoid ambiguous output states of the system. This may include the textual information of the echo machine setting, measurement lines, arrows identifying the focus into the field of view of the acquisition etc. The presence of such kind of information in the LUS scans may lead the networks to learn the ambiguous interpretation of regions causing anomalous prediction as pointed out in the study [44]. To avoid such behavior, pre-processing techniques are applied to filter the COVID-19 patient data from the redundant information, preparing them for the latter steps. Pre-processing is carried out in the form of image cropping using a MATLAB toolbox. As a result, LUS videos were centred and cropped. This resulted in the extraction of the field of view (FOV) preserving the spatial resolution of each pixel and removing noise.

Before starting the training step, data augmentation is performed on the dataset. Data augmentation is crucial to increase the model's generalization capacity. The augmentation technique at frame-level classification is based on augmentation functions including affine transformations (such as translation (max. $\pm 15\%$), rotation (max. $\pm 15°$), scaling (max. $\pm 45\%$), and shearing (max. $\pm 4.5°$)), horizontal flipping (p = 0.5), multiplication with a constant (max. $\pm 45\%$), contrast distortion (max. $\pm 45\%$), Gaussian blurring ($\sigma_{max} = 3/4$), and additive white Gaussian noise ($\sigma_{max} = 0.015$).

Chapter 3
Methodologies

In this chapter, all methodologies are presented. The first section explains the *Frame-level scoring system*. The scoring system is the core of this research study, which aims to identify the above-mentioned pathological patterns in the LUS frames and classify them to their corresponding scores. Different state-of-the-art convolutional neural networks (CNNs) have been employed in this section. Second section includes the *Grad-CAM* algorithm, which has the ability to highlight the LUS frame pixels that are responsible for the score prediction. In addition, the third section explains how the generalization capability of deep neural network has been assessed across different medical centers. The last section of this chapter describes the training environment, including the use of a Linux cluster and Singularity container.

3.1 Frame-level scoring system

The frame-level scoring system aims to estimate the severity of the pathology present in a LUS frame of a COVID-19 patient. It takes a LUS frame as input and predicts a score $\in \{0, 1, 2, 3\}$ as output, where, the score represents the level of severity. Score 0 indicates the presence of a continuous pleural-line accompanied by horizontal artifacts, which characterize a healthy lung surface [10]. In contrast, score 1 indicates the first signs of abnormality. Scores 2 and 3 are representative of a more advanced pathological state.

Several methods were proposed related to frame-level score prediction. However, most of them are based on custom and complex techniques of convolutional neural network. For instance, Roy et al in [12] proposed a cascaded model with Spatial Transformer Network followed by CNN to categorize the LUS video frames into four possible classes. They utilized STN to predict and apply transformations on the input image to generate their class-based salient cropped versions, and classify localized cropped input frames using CNN. But in this framework, only state-of-the-art architectures like ResNet, DenseNet, and Inception are used for frame-level score prediction. They are used as the backbone of the network when trained and tested with the ICLUS-DB dataset.

To split the dataset patient level split is followed. The data is splitted according to the ratio proposed in [12], where 60% data is used for training and the remaining 40% data for testing. According some studies [45][10], *patient level* splitting is the most suitable splitting method for LUS data. Patient level splitting avoids any overlap among the patients in the training and testing data. For instance, if we randomly select frames for creating a training and testing split, it is very likely to have overlap among patients in the splits. This could happen also if we adapt Exam level split (split data by exams) or Video level split (split data by videos). Both do not avoid patient overlapping, which arises the problem of information leakage adding bias to the evaluation. To avoid this leakage between the train and test data, the split at the Patient level is most suitable, which ensures no overlap between patients.

ResNet, DenseNet, and Inception architecture come with different variants. These variants are characterized by the different number of parameters and number of layers. Here in particular ResNet-18, ResNet-50, ResNet-101, DensNet-121, DensNet-201, and InceptionV3 are utilized. To prevent overfitting, dropout layers are added to the end of the network, followed by fully connected layers. Moreover, a softmax layer is attached for multi-class classification. During the training stage, these state-of-the-art models are trained and evaluated over the dataset and their performances are recorded. All of the models are trained by back-propagation of errors in batches of size 4, with images resized to 224 x 224 pixels for 50 epochs with a learning rate of 10^{-4}. These models are trained using Stochastic Gradient

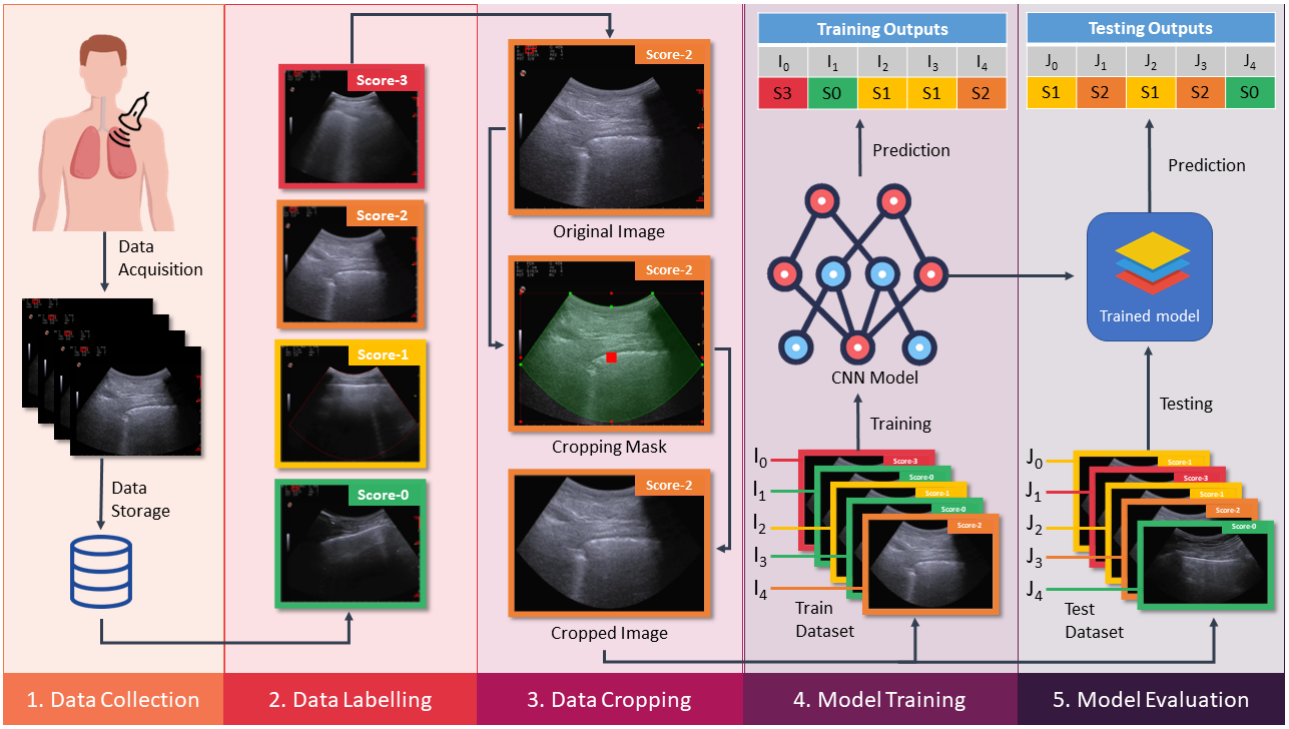


Figure 3.1: Frame-level scoring system development workflow

Descent (SGD) as the optimizer with a weight decay of 10^{-6}. Transformations of random horizontal flip, rotation, translation, sheering of images and gaussian blur are added to each batch during training as data augmentation in order to enable an improved and generalized network learning.

To extend further the experiment, transfer learning is applied to the best-performing model. Transfer learning helps developers to take a blended approach from different models to fine-tune a solution to a specific problem. Most of the time transfer learning is preferred over training from scratch to accelerate the learning process. In order to apply transfer learning, pre-trained weights from the ImageNet dataset are loaded into the best model. Then, new layers are attached towards the end of the network, with randomly initialized parameters. Then, the model is trained using the same hyper-parameters as used to train the models from scratch. However, to further improve the performance, these models are fine-tuned using Adam as an optimizer with a relatively smaller learning rate of 10^{-5}.

From an algorithmic point of view, the prediction task is performed using the input data to obtain the best prediction at the outcome. The model takes one image at a time and predicts an output class. To do so, the model should be trained appropriately such that it can recognize the patterns related to the target classes. The learning process of a deep neural network could be supervised, semi-supervised and unsupervised. Since in this framework the training procedure employs labelled data to build models, the learning process is considered supervised. In fact, all the LUS frames of the ICLUS-DB dataset are associated with a specific score or class. These associated classes are called ground truth (denoted with y). During a supervised learning process, the model is trained with a training dataset for k times, where k is the number of epochs (i.e. iteration). In each epoch, the model predicts a set of probabilities for each training image, which represents the prediction confidence of the model for each possible output class. High probability means the model is quite confident about the relation between the class and the input image. Whereas, low probability means low confidence in input-class relation. Therefore, the class with the highest probability is selected as the final prediction (denoted with $\hat{y}$). Once the model predicts classes for all training images, these predictions get compared with the ground truth and the loss gets calculated. Loss is very useful for the back-propagation algorithm. The back-propagation algorithm fine-tunes the weights of the neural network based on the loss (i.e. error rate) obtained in the previous epoch. Proper tuning of the weights ensures lower error rates,

making the model reliable by increasing its generalization. And this process gets repeated for k epochs.

However, the whole development process of this scoring system can be summarized into 5 main steps. They are: 1) *data collection*, 2) *data labelling* 3) *data cropping*, 4) *model training* 5) *model evaluation*. The first three steps were discussed in the Dataset chapter. On the other hand, the last two steps (model training and model evaluation) were explained above in this chapter. The diagram in Figure 3.1 illustrates all the five steps.

3.2 Explainable AI with Grad-CAM

Explainable AI is Artificial Intelligence (AI) in which humans can understand the decisions or predictions made by the AI [46]. The majority of neural networks are enormous in size and complicated, which makes it challenging for humans to comprehend their behavior. Gradient weighted Class Activation Mapping (Grad-CAM) is a very useful tool for understanding the behaviour of a deep neural network, especially of a convolutional neural network (CNN). It makes it possible to comprehend spatially which zone of the image is accountable for a target class prediction. Moreover, it allows building transparent models with the ability to explain *why they predict a certain class*. Besides that, transparency is very helpful for determining the model's reliability and deployability. For that reason, Grad-CAM has been integrated into this framework for evaluating further the behaviour of the best-performing model. It is an additional inspection tool to verify that the network learns patterns pertinent to the score and not any unrelated artifacts.

Grad-CAM uses gradients of a particular target class that flows through the convolutional network to localize and highlight regions of the target in the image. The intuition behind the algorithm is based upon the fact that the model must have seen some pixels (or regions of the image) and decided on what object is present in the image [47]. From a mathematical point of view, influence can be described with gradient [47]. Grad-CAM algorithm starts with finding the gradient of the most dominant logit (probability) with respect to the latest activation map in the model (layer preceding the output layer), which corresponds to the activation of different parts of the image. We can interpret this as some encoded features that ended up activated in the final activation map persuaded the model as a whole to choose that particular logit (subsequently the corresponding class). The gradients are then pooled channel-wise, and the activation channels are weighted with the corresponding gradients, yielding the collection of weighted activation channels [47]. By inspecting these channels, it is possible to tell which ones played the most significant role in the decision of the class. As the final result Grad-CAM algorithm produces an output mask which highlights the part of the image responsible for the class prediction.

3.3 Generalization capability of the network

A neural network is considered to have a strong generalization ability when it can tackle novel data by providing reliable forecasts. Here, the analysis has been extended to evaluate the generalization capability of the top-performing model, across medical centers. The goal of this experiment is to evaluate the capacity of the best-performing architecture (trained from scratch model and pre-trained model) to tackle novel data from a medical center while trained with data from another medical center.

In ICLUS-DB dataset, Brescia constitutes more than 70% of the overall dataset (Fig. 2.1). For this reason, LUS data from Brescia was considered for training, while Rome having the second largest contribution to the dataset, is considered for testing. Another outcome of this analysis is to examine how much data is required to achieve comparable performance. LUS frames from Brescia are collected from 13 patients thus the training is performed in three splits based on the number of patients. The first split comprises 1 patient of Brescia in training and the rest in validation, the second split considers half of the patients in train and half in validation. The third split considers 12 patients in training leaving 1 for validation. To keep the training process fair, data shuffling is performed at each split, ensuring that the patient in train also becomes part of the validation during the training process.

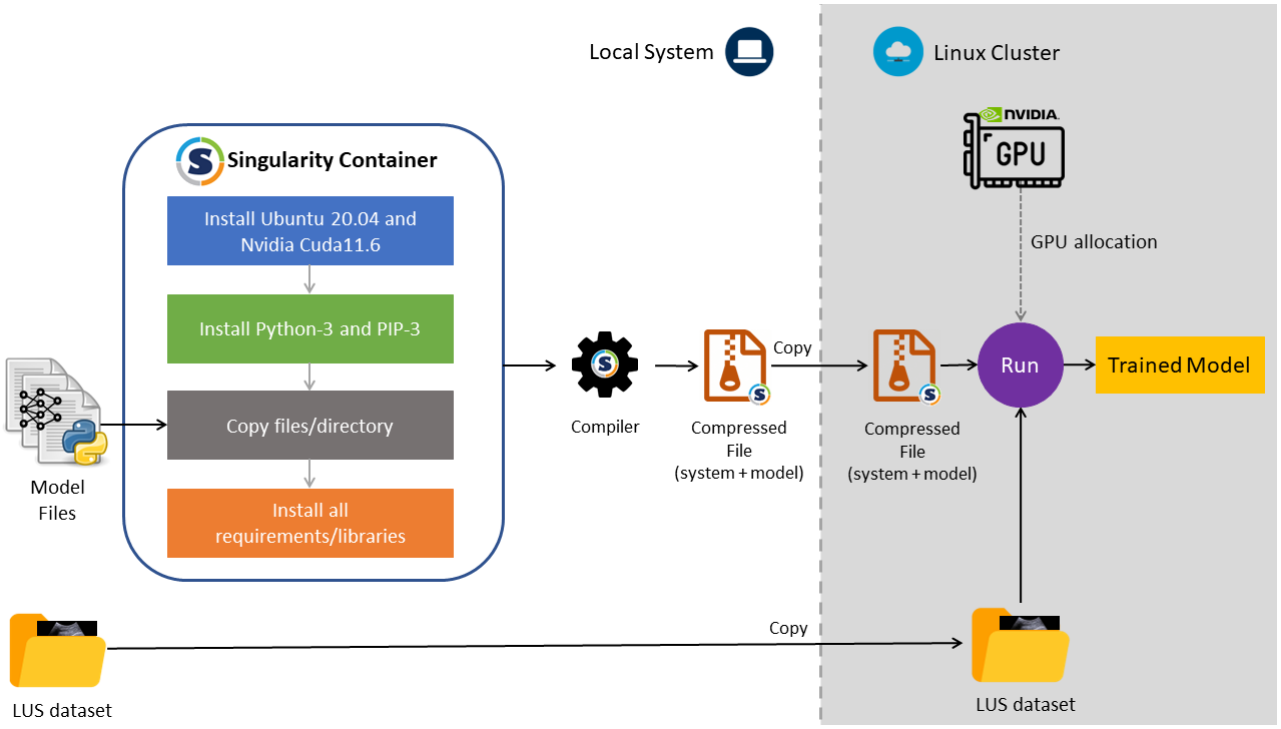


Figure 3.2: Model training workflow with Singularity container

3.4 Training setup

All the models are trained on a Linux cluster of the University of Trento, called *deeplearning cluster*. In total this cluster has 8 GPUs. They all have the same model- *A100 Tensor Core* model of Nvidia, with 40GB dedicated memory. All GPUs support CUDA, which is essential for neural network training. CUDA is a hardware/software solution for parallel computing on Nvidia GPUs. It supplies compilers and libraries to develop custom C/C++ applications. A high number of languages support CUDA through libraries (such as python).

However, in a Linux cluster system, the use of a container is fundamental. It makes sure that no incompatibility happens between cluster users. Usually, a cluster has a big number of users. Each user could have different requirements, need a different system set up and use different libraries. So, parallel usage of the system's resources can produce internal collisions. To avoid incompatibility, a container system called *Singularity* is used during the model training. All required software and dependencies are installed inside this container. Then, the container is deployed from the local system to the cluster via SSH protocol. The workflow of the singularity containerization is very simple, it has a few key steps as follows:

1. Create the container locally
2. Download and install an operating system
3. Install all needed software (e.g. python, pip)
4. Copy files/folder system
5. Install required libraries
6. Compile and compress the container.
7. Copy the compressed container to the cluster
8. Run it with necessary setup parameters (e.g. GPU allocation)

Figure 3.2 illustrates graphically the whole workflow. However, the best part of Singularity is that it does not need admin privileges while working on a cluster unlike other containers, such as Docker. In other words, Singularity facilitates a mobile workspace of software. In this way, Singularity containers are incredibly useful to the open-source community by contributing to the reproducibility of workflows and tools. For this study, each model was trained separately, one by one. During the training process, one or more GPUs can be allocated. For this work, only one GPU with 40GB was used for model training.

Chapter 4

Experimental results

4.1 Frame-level scoring system

In this section, the results obtained from the frame-level scoring system are analyzed. Here, different types of the deep neural network are considered. To gauge the performance of the state-of-the-art CNN models on the ICLUS-DB dataset, the agreement between the frame-level predicted score $\hat{y}$ by the models and the frame-level score as the ground truth y needs to be measured. To do so, performance indicators are quite useful to evaluate and compare different classification models and techniques. In this study, the model's performance has been evaluated based on the metric already used in the previous studies [12] [48] i.e, F1-Score; assessing classification performance based on the weighted average of precision and recall.

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.3}$$

Where TP, FP and FN are *true positive*, *false positive* and *false negative*, respectively.

These models were evaluated using the train/test split used in [12] and [48] for bench-marking and comparison purposes. Classification results of these models are represented in Table 4.1. As per the evaluation, ResNet-18 performed better than the rest of the models with an F1-Score of 0.659 followed by ResNet-50 with 0.651. It is worth noticing that among the variants of ResNet, DenseNets, and Inception, all of the ResNet models performed better than the rest with an average of 0.653 F1-Score.

With ResNet-18 turning out to be the best-performing model, the evaluation is extended towards analysing the model with pre-trained weights from the ImageNet dataset, loaded in the convolutional layers of the network. Training and fine-tuning the model over the same dataset and with the same train/test split, resulted in giving a comparable performance with 0.645 F1-Score.

For developing a comparative study and for benchmarking purposes different models proposed in the existing studies were compared with the models developed for this study. Roy et al in [12] proposed a cascaded model with Spatial Transformer Network followed by CNN to categorize the LUS video frames into four possible classes. They utilized STN to predict and apply transformations on the input image to generate their class-based salient cropped versions and classify localized cropped input frames using CNN. They also employed data augmentation techniques to introduce generalization to their models. They also utilized the initial version of ICLUS-DB with their proposed train/test split. As a result, their proposed model predicted the input frames with an F1-Score 308 of 0.651.

Another baseline study, working on the same dataset, belongs to Frank et al [48], where they not only used the raw LUS frame as input data but also used additional information of vertical artifacts and pleural line. This multi-channel information is then fed to a set of pre-trained networks and is trained and fine-tuned resulting in an F1-Score of 0.688 when using all of the information channels as input to the pre-trained model, ResNet-18. In comparison to these baseline techniques, the best performing model in this study ResNet-18 gave a comparative F1-Score of 0.659 when trained from scratch while

giving an F1-Score of 0.645 when used under the transfer learning approach. This showed that with only raw LUS image frames as input these state-of-the-art DL models can perform equally well and in fact outperform them by a fair margin. Furthermore, it also implied that the complex network structures do have a significant impact in improving the prediction however, comparable performance can be achieved by a relatively less complex one.

Methodology	**Employed Model and Technique**	**F1-Score**
Baseline	Reg–STN+CNN [12]	0.651
	ResNet-18+Annotations [48]	0.688
Proposed	ResNet-18[a]	0.659
	ResNet-50	0.655
	ResNet-101	0.651
	DensNet-121	0.6513
	DensNet-201	0.6517
	InceptionV3	0.612
Pre-Trained	ResNet-18[b]	0.645

Table 4.1: Frame-level Performance of employed models and techniques

[a] The best performing CNN model when trained from scratch, [b] Pre-trained version of the best performing model

4.2 Grad-CAM

In this section results from the Grad-CAM algorithm are analyzed. As mentioned in the previous chapter, Grad-CAM highlights the parts of the image that are responsible for the class prediction made by a CNN. Grad-CAM was applied to the best model after all the models' performances had been assessed based on the F1-Score metric, in order to determine its interpretation capacity for the LUS image. In Figure 4.1 some examples are shown, where, from the left to right each column represents a prediction task with Score 0, Score 1 Score 2 and Score 3, respectively. The first row contains input images and the second row shows Grad-CAM heatmaps representing the model's neural activation. In the first column, it can be noticed that the model has predicted score 0 influenced by horizontal artifacts. According to the Grad-CAM image shone in the second column, it is evident that the model has predicted score 1 due to a vertical artifact (highlighted with red color in the Grad-CAM image). Additionally, because of the broken pleural line and consolidations present along the vertical artifacts, the model has predicted score 2 in the third column. The last column contains a LUS frame of score 3, where the model was successful in predicting the output class influenced by the extension of vertical artifacts. The model's ability to interpret LUS images is amply demonstrated by these four Grad-CAM images.

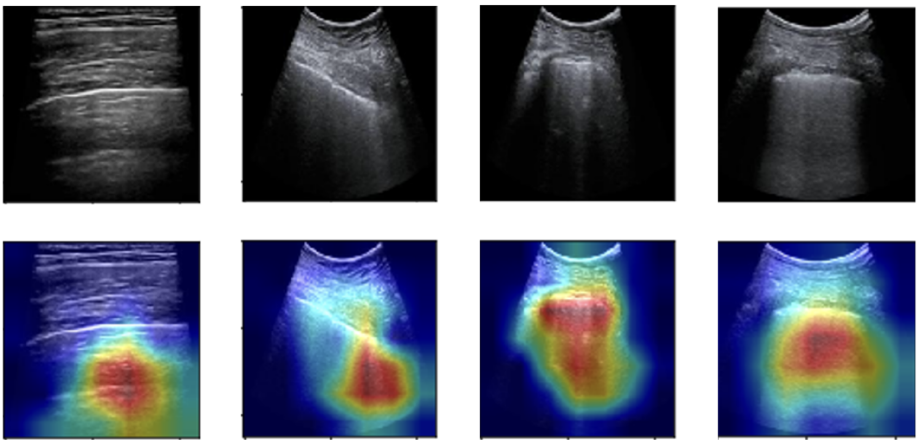


Figure 4.1: Gradient Class Activation Maps for each class (left to right representing Score 0 to Score 3)

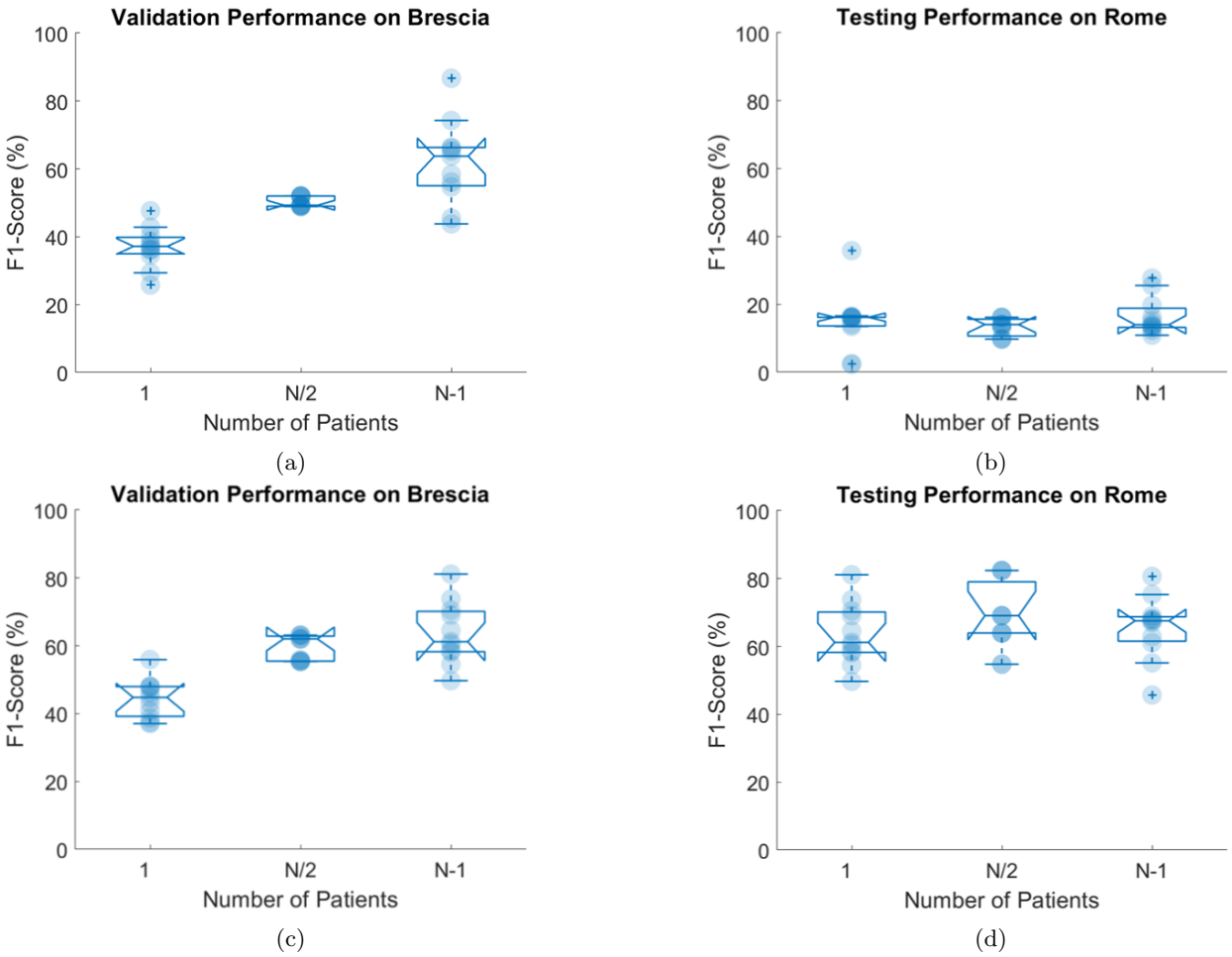


Figure 4.2: Analyzing generalization capability of both models of ResNet-18 (a) Represents the validation performance on LUS data from Brescia using ResNet-18^a (b) Represents the testing performance on data from Rome using ResNet-18^a. (c) Represents the validation performance on LUS data from Brescia using ResNet-18^b (b) Represents the testing performance on data from Rome using ResNet-18^b. The number of patients in training data from Brescia is shown to increase from 1, N/2, to N-1 in all of the horizontal axis. The vertical axis represents the F1-Score in (%) and each box in the boxplot represents the performance range spanning over the box length.

4.3 Generalization capability of the network

The generalization capability of ResNet-18 (the best model) is analyzed by training over data from one medical center and testing on data from another center. For this experiment ResNet-18^a (trained from scratch) and ResNet-18^b (pre-trained with ImageNet dataset) both are evaluated. As mentioned in the methodologies chapter, for this purpose, LUS data from Brescia is used for training and LUS data from Rome is used for testing. Doing so, results from ResNet-18^a and ResNet-18^b are collected. Firstly, the result from ResNet-18^a is analyzed. Validation and testing performance on the clinical center of Brescia and Rome are shown in Figures 4.2a and 4.2b respectively. According to the validation results obtained using Brescia data, the validation performance improved as the number of patients increased. However, no significant performance or improvement is seen across the three splits when testing performance is evaluated using data from Rome. These results suggest that no impact of the increase in the training data on the testing performance is reported, indicating signs of overfitting of the training model. Variations in the testing data from Rome, acquired from a different type of LUS probe, are causing false predictions because the data from Brescia was acquired from one type of LUS probe. As a result, poor generalization is observed across the different medical center when the model was trained from scratch. It also highlights the significance of including LUS acquired from similar probes in the training set as seen in the frame-level classification of LUS data from the ICLUS-DB dataset. Moving towards the second part of this analysis, Figure 4.2c and Figure 4.2d

show the validation and testing results using the ResNet-18^b. Figure 4.2c shows a similar trend of improving validation performance as the number of patients in the training data increases. However unlike in Figure 4.2b, here Figure 4.2d represents generalization of the model across all three splits. Furthermore, it is evident that as the number of patients in the training increases from 1 to N/2, the testing performance for data from Rome also improves. But when increasing the training data from N/2 to N-1 patients, no significant improvement is observed suggesting that the model reached the saturation point. The analysis demonstrates that the transfer-learning approach works well in situations where not only data is limited but is also has limited representative distribution in the training set. With all these findings, it can be verified from the results that when the training data does not possess variability across the distribution, the model fails to adapt towards the new data and thus shows poor generalization performance. However, this generalization is observed to hold when the model is used in the pre-trained setting and is fine-tuned to the target domain. Choosing the right model and approach, it can be seen from the results that the model trained for the maximum number of patients, showed the highest classification performance of 67% for the testing set. With the average classification performance of 61.3% for the maximum number of patients in training, the model was able to achieve a testing performance similar to the one achieved over the validation set i.e. 63%. This clearly demonstrated the network's ability to perform equally well with unseen data from another clinical center as it did with the data used for validation. Furthermore, compared to the frame-level F1-Scores obtained for the ICLUS-DB dataset, comparable performance is achieved using data from only 1-patient in training. This shows that by reducing the data from 58,924 samples acquired from multiple medical centers to an average of 20,000 samples obtained from one medical center, using the pre-trained model, state-of-the-art performance can be achieved.

Chapter 5
Conclusion

In this dissertation, state-of-the-art CNN models were analyzed while applied over LUS data. In the previous research studies, different techniques were presented to perform frame-level score prediction on LUS data of Covid-19 patients. These techniques showed that with the employment of combined networks and by introducing annotated information along with input, notable performance can be achieved. However in this work, outperforming results were achieved by using state-of-the-art CNN models as the backbone of the network with an image frame as the only input. Among the tested models, ResNet-18 was found to be the best-performing model with an F1-Score of 0.659. All models were evaluated over the ICLUD-DB database. In addition, the behaviour of the top model was assessed by applying Grad-CAM algorithm, which highlighted the parts of the input image that were responsible for the score prediction. This algorithm has helped to develop a transparent and explainable AI model. Moreover, it has made possible to understand whether or no the model has learned the right patterns and features related to the score prediction. Data availability has always been one of the major issues surrounding research in medical imaging and the lack of adequate data for training CNNs has always been a concern. To address this issue, frame-level scoring analysis was extended by examine ResNet-18 with pre-trained layers with weights from the ImageNet dataset. In this transfer-learning approach, an F1-Score of 0.645 was obtained falling short of the performance achieved when trained from randomly initialized weights. This suggested that with the amount of data available, training from scratch proves to be better than the transfer-learning approach. Furthermore, the generalization capability of the network has been evaluated across data from different medical centers. To do so, a pre-trained model of ResNet-18 was trained and fine-tuned over data from Brescia and tested over data from Rome. It was found that the generalization capability across the different medical centers holds and showed an increasing trend when the number of patients in the training samples were also increased, when pre-trained model was used. Whereas, despite of the increasing trend of performance for the validation set, the model tend to overfit for testing data from Rome. This suggested that not only the amount of data but also the variation with in the distribution is required when to train the model from scratch as it fails to generalize otherwise. While using the pre-trained version it was found that with an average F1-Score of 0.60 achieved with half of the patients in the training samples, state-of-the-art comparable performance was achieved. As future work, the application of transformer-based models for classification tasks and the transition from frame-level to video-level classification could be done.

Bibliography

[1] Johns Hopkins Medicine. Covid-19 lung damage. `https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus/what-coronavirus-does-to-the-lungs`.

[2] A. Parasher. Covid-19: Current understanding of its pathophysiology, clinical presentation and treatment. *Postgrad. medical journal*, 2021.

[3] P. Joshi R. Tyagi K. M. Agarwal. Technological resources for fighting covid-19 pandemic health issues. *J. Ind. Integration Manag.*, 2021.

[4] S. Woloshin N. Patel A. S. Kesselheim. False negative tests for sars-cov-2 infection—challenges and implications. *New Engl. J. Medicine*, 2020.

[5] Y. Yang et al. Laboratory diagnosis and monitoring the viral shedding of sars-cov-2 infection. *The innovation*, 2021.

[6] S. Salehi A. Abedi S. Balakrishnan A. Gholamrezanezhad. Coronavirus disease 2019 (covid-19): a systematic review of imaging findings in 919 patients. *Am. J. Roentgenol*, 2020.

[7] G. Soldati et al. Is there a role for lung ultrasound during the covid-19 pandemic. *J. Ultrasound Medicine*, 2020.

[8] L. Demi. Lung ultrasound: The future ahead and the lessons learned from covid-19. *The J. Acoust. Soc. Am.*, 2020.

[9] G. Soldati M. Demi A. Smargiassi R. Inchingolo L. Demi. The role of ultrasound lung artifacts in the diagnosis of respiratory diseases. *Expert Review of Respiratory Medicine*, 2019.

[10] G. Soldati A. Smargiassi R. Inchingolo D. Buonsenso T. Perrone D. F. Briganti S. Perlini E. Torri A. Mariani E. E. Mossolani F. Tursi F. Mento L. Demi. Proposal for international standardization of the use of lung ultrasound for patients with covid-19. *J Ultrasound Med*, 2020.

[11] R. J. Van Sloun L.Demi. Localizing b-lines in lung ultrasonography by weakly supervised deep learning, in-vivo results. *IEEE journal biomedical health informatics*, 2019.

[12] 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*, 2020.

[13] World Health Organization. To x-ray or not to x-ray? `https://www.who.int/news-room/feature-stories/detail/to-x-ray-or-not-to-x-ray-`.

[14] H. Shin et all. Learning to read chest x-rays:recurrent neural feedback model for automated image annotation. *CVPR*, 2016.

[15] T. A. Tabbara et all. Acute epiploic appendigitis: Diagnostic and laparoscopic approach. *International Journal of Surgery Case Reports*, 2018.

[16] A. Dimoka. How to conduct a functional magnetic resonance (fmri) study in social science research. *MIS Quarterly*, 2012.

[17] J. S. Abramowicz. Ultrasound imaging of the early fetus: is it safe? *Imaging in Medicine Journal*, 2009.

[18] Obstetric ultrasound History Web. Karl dussik. https://www.ob-ultrasound.net/dussikbio.html.

[19] Obstetric ultrasound History Web. George d. ludwig. https://www.ob-ultrasound.net/ludwig.html.

[20] T G Brown I. Donald J Macvicar. Investigation of abdominal masses by pulsed ultrasound. *Lancet*, 1958.

[21] Obstetric ultrasound History Web. I. donald. https://www.ob-ultrasound.net/iandonaldbio.html.

[22] Imaging Technology News. Emerging trends in ultrasound imaging. https://www.itnonline.com/article/emerging-trends-ultrasound-imaging.

[23] D. Junuzovic A. Carovac, F. Smajlovic. Application of ultrasound in medicine. *Acta Inform Med*, 2011.

[24] Imaging Technology News. Healcerion receives fda approval for sonon 300l handheld ultrasound device. https://www.itnonline.com/content/healcerion-receives-fda-approval-sonon-300l-handheld-ultrasound-device.

[25] N. M. Tole. *Basic physics of ultrasonographic imaging*. Number page 7-92. World Health Organization, first edition, 2005.

[26] HEALING PICKS. How long do piezoelectric crystals last? https://healingpicks.com/how-long-do-piezoelectric-crystals-last/.

[27] M. Mischi J. M. Thijddrn. *ultrasound imaging arrays*. Number page 323-341. Oxford: Elsevier, first edition, 2014.

[28] L. Demi. Practical guide to ultrasound beam forming: Beam pattern and image reconstruction analysis. *Applied Sciences*, 2018.

[29] WebMD. Picture of the lungs. https://www.webmd.com/lung/picture-of-the-lungs.

[30] L. Demi T. Egan M. Muller. Lung ultrasound imaging, a technical review. *Applied Sciences*, 2020.

[31] G. Soldati A. Smargiassi L. Demi R. Inchingolo. Artifactual lung ultrasonography: It is a matter of traps, order, and disorder. *Applied Sciences*, 2020.

[32] G. Soldati M. Demi A. Smargiassi L. Demi R. Inchingolo. On the physical basis of pulmonary sonographic interstitial syndrome. *J Ultrasound Med*, 2016.

[33] Johns Hopkins Medicine. Covid-19 lung damage. https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus/what-coronavirus-does-to-the-lungs.

[34] I. Goodfellow Y. Bengio A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[35] Openstax. Neurons and glial cells. https://openstax.org/books/biology/pages/35-1-neurons-and-glial-cells.

[36] A. Woodruff. What is a neuron? queensland brain institute. https://qbi.uq.edu.au/brain/brain-anatomy/what-neuron, 2019.

[37] A. Nagpal. L1 and l2 regularization methods. https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c.

[38] P. Sanagapati. What is dropout regularization? https://www.kaggle.com/code/pavansanagapati/what-is-dropout-regularization-find-out/notebook.

[39] MRI Questions. Convolution. https://mriquestions.com/what-is-convolution.html.

[40] M. Rizwan. Lenet-5 a classic cnn architecture. https://www.datasciencecentral.com/lenet-5-a-classic-cnn-architecture/.

[41] A. Krizhevsky I Sutskever G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 2017.

[42] S. A. G. Shakhadri. Build vgg net from scratch with python! https://www.analyticsvidhya.com/blog/2021/06/build-vgg-net-from-scratch-with-python/.

[43] V. Feng. An overview of resnet and its variants. https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035.

[44] N. Savage. Breaking into the black box of artificial intelligence. *Nature*, 2022.

[45] R. Roshankhah Y Karbalaeisadegh H. Greer F. Mento G. Soldati A. Smargiassi R. Inchingolo E. Torri T. Perrone S. Aylward L. Demi and M. Muller. Investigating training-test data splitting strategies for automated segmentation and scoring of covid-19 lung ultrasound images. *the Journal of The Acoustical Society of America*, 2021.

[46] L. Longo G. Vilone. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, 2021.

[47] S. Ulyanin. Implementing grad-cam in pytorch. https://medium.com/@stepanulyanin/implementing-grad-cam-in-pytorch-ea0937c31e82.

[48] 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. *National Library of Medicine*, 2021.