Universitat de Girona
Escola Politècnica Superior

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Final Degree Project

Grade: Biomedical Engineering

Title: Semantic Segmentation of LUS Retraining a Convolutional Neural Network

Document: Memory

Student: Ferran Franco i Moral

Tutors: Xavier Cufí, Joan Aranda, Maria Farahi
Department: Arquitectura i Tecnologia de Computadors
Area: Arquitectura i Tecnologia de Computadors

Call (month/year): June 2023

Acknowledgments

First of all, I would like to thank the work of my tutors to assist me during this project providing me with knowledge, experience and tranquilizing me when it was needed. I specially would like to thank Maria Farahi work as a mentor during this entire project and Xavier Cufí for aiding me when was needed.

Finally I will be thanking to my family, my couple and my roomates for helping me and motivating me through these months where I might have been sometimes frustrated and a little bit irritated.

To all of them and to anyone that might read this final degree project, thank you.

Ferran Franco i Moral

Abstract

On the last years respiratory diseases have been on the news almost every day. After the Covid-19 global pandemic, many studies where done in order to detect and avoid spreading the disease. As in many other different diseases, this studies tend to work with techniques that are available mostly on first world countries (using MRI, TC and X-Ray). **Lung Ultrasound (LUS)** has been increasingly used to diagnose and monitor different lung diseases. It's a non-invasive and cheaper technique than the others talked previously. The main issue is that acquiring a LUS scan is easier than analysing and understanding the images and characteristics of each respiratory disease.

To try solving this issue, over the last decade, many studies had the main purpose to demonstrate that **deep learning** techniques can be a possible solution. Despite having very promising results in automatic segmentation, there's still a necessity of **available data** to train and test this structures.

Through this work, a convolutional neural network based on the architecture **U-Net** has been re-trained, validated and tested in order to segment different patterns from lung ultrasound images obtained from papers published around the last years.

At the end of this project, a U-Net network has been obtained with 98.1455 % accuracy detecting a mask with all 3 types of lines that can be found in LUS, **A-lines, B-lines and Pleural line**. And a U-Net Network with 98.6894 % accuracy detecting a **B-line mask**. Despite being high accuracy values, as it will be seen on this project, the results aren't good.

From this bad results, at the end of this final degree project, it couldn't be concluded that the use of deep learning techniques in **Semantic Segmentation of LUS** isn't a good combination. In order to discard this theory, more tests and further investigation should be done.

Keywords: Lung Ultrasound, U-Net, Deep Learning, Convolutional Neural Network, Semantic Segmentation.

Index

Image Index

1. Introduction

From November 2022 until June 2023 I had the opportunity to work at the Research Center in Biomedical Engineering (CREB in Spanish) at the Universitat Politècnica de Catalunya, UPC. My final degree project is one of the parts included inside a thesis of a student of the UPC, Maria Farahi. The results obtained on this final degree project will be a piece of a larger thesis that has the objective to find the best method to create a mask of the different patterns that can appear on a Lung Ultrasound, known as A-lines, B-lines and Pleural lines, that can have a direct correlation with some concrete lung diseases such as Pneumothorax, Covid-19 and Pneumonia.

I was tasked to retrain a U-Net network, with Lung Ultrasound Images (LUS) obtained from a public dataset in order to distinguish the desired patterns that have just been explained.

The main objective in this final degree project is being able to segment these structures from a LUS image with the highest accuracy possible and, based on these results, determine if the use of Deep Learning on LUS images could be a good technique to implement.

In order to understand everything explained on the practical part of this final degree project, some theoretical fundaments are required. These are divided in two different areas, the medical area and the engineering area.

Firstly talking about the medical area, this one consists in a few concepts related with anatomy and diseases of the respiratory system and how different respiratory diseases can be distinguished using ultrasound imaging.

On the other hand, while talking about the engineering part, some concepts about Semantic Segmentation, Artificial Intelligence, Artificial Neural Networks, Convolutional Neural Networks and U-Net networks appear.

The following is a proposal of a method (using MATLAB), to detect with the highest achieved accuracy, a mask containing A-lines, B-lines and Pleural lines from LUS images in order to, in future stages, facilitate healthcare system detection of lung diseases.

2. Previous Concepts

3.1 Respiratory System Anatomy

Lungs make up a large part of the respiratory system (shown in Image 1), which is the network of organs that are responsible of breathing.

The right lung is divided in three lobes: the superior, the middle and the inferior. It's shorter and wider than the left lung. The left lung is divided in two lobes: the superior and the inferior.

Both lungs are covered with a protective covering called pleural tissue.

These organs are responsible to make oxygen available to the body and remove other gases such as carbon dioxide. When an inhalation is done the air comes through the nose or mouth, it travels through the pharynx, larynx and into the trachea. This cartilaginous structure is divided into two tubes called bronchial tubes (one for each lung). These tubes lead to smaller air passages called bronchi and these ones into bronchioles. At the end of these structures there are tiny air sacs called alveoli. This last structure is responsible for transferring oxygen into the blood from the air inhaled.

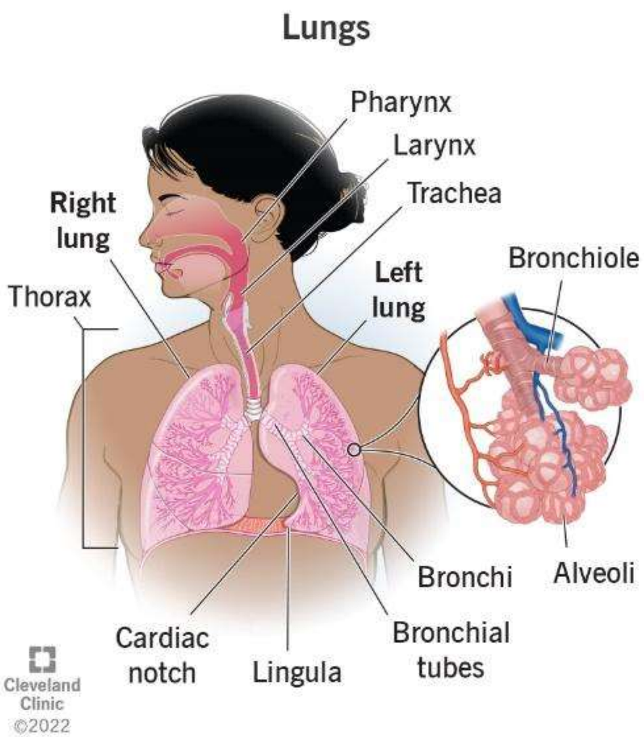


Image 1: Respiratory System Anatomy. Source: [1]

3.2 Lung diseases

In actuality there are many different lung pathologies. Each one of them has a specific treatment and a specific way to detect. These diseases can be caused by different factors[1]:

1. Due to the conditions where the affected live, an example could be Middle Eastern Respiratory Syndrome (MERS). A virus that passes primarily to people from camels, most cases have been found in Saudi Arabia. This disease might develop pneumonia and the affected may also experience breath failure, kidney damage, high fevers and even death.
2. The habits that the affected have, an example could be Silicosis. A disease mainly found on workers exposed to silica dust in jobs such as construction or mining. After some time, silica particles cause scarring in the lungs, reducing the ability to breathe.
3. Presence of germs, an example could be Respiratory Syncytial Virus (RSV). A disease caused by a common respiratory virus. It can affect people from all ages and it tends to have similar symptoms as the common cold, but it could be severe and life-threatening.
4. Others.

As it can be seen, many factors need to be taken into account in order to determine what disease is affecting a lung. Due to the variability of pathologies and symptoms sometimes can be tricky to detect and help to diagnose what is happening to the lungs patient. The pathologies explained on the following Table (*Table 1*) are the ones that appear on the dataset used on the practical part of this final degree project.

[1] In case more information about different diseases is required, consult: [2] (American Lung Association) & [3] (Breathe the lung association)

Condition / Dysfunction	Parts Affected	Brief Explanation	Symptoms	Prevalence
Covid-19	This disease can affect the lower or upper part of the respiratory system.	Covid-19 and its symptoms can vary from mild to severe. This disease is produced by the virus SARS-CoV-2. This virus can irritate and inflame the respiratory tract.	- Fever or chills - Cough - Shortness of breath or difficulty breathing - Fatigue - Muscle of body aches - Headache - Loss of taste or smell - Sore throat - Congestion or runny nose - Nausea or Vomiting - Diarrhoea	On 12 April 2023 there were 762.791.152 confirmed cases including 6.897.025 deaths according to the World Health Organization (WHO).[2]
Pneumonia	This infection causes alveoli to become inflamed and fill up with fluid or pus in one or both lungs.	Lung infection caused by bacteria / viruses / fungi. Pneumonia and its symptoms can vary from mild to severe. It is difficult for the oxygen to pass from the lungs into the bloodstream. There are many factors that can increase the complexity of this disease such as the type of germ, the age of the patient, and their overall health.	- Cough - Fever - Shortness of breath - Rapid, shallow breathing - Sharp or stabbing chest pain - Loss of appetite, low energy, fatigue - Nausea or Vomiting - Confusion	Pneumonia is a leading cause of hospitalisation and deaths in seniors and with people with chronic diseases. Ten of thousands of people in the U.S die from pneumonia every year, most of them adults over the age of 65. There are many things you can do to lower your risk of getting pneumonia and it has a treatment[3].
Pneumothorax	This disease affects the pleural cavity. It can be a complete or a portion lung collapse.	Pneumothorax (also known as collapsed lung) is defined as the presence of air or gas in the pleural cavity. This disease prevents the lung from expanding when inhaling. This can be caused spontaneously, by underlying lung diseases (such as asthma, pneumonia and others) by blunt or penetrating chest injury or activities with a dramatic change of air pressure such as diving or flying in an aeroplane.	- Sharp, stabbing chest pain - Shortness of breath - Bluish skin caused by a lack of oxygen - Fatigue - Rapid breathing and heartbeat - Dry cough	This condition occurs (worldwide) 7.4 to 18 per 100.000 men each year and 1.2 to 6 per 100.000 women each year.

Table 1: information about the 3 main diseases talked about in this final degree project.

[2] [4]
[3] [5]

3.3 Comparison between different Image Techniques

There are many different Image Techniques to detect and assist in the fight against different diseases and pathologies that might occur on the human body. In this case, it will only be focused for the uses that these techniques can be applied in the lung field.

We must talk about the 3 Image Techniques that we have nowadays in a big majority of the first world countries, those are the Magnetic Resonance Imaging (MRI), the Computed Tomography (CT) and the Ultrasound (US).

For this final degree project ultrasound images have been used. US is widely used as a diagnostic tool for analysing live images of the main structures of the body, especially the tissue joints within the body.

It uses high frequency waves to create a live video from inside the body. Unlike X-ray/ CT/ MRI, you can analyse in real time how the body works, and this is done without radiation. With this technique you can study how the blood flows through the vessels or how a muscle moves doing a certain movement.

Talking about the image quality, US images have a very poor resolution whereas MRI and CT scan have a better resolution. The one with the higher quality image is the MRI but you need much more time to have the results than with the other techniques.

Another strength of the US is that its devices (compared with the other techniques) are much cheaper and they don't really need a huge structure behind. When we talk about MRI and CT scans there must be a huge investment behind. Both of these devices need to be protected within a special sealed room (the first one to avoid magnet issues and the second one requires a plumbed room in order to avoid radiation to leave the room).

As a summary, on the following table (Table 2), it can be seen each strengths and weaknesses of the already talked techniques.

MRI	**CT**	**US**
Slow	Fast	Fast
Expensive	Expensive	Cheap
Needs a special room to work safely	Needs a special room to work safely	Doesn't need an environment for working
Higher Definition	High Definition	Poor Definition
No Irradiation	Irradiation	No Irradiation
Works with electromagnetic fields	Works with radiation	Works with ultrasounds

Table 2: Strengths and weaknesses of the talked techniques

3.4 Lung Ultrasound patterns in different diseases

As it has been said, lungs are the main organs that contain air of the human body. When a lung is affected by a disease, the air-liquid ratio in the lung changes. By this changing, some abnormal artifacts can be seen during a US examination (see Image 2). Some different pathologies have some common LUS artifacts, these are the ones that are mostly studied on this project, A- and B-lines.

A-lines, seen on the LUS as a horizontal linear structure parallel to the pleura. These artifacts are caused by a multiple reflection of sound waves due to the difference of acoustic impedance between the pleura and the lung. Their presence can’t be taken as an indicative of a healthy nor affected lung because they can appear on both cases. On the other hand, B-lines can be seen as a vertical artifact and they appear when there is less air on the lung and the quantity liquid is superior than it should be [13].

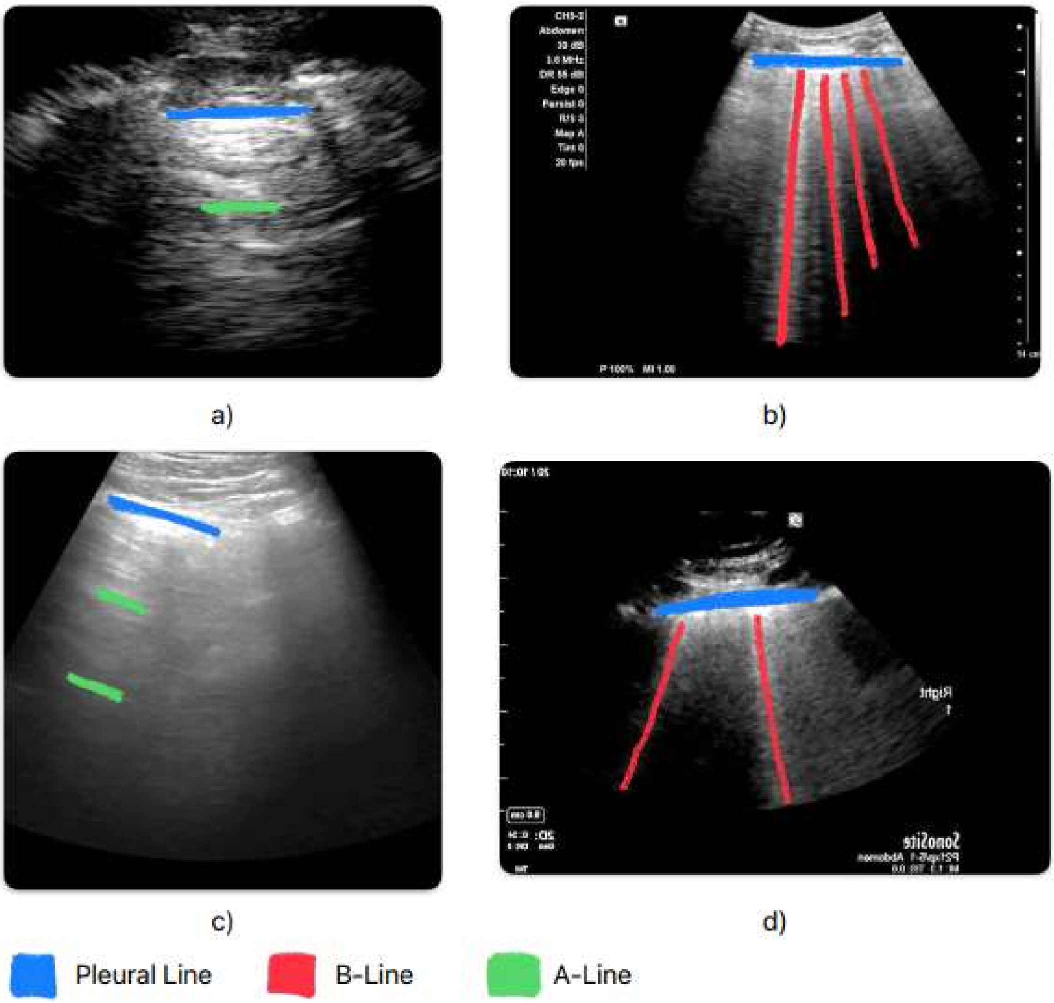


Image 2: Lung US of different patients: a) Pneumothorax; b) Pneumonia; c) Healthy lung; d)Covid-19. With Plaural line, B-line and A-line shown on LUS images.

There are some protocols that have the objective to assist technicians to diagnose lung pathologies through US, one example could be the BLUE-protocol [14]. This technique can be done in less than 3-minutes and allows determining which of the following diseases might a patient have.

- Pulmonary edema
- Pulmonary embolism
- Pneumonia
- Chronic Obstructive Pulmonary Disease
- Asthma
- Pneumothorax

Diagnose will depend on which type of the following profiles is found (shown on Table 3).

- The A-profile associates anterior lung-sliding with A-lines.
- The A'-profile is an A-profile with abolished lung-sliding.
- The B-profile associates anterior lung-sliding with lung-rockets.
- The B'-profile is a B-profile with abolished lung-sliding.
- The C-profile indicates anterior lung consolidation, regardless of size and number. A thickened, irregular pleural line is an equivalent.
- The A/B profile is a half A-profile at one lung, a half B-profile at another.

Mechanism of dyspnea	Profiles of BLUE-protocol	Sensitivity	Specificity	Positive predictive value	Negative predictive value
Acute hemodynamic pulmonary edema	B-profile	97%	95%	87%	99%
		(62/64)	(187/196)	(62/71)	(187/189)
COPD in exacerbation or severe acute asthma	Nude profile	89%	97%	93%	95%
		(74/83)	(172/177)	(74/79)	(172/181)
Pulmonary embolism	A-profile (with deep venous thrombosis)	81%	99%	94%	98%
		(17/21)	(238/239)	(17/18)	(238/242)
Pneumothorax	A'-profile (with lung point)	88%	100%	100%	99%
		(8/9)	(251/251)	(8/8)	(251/252)
Pneumonia	B'-profile	11%	100%	100%	70%
		(9/83)	(177/177)	(9/9)	(177/251)
	A/B profile	14.5%	100%	100%	71.5%
		(12/83)	(177/177)	(12/12)	(177/248)
	C-profile	21.5%	99%	90%	73%
		(18/83)	(175/177)	(18/20)	(175/240)
	A-V-PLAPS profile	42%	96%	83%	78%
		(35/83)	(170/177)	(35/42)	(170/218)
	The four profiles	89%	94%	88%	95%
		(74/83)	(167/177)	(74/84)	(167/176)

Table 3: Detailed performances of the BLUE-protocol. Numbers of patients are shown in parentheses. [14]

3.5 What is semantic segmentation?

Semantic segmentation is a deep learning algorithm that assigns a label or category to each pixel of an image. It is used to recognise a group of pixels that form each different category.

A modern example could be how autonomous cars need to distinguish other cars, the street, the sky… Some other fields where it is used could be the following:

- Industrial inspection
- Medical image processing
- Satellite image generation
- Robotic Vision

An example of semantic segmentation is the separation of an image between two classes. As it can be seen on Image 3, a person appears and you can see it is segmented between the pixels that are related to the person and the ones that are related to the background of the image. Semantic segmentation can work with more than two classes, this is just an example.

Image 3: Image and it's semantic segmentation separating background and the person. [15]

3.6 Artificial Intelligence (AI) and Artificial Neural Network

3.6.1 Basic concepts

AI is described as software capable to behave, with some limitations, as a human being. The word artificial comes from the Latin root word "*facere arte"* which means "make something".

Some of the common descriptions/definitions of what is AI are the following:

- AI is intelligent because it's able to learn.
- AI is a good tool to work with problem solving.
- AI is a good representation of the ability to adapt to the environment and to work with complete or incomplete knowledge.

An Artificial Neural Network (ANN) is a method used in artificial intelligence that consists on teaching computers to process data in a way inspired by the human brain. This type of machine learning process (known as **Deep Learning**) uses interconnected nodes or neurons in a structure formed from layers. This technique is also known as:

- Parallel distributed processing model
- Connectivist/connectionism model
- Adaptive system
- Self-organising system
- Neurocomputing
- Neuromorphic system

As it has been said, an ANN tries to mimic the human brain's biological neural network. The biological neural network is the structure through which a living organism is able to perform complex tasks instinctively. The central processing unit is called **neuron**. The human brain has around 10 to 100 billion neurons. The connection that occurs between each one of them to many others is called **synapses**. Human brain has around 100 trillion synapses.

Talking about an Artificial Neural Network, this structure contains nodes called **neurons**. These neurons are connected between them by **edges**. Each node sends information through an edge and it will be computed in the next node. As in the biological brain, each neuron can be connected to multiple neurons. The common structure of an ANN is to divide the neurons by **layers**.

Input Layer	Responsible for accepting data and passing it to the rest of the network.
Hidden Layers	Responsible for the optimum performance and complexity of the neural network. As many hidden layers there are, the more complex the network will be.
Output Layer	Responsible for holding the result or output to the problem the ANN is trained to solve.

Table 4: Artificial Neural Network tend to be divided into layers. Here are what types there are and their purposes

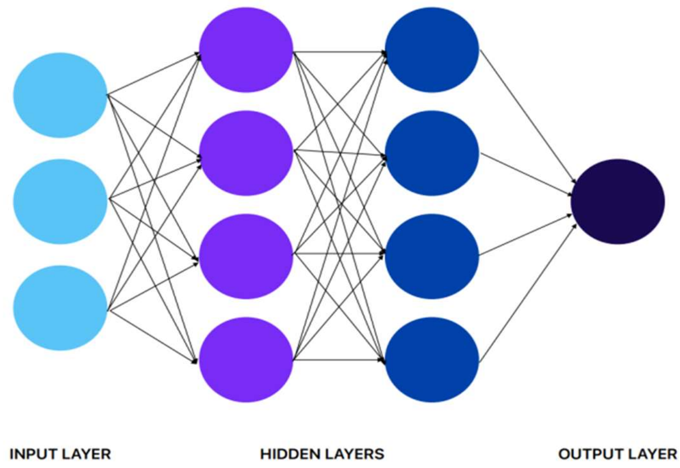


Image 4: Common structure of an Artificial Neural Network, it contains the input and the output layers with two hidden layers between them.

An Artificial Neural Network is created to learn how to do a task. This is done through some **weights** related to the edges. These weights will be constantly updating in order to train the network during the training process; they self-adjust depending on the difference between predicted outputs and training outputs. Characteristics with close to zero weights values will be considered as low-importance in the prediction process compared to the characteristics with a higher weight.

Some other components are needed in order to have a good neural network. Activation Function and Bias are two crucial components in artificial neural networks.

Activation Function is required to add non-linearity to the neural network. This mathematical formula helps the neuron to switch between ON and OFF.

Bias can be defined as the constant which is added to the product of features and weights. It's used to offset the result, this will help the models to shift the activation function towards the positive or negative side.

The explained variables are used to determine the structure or architecture of the Neural Network, but are the following (**optimizer** and **loss function**) the ones that allow these systems to learn how to do a task.

An **optimizer** is a function or an algorithm that modifies the attributes of the neural network, such as weighs and learning rate. By doing so, it helps reducing the overall loss and improving the accuracy. It can be challenging to choose the rights weights because a deep learning model generally consists of millions of parameters. That's why a suitable optimizer algorithm for each application can be very useful.

The learning process is done through repeating different paths and learning from the mistakes to improve continuously. This is calculated through the **Loss Function**. This function compares the target and predicted output values. In a few words, it measures how accurate is the neural network modelling the training data. The objective is to have a low loss function value.

These kinds of tools can be very useful in order to solve difficult problems, such as the one exposed in this final degree project, image recognition.

3.6.2 Transfer Learning

Transfer Learning is a Deep Learning methodology in which a model that has been trained to do one task is used as a starting point to create a new model that performs a similar task. This is useful because it tends to be much easier and faster to update and retrain a network with this technique than train it as new.

Transfer Learning is a commonly used technique because:

- It allows to train models with less labelled data by reusing commonly used models that have been previously trained with large data sets.
- Helps reduce training time and computing resources. Since the previously trained model has already learned the weights from the previous learning, these new weights are not learned from scratch.
- It allows to take advantage of model architectures developed by the deep learning community, some examples could be GoggLeNet and ResNet networks widely used.

As it can be seen on Image 5, Transfer Learning consists in 5 main steps:

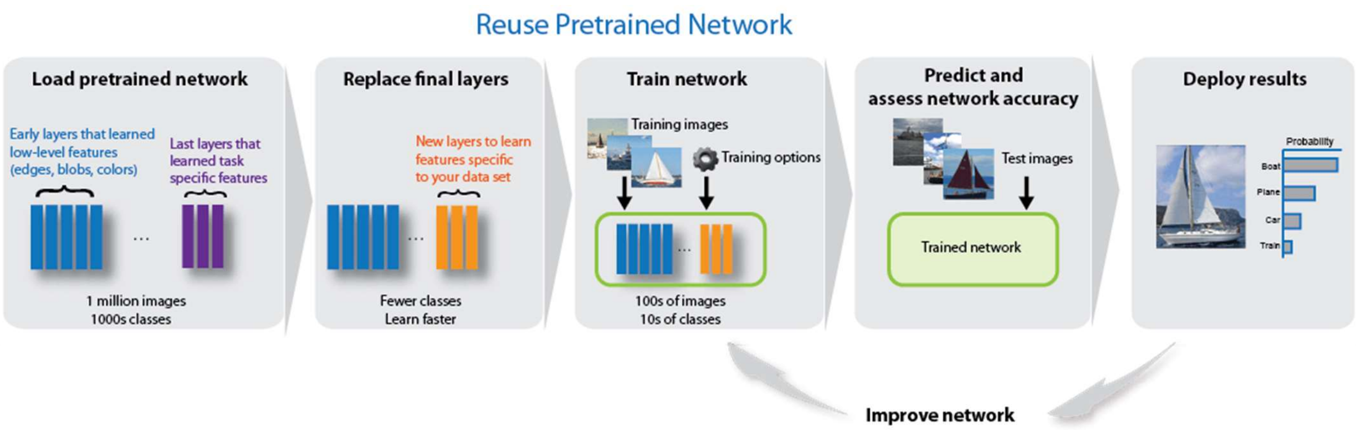


Image 5: Transfer Learning Process. [16]

First of all the pretrained network that has been chosen has to be loaded. On this example, the loaded network has 1 million images and 1000 classes, some early layers (that have learned some low-level features such as colors, edges…) and some last layers that have learned task specific features.

On the second step, these last layers must be changed for a new algorithm that will be able to learn the new characteristics of the new dataset. Once we have this modificated network, training can be done. In order to do this step, some options must be defined, these are the following:

Field	Description
MaxEpochs	Maximum number of epochs you want to use for training, specified as a positive integer. An epoch is the full pass of the training algorithm over the entire training set.
MiniBatchSize	Minibatch size you want to use for each training iteration, specified as a positive integer. A minibatch is a subset of training set that is used to evaluate the gradient of the loss function and update the weights.
InitialLearnRate	Initial learn rate used for training, specified as a positive scalar. If the learning rate is too low, training can take a long time and if the learning rate is too high, the training could achieve a suboptimal result or diverge.
Shuffle	Option to change the order of data.
ValidationData	Data that you want to use for validation during training, specified as a [], a data store, a table or an array of cells that contains the validation predictors and responses.
ValidationFrequency	Frequency of network validation in number of iterations, specified as a positive integer.
Verbose	Flag to display training process information in the command window, specified as 1(true) or 0 (false). Verbose output displays the following information shown on Image 6.

Table 5: Options used during the training[18]

Field	Description
`Epoch`	Number of epochs. One epoch corresponds to one complete pass of the data.
`Iteration`	Number of iterations. One iteration corresponds to one mini-batch.
`Time Elapsed`	Elapsed time in hours, minutes, and seconds.
`Mini-batch Accuracy`	Classification precision in the mini-lot.
`Validation Accuracy`	Classification accuracy on validation data. If you do not specify validation data, the function does not display this field.
`Mini-batch Loss`	Loss in the minilot. If the output layer is an object `ClassificationOutputLayer`, the loss is the cross-entropy loss for multi-class classification problems with mutually exclusive classes.
`Validation Loss`	Loss in validation data. If the output layer is an object `ClassificationOutputLayer`, the loss is the cross-entropy loss for multi-class classification problems with mutually exclusive classes. If you do not specify validation data, the function does not display this field.
`Base Learning Rate`	Base learning rate. The software multiplies the learning rate factors of the layers by this value.

Image 6: Verbose output displays[18]

The last two steps consist in doing the prediction and obtain the confusion matrix in order to analyse how good the training has been performed. If the results are not as good as it was expected, some training options can be modified in order to try to obtain better accuracy with test images.

3.6.1 Neural Networks types

There are many different types of neural networks. This variability appears because they can be classified depending on their: Structure, Data Flow, Neurons used in their density, Layers and their depth activation filters…

- Perceptron
- Feed Forward Neural Network
- Multilayer Perceptron
- Convolutional Neural Network (CNN)
- Radial Basis Functional Neural Network
- Recurrent Neural Network (RNN)
- LSTM – Long Short-Term Memory
- Sequence to Sequence Models
- Modular Neural Network

CNN and RNN are the two most popular architectures included inside the supervised deep models. These kind of deep models are widely used for classification, segmentation, and detection of anatomical structures in medical US images. They are called supervised models because they require the utilization of time-consuming, labour-intensive, and expensive human annotations.

3.6.3 Convolutional Neural Network

Convolutional Neural Networks, known as CNN, are a type of artificial neural network widely used in image recognition due its ability to recognize patterns in images. It's a powerful tool, but the size of the available training sets and the size of the considered networks are two clue factors that limit their success.

CNN can have tens or hundreds of layers, and each one of them learns to detect different characteristics of an image. On each layer some filters are applied to each training image and the result of convoluting each image is then used as an input for the following layer. These filters can be very simple such as changing the brightness or change the boundaries, and they can be more complex with each layer reaching some characteristics that define the object [19].

CNN architecture consists in an input layer, some hidden layers and an output layer (see Image 4). The more typical layers are the following.

- Convolution: It applies a group of convolutional filters to the input images, each filter activates different characteristics of the image. This layer will decrease the image size as well as bringing all the information in the field together into a single pixel. Lower or earlier layers identify low-level features (such as edges or colors), while higher layers identify high-level features (such as concepts or complex patterns).

- Rectified Linear Unity (ReLU): Keeps positive values and sets negative values to 0, allowing a faster and efficient training. On the chapter 3.7 was referred as Activation Function, because only the "activated" characteristics can go to the following layer.

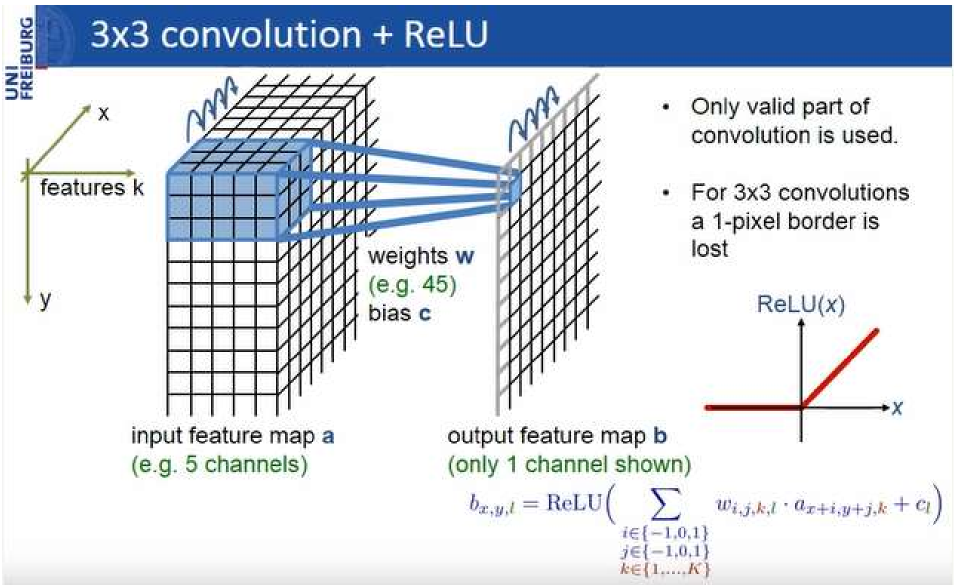


Image 7: Example of a convolution layer in a CNN followed by a ReLU activation function. [20]

- Pooling: Simplifies the output through a nonlinear downsampling, reducing the number of parameters that the network needs to learn and decreasing the computational power required to process the data. It is useful to extract dominant features which are invariant to position and rotation, maintaining effective the process of training the model. There are two types of Pooling (see Image 8):
    - Max Pooling: returns the maximum value of all values of the portion of the image covered by the Kernel.
    - Average Pooling: returns the average of all the values of the portion of the image covered by the Kernel.

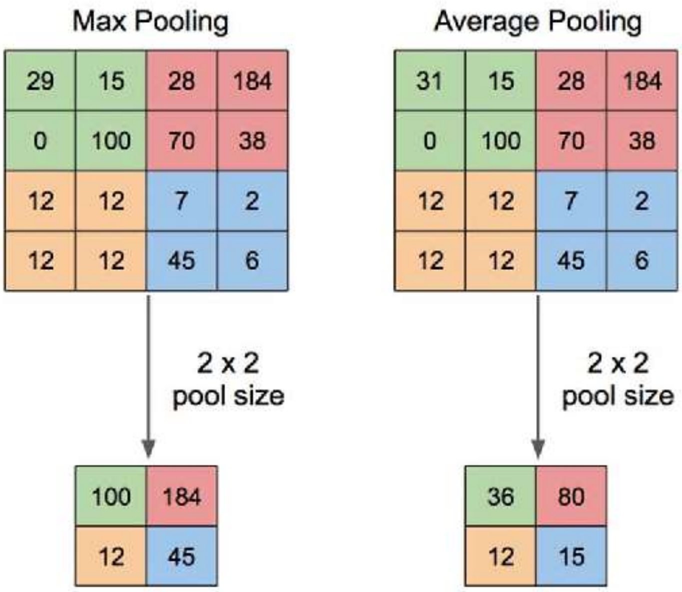


Image 8: Each type of pooling. [21]

After learning features in many layers, the architecture of a CNN shifts to classification. The previous layer at the last layer has a vector of K dimensions, where K is the number of classes to be predicted, and will contain the probabilities of each class of an image being classified.

The last layer of the CNN architecture uses a classification layer to provide a final classification output.

3.6.4 U-Net

U-Net architecture is a neural network located in the group of "fully convolutional neural network". It was proposed on 2015 by Ronneberger et. Al. [22], which is popularly used for echocardiogram segmentation.

The main characteristic is the symmetrical U-shaped architecture that this network has, shown on Image 9. This allows optimization an easier task as the two symmetrical parts are complementary to each other and modifications will be applied in both sides at the same time. The network architecture consists of a contracting/encoding path (left side) and an expansive/decoding path (right side).

The **contracting path** follows a typical architecture of a convolutional network. It consists in a repeated application of two 3x3 convolutions, each followed by 2x2 max pooling operation. At each downsampling step, the feature channels are doubled (number above the blue boxes).

The **expansive path** consists in an upsampling of the feature map followed by a 2x2 convolution that halves the number of feature channels, a concatenation with the cropped feature map from the contracting path (necessary due to the loss of border pixels in each convolution), and two 3x3 convolutions, each followed by a ReLU. On the last layer, a 1x1 convolution is used to map each 64- component feature vector to the desired number of classes.

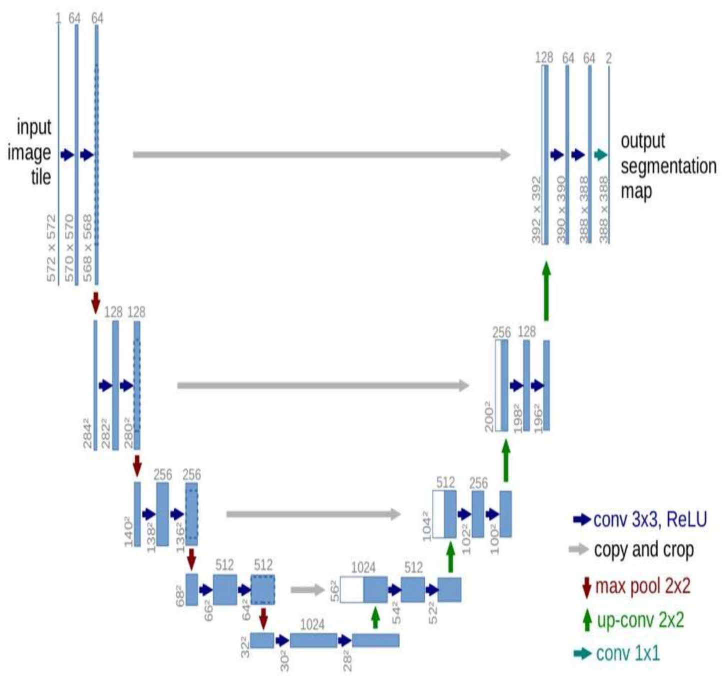


Image 9: U-net architecture. The operations denoted by each are arrow are mentioned in the image. [22]

3.7 Grad-CAM

Now that concepts such as Deep Learning, Convolutional Neural Network and U-Net architecture have been introduced, we can continue with how to analyse them. When we work with Deep Learning we don't truly know what is going on with the algorithm, we obtain a result as accuracy or matrix confusion without any more information (see Image 10). This problem is known as explicability.

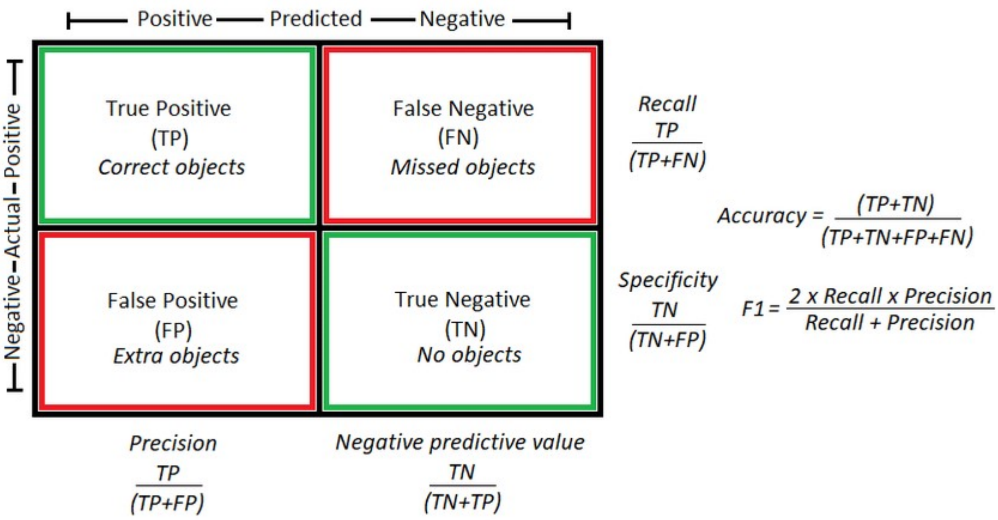


Image 10: Accuracy and matrix confusion formulas. [23]

To analyse the results of the Network, Grad-CAM is a very useful tool to use. This method, that is applicable to a wide variety of CNN and was published on 2019 [24]. Uses the gradient of any target concept ('dog' for example) flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept [23].

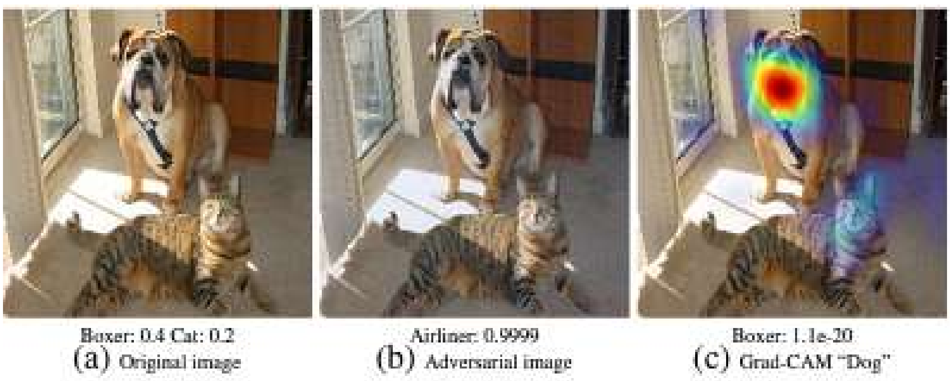


Image 11: a) Original image, b) Adversial image created for the category "airliner". c) Grad-CAM visualization for the category "Dog" along with the confidence. [24]

This method is a common technique used in order to make the neural network more transparent and help finding issues in case an error is found. GradCAM generates a “heat map” that localises the target concept that has to be found and can be a qualitative method to determine how robust a neural network is.

4 State of the art

Ultrasound technique after Covid-19 pandemic has increasingly used in worldwide countries. Because of its challenging analysis Deep Learning has been a very useful tool to combine with medical US analysis. On the following Image 12, there is an image with the applications of this technique in medical US analysis until 2018. After the pandemic on 2019-2020 many more papers have appeared where the anatomical structures studied are the lungs.

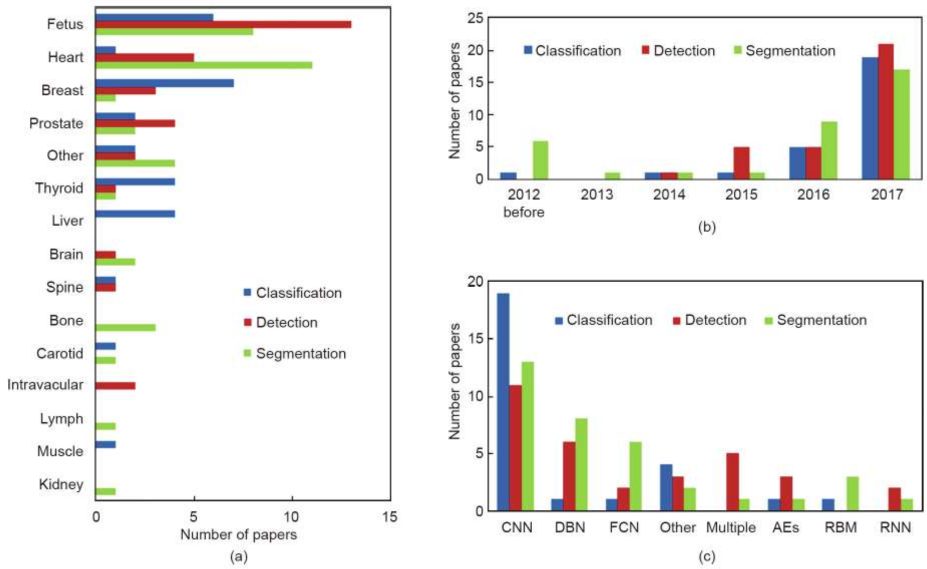


Image 12: Applications of deep learning in medical US analysis in 2018. (a) Anatomical structures; (b) year of publication; (c) network architectures. DBN: deep belief network; FCN: fully convolutional network; Multiple: a hybrid of multiple network architectures; RNN: recurrent neural network; AEs include its variants, the sparse auto-encoder (SAE) and stacked denoising auto-encoder. [25]

Current applications of Deep Learning techniques applied in medical US analysis tend to involve in these three types of tasks: classification, detection and segmentation for different anatomical structures or tissues. Despite this, nowadays there isn’t much investigation working on LUS and segmentation.

When Covid-19 was declared a global pandemic on 2020, many studies and some research appeared in order to obtain quality datasets to face this disease that had the entire world paralyzed[26][27][28]. After this disease appeared, and the scientific community discovered that the use of Deep Learning applied to LUS could have good results in order to differentiate other diseases, on 2021 some papers investigated on how this combination could work [29][30][31].

On many of these papers the main concern is that despite the increasing types of medical image datasets, there is still a deficiency of datasets focused on specific problems [32].

4.1 Data Augmentation

In order to solve this problematic, data augmentation is one of the best techniques to keep investigating without huge real datasets. It consists in, through different techniques such as rotating, modifying the brightness, flipping … the original images, obtain a bigger dataset with some variables and new aspects to take account as “new” cases to train the algorithm.

5 Hypothesis and objectives

5.1 Research Question

Can it be obtained a good accuracy by combining Deep Learning (U-net network) with Lung Ultrasound Segmentation in order to differentiate the patterns that appear on the images?

5.2 Hypothesis

We are able to obtain good accuracy results retraining a pre-trained U-net network to differentiate the patterns that appear on lung ultrasound images.

5.3 Objectives

The main objectives of this final grade project are the following: being able to differentiate background and the mask for each frame extracted from an ultrasound video, retrain a U-Net network and determine if the combination of Deep Learning and Lung Ultrasound Segmentation is a good procedure with high accuracy.

6 Materials and methods

6.1 Data

All the data is obtained from public or already used datasets from other different papers. For this final degree project different LUS data has been used. The data can be divided into the following groups:

Pathology	Description	Obtained from:
Covid19	Ultrasound image/video from patients that have Covid19	[33]
Pneumonia	Ultrasound image/video from patients that have Pneumonia	[33]
Pneumothorax	Ultrasound image/video from patients that have Pneumothorax	[34]
Regular	Ultrasound image/video from healthy subjects	[33]
Uninformative	Ultrasound image/video taken by a convex transducer from any organ except lung or heart.	Own data obtained by the chief of the group from different health centers.
Simulated Data	Simulated data of LUS used to train the network.	[35]

Table 6: Information about the data that has been used on this final degree project.

As it is explained on Annex D, all the data comes from public datasets. That's why talking about legal normative and data protection, as a team, we didn't need the presence of an ethics committee.

6.2 Manual Segmentation

The data used in order to train comes from a public dataset of simulated data from a paper [35] where the main goal was to detect Covid-19 feature detection using deep neural networks trained on simulated lung ultrasound.

To prepare the test data the first thing needed was a mask relating the A-lines, B-line and Pleural line. To do so, after a previous formation about how these lines looked on a LUS with the team where I was working, we started to manually segment them. The obtained mask appears as a black background with white lines where the marks have been done (see Image 13).

This procedure was done through a MATLAB code in order to do it easier. The property of this code isn't mine, which is why it's not located on the Annex B.

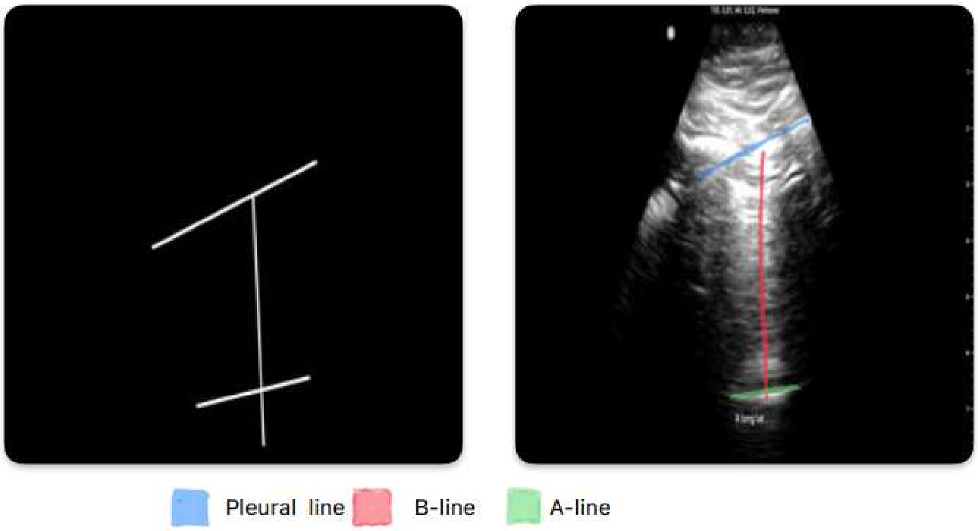


Image 13: Mask (left) of a original LUS imatge (right). This mask contains the 3 different lines.

6.3 Dividing the data (k-fold)

The original technique used to divide the data into training and validation was K-fold cross-validation. This dividing strategy consists in dividing the augmented data into K number folds as it is shown on Image 14.

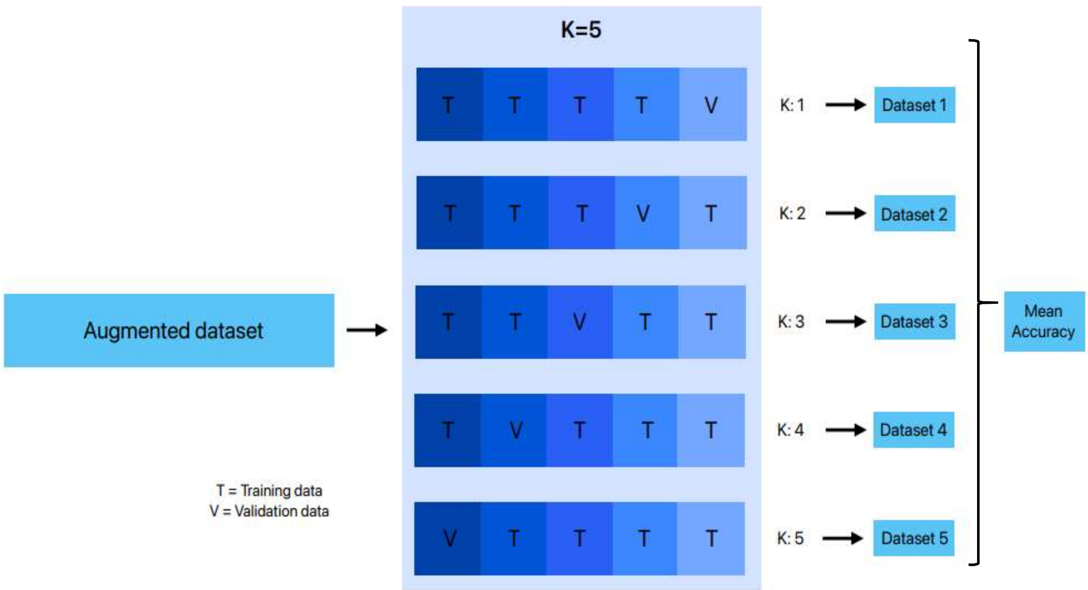


Image 14: 5 K-fold cross validation. From a single augmented data set obtain 5 different datasets with the respective training and validation data. The final objective is to obtain a mean of the 5 accuracies obtained on each of the 5 net validations.

This technique allowed to, from one single dataset, be able to train with k different datasets and to obtain an accuracy by calculating the mean accuracy of each training with each dataset. Despite being able to write the code that divided the data into 5 folders, one main issue was encountered, the network training had to be done through 5 different datasets increasing 5 times the time of training the network.

As an extraordinary measure, the team decided to work with a dataset divided into 70% of data into training and 30% into validation allowing to obtain some results at the end of the project.

6.4 U-Net

The implementation of U-net in this project was carried out in MATLAB R2022b and MATLAB R2023a for Windows 10 with Computer Vision, Deep Learning, and Parallel Computing toolboxes.

To create the U-Net layers the MATLAB function **unetLayers** [36] has been used. In our case, the image size was 256x256 and there were 2 classes with respective values 0 and 1: background and GT mask (containing A-lines, B-lines and Pleural-line). The neural network was trained using the function **trainNetwork** [37]. The options chosen for this training are explained on the following section 6.5.

The training has been used in 2 different systems. The environments are the following:

a) Intel(R) Core(TM) i9-7929X CPU @ 2.90GHz 2.90 GHz with 16 GB of RAM and NVIDIA titan Xp GPU with Computing Capability 6.1

b) Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz 3.19 GHz with 16 GB of RAM and NVIDIA GeForce GTX 1070

6.5 Hyperparameter Optimization

The training procedure was repeated 16 times obtaining 16 different nets with different accuracies.

All the nets have been trained with the first 10.000 images (7000 for training and 3000 for validation) from the simulated data obtained from the paper [35]. All these nets have been trained with the same data but what changed were the hyperparameter options used in order to train it.

Once the U-net architecture was implemented, the following list of hyperparameters were chosen to be modified to study how the neural network could be optimized.

5. Optimizer (explained in section 3.6.1): In training options, the optimizers used have been stochastic gradient descent with momentum (sgdm) and Adam optimizer.

    a. The **stochastic gradient descent** is an algorithm that updates the model parameters by taking a step in the direction opposite of the gradient of the loss function with respect to the parameters, this update is typically scaled by a learning rate. However, this algorithm can be slow to converge, especially in the presence of noisy or sparse gradients. By adding a momentum this issue is addressed. This momentum accumulates the past gradients to provide additional direction and speed during updates.

    o **Adam** (Adaptative Moment Estimation) maintains a set of exponentially decaying average of past gradients (first moment) and the squared gradients (second moment). It adapts the learning rate for each parameter by calculating individual adaptative learning rates, allowing it to perform well in the presence of sparse gradients and noisy data.

6. The next hyperparameter to be optimized is the **Initial learning rate** (explained in section 3.6.2). The default values for ‘sgdm’ solver is 0.01 and 0.001 for the ‘adam’ solver.
7. Another hyperparameter that has been modified is the **MaxEpochs** (explained in section 3.6.2). The values that have been tested are 4, 5, 10 and 40.

8. Finally the Mini-Batch Size hyperparameter (explained in section 3.6.2). For this project five different values have been tested: 4, 8, 16, 32 and 64.

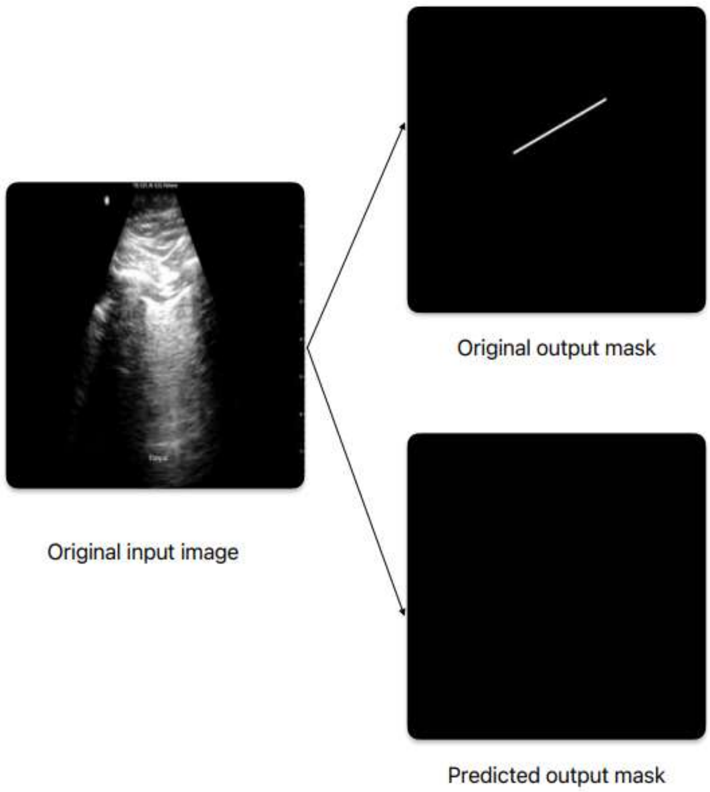
The other training parameters were fixed (for each of the training groups explained previously):

Parameter	**Explanation**
Shuffle	Shuffle the training data before each training epoch, and shuffle the validation data before each network validation.
ValidationData	explained in section 3.6.2
ValidationFrequency	explained in section 3.6.2
Verbose	explained in section 3.6.2
ExecutionEnvironment	Specification for the network to be trained with the Graphics processing unit (GPU).
Plots	This plot shows mini-batch loss and accuracy, validation loss and accuracy and additional information on the training process.
OutputNetwork	Return the network corresponding to the training iteration with the lowest validation loss.

Table 7: List of Fixed Hyperparameters used while training.

7 Results

Once we had all the tests done, on this section it will be shown all the results obtained analytically (with the data obtained after every training) and graphically with the use of Grad-CAM.



When we started working with the simulated data (manually segmented by the team) using it as training and validation data, some problems appeared. When the training was done, for each original image, the net predicted a completely black mask, assuming that everything shown on the original image was background (see Image 15). The team reached the conclusion that this happened due to the amount of noise that appears on the training dataset.

Image 15: First results (Predicted output mask) obtained from the original imatge.

In order to try solving this issue, as a training and validation dataset the simulated data obtained from Zhao L, Fong TC, Bell MAL [35] was used. On section 6.5 it has been told that in order to modify and try to obtain better accuracy results, the hyperparameters values have been modified in order to find the best outcome to the net training.

These main modified hyperparameters have been the Initial learning rate (Ilr), the Optimizer (Solver), the Mini Batch Size (MBS) and the Max Epoch (MxE) shown on the following Table 8.

Trial	Elapsed Time (h)	Ilr	Solver	MBS	MxE	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	MASK	Weighted
1	15:26:54	0.01	adam	16	10	0.01647	99.8309	99.89	NaN	GT	No
2	7:29:36	0.01	adam	32	10	0.0259	99.745	99.89	NaN	GT	No
3	0:05:26	0.01	sgdm	16	10	NaN	NaN	NaN	NaN	GT	No
4	0:06:12	0.01	sgdm	32	10	NaN	NaN	NaN	NaN	GT	No
5	14:31:45	0.001	adam	16	10	0.0083	99.8608	99.91	NaN	GT	No
6	7:32:14	0.001	adam	32	10	0.0128	99.8283	99.90	NaN	GT	No
7	10:51:27	0.001	adam	64	10	0.060	98.9939	99.52	NaN	GT	No
8	14:09:21	0.001	sgdm	16	10	0.0102	99.8749	99.89	NaN	GT	No
9	7:32:34	0.001	sgdm	32	10	0.0095	99.8565	99.87	NaN	GT	No
10	0:07:32	0.01	sgdm	4	4	NaN	NaN	NaN	NaN	GT	No
11	1:52:34	0.01	sgdm	4	4	98.1455	0.433	89.68	3.04	GT	Yes
12	30:45:55	0.01	adam	4	4	0.2535	92.5946	96,75	NaN	GT	No
13	5:18:22	0.01	adam	64	5	0.2329	92.8338	94.69	NaN	GT	No
14	0:08:57	0.01	sgdm	64	5	NaN	NaN	NaN	NaN	GT	No
15	1:50:07	0.01	adam	4	4	97.0846	0.0672	76.39	2.528	BL	Yes
16	1:43:48	0.01	sgdm	4	4	98.6894	0.0331	79.34	5.4995	BL	Yes

Table 8:Results of nets training depending on the modified hyperparameters that has been used.

From this point of this final degree project, the number of trial will be used as the name of each trained net.

On this table there are some values obtained for each trained net with the respective modified hyperparameters. These values are obtained at the end of each respective training from a confusion matrix (see Image 16) obtained as a result from the same function that has been used for the training (**trainNetwork).**

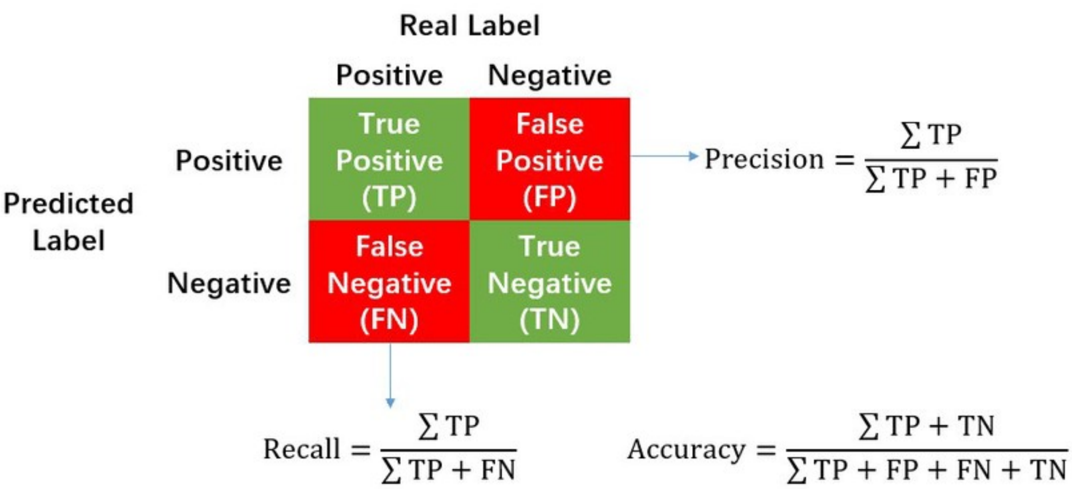


Image 16: Confusion matrix representation with formulas of Accuracy, Recall and Precision shown. [39]

It can be seen that on all of the low Training Accuracy values the Validation Loss Values are NaN. This happens because this MATLAB function returns a mean of all the Validation Losses of each iteration, if one of them has a value of 0, the final value will be NaN.

From a group of 16 tests that have been done, only 3 trained nets obtained a Training Accuracy superior to 90%. The common characteristic that these 3 have that the others don't share is the **weighting process** they have been through. In order to do so, it can be done with the following code lines.

```
tbl = countEachLabel(pxds);
imageFreq = tbl.PixelCount./tbl.ImagePixelCount;
classWeights = median(imageFreq)./imageFreq;
```

Once we have these *classWeights*, instead of using a pixelLabelDatastore [38] they should have been introduced in a *pixelClassificationLayer* [40].

By doing so, the background mask will have more protagonism than the GT mask. It has been seen that this implementation allows the network to work better and obtain more accurate results in relation with this predominant class.

7.1 Grad-CAM

On this section some of the most accurate and one of the worst accurate networks will be studied graphically with the use of Grad-CAM tool. All these following tests and plots will be acquired from the same image, Image 17.

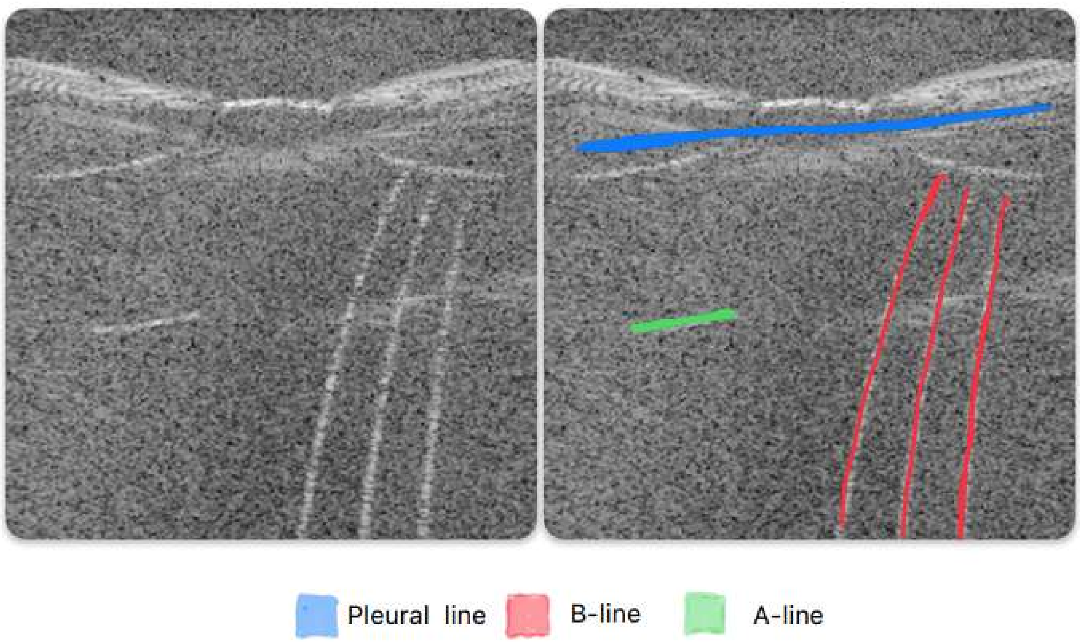


Image 17: Input Image (left) and Input Image with B-lines, Pleural line and A-line drawn (right)

On one hand, the big majority of results have very low training accuracy values. One example could be the network number 4. This network had an Initial Learning Rate of 0.001, a sgdm optimizer, a MiniBatchSize of 32 and number of MaxEpoch of 10. This network was trained with a GT mask containing all the A-lines, B-lines and Pleural lines.

The following Image 18 is the plot that appears while training the network. As it can be seen, it's almost 100% straight. Fact that tends to point into a bad training.

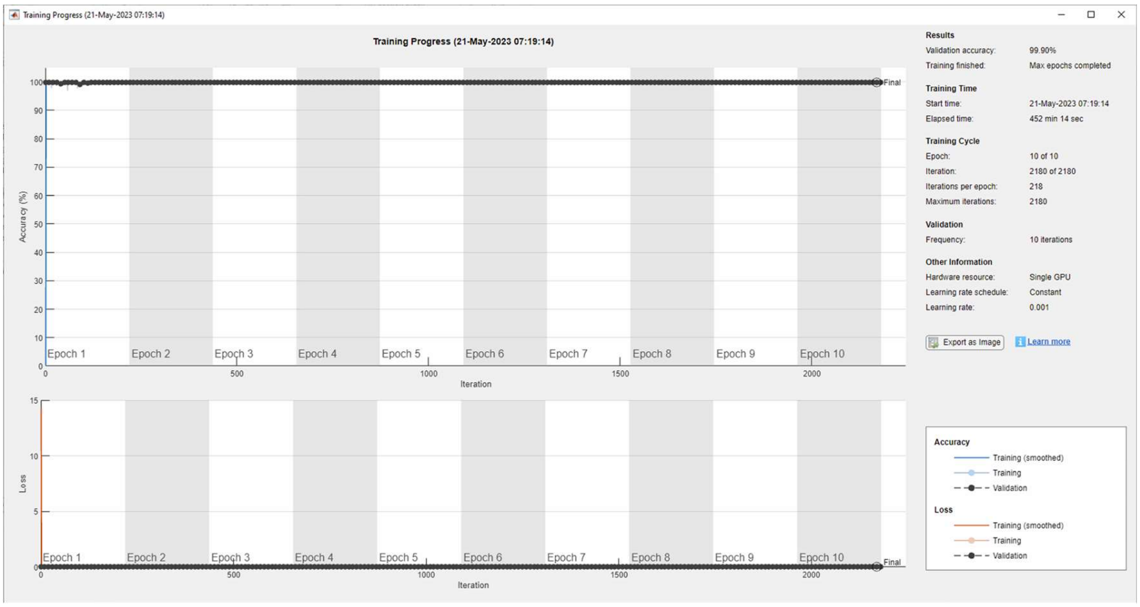
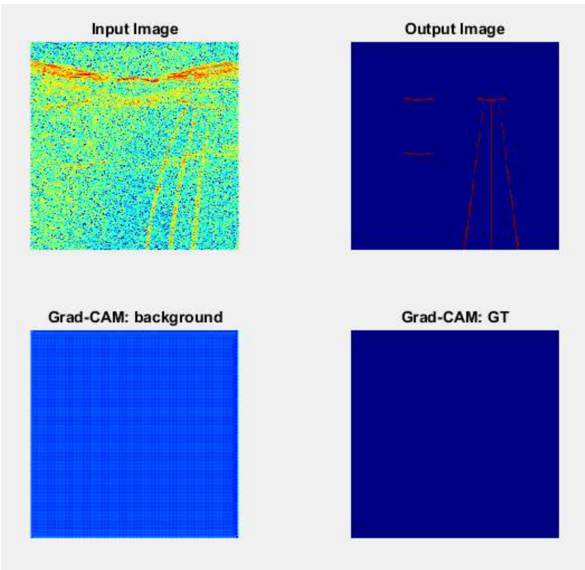


Image 18: Network Training plotting of net number 4.



Another method to be sure that the training hasn't been done properly is using Grad-CAM. The next Image 19 shows the Grad-CAM analysis of the output layer.

It can be seen that the results are quite bad because the result of predicted mask is completely blue, showing that the trained network interprets that the image is not a GT mask nor a background mask, that there are no lines to study.

Image 19: Analysis of network 4 classification output layer using Grad-CAM.

Another method to try to study the network results is to study the outputs of each weight of the network. This can be found using the function *analyzeNetwork*() [41]. The next Image 20 is the result of using Grad-CAM in these layers.

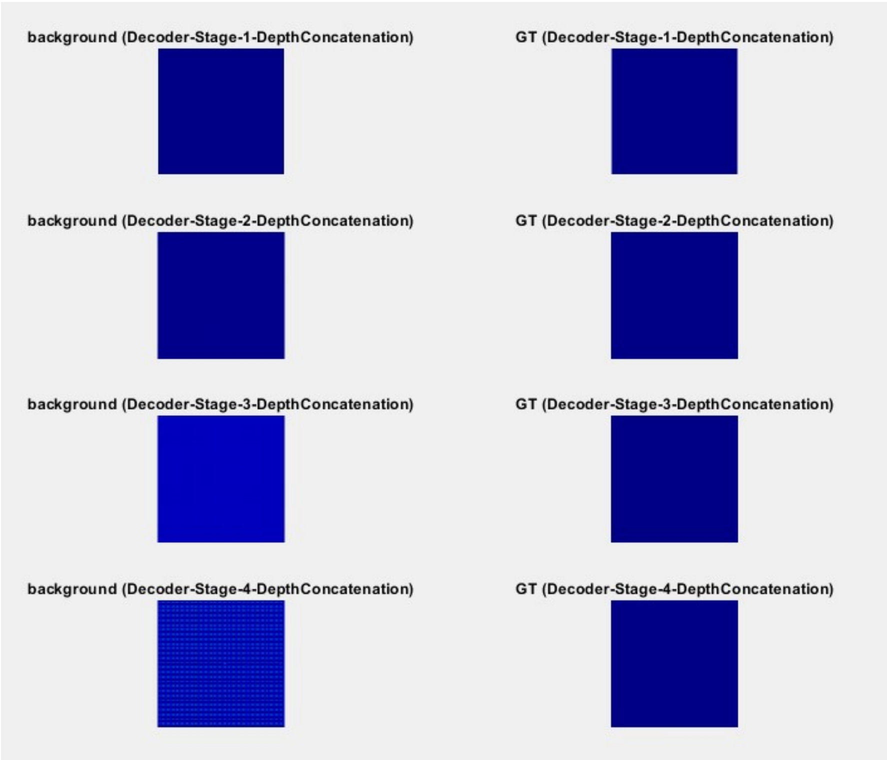


Image 20: Grad-CAM analysis of the diferent output layers found on the network.

As it can be seen, no good results appear. All the different layers are shown as completely blue which is not good. This fact is repeated with most of the trained networks.

Analysing some of the high training accuracy trained networks, it can be seen on the previously mentioned Table 8, that the ones with higher accuracy are the ones that have gone through a process of weighting (networks 11, 15 and 16).

Analysing with further attention the results obtained, these networks will be compared with the other networks with similar characteristics.

First of all, network 10 and 11 will be compared. The difference between them is the weighting process. Both have the same Initial Learning Rate, MaxEpoch, MinBatchSize and solver (*sgdm*). By applying this procedure the network 11 was able to finish the training in 1 hour and 52 minutes and obtain a traininc accuracy of 98.1455 while on the net 10 the network couldn't finish the training process. This happened due to the huge change of the accuracy values acquired between different iterations while training (seen on Image 21).

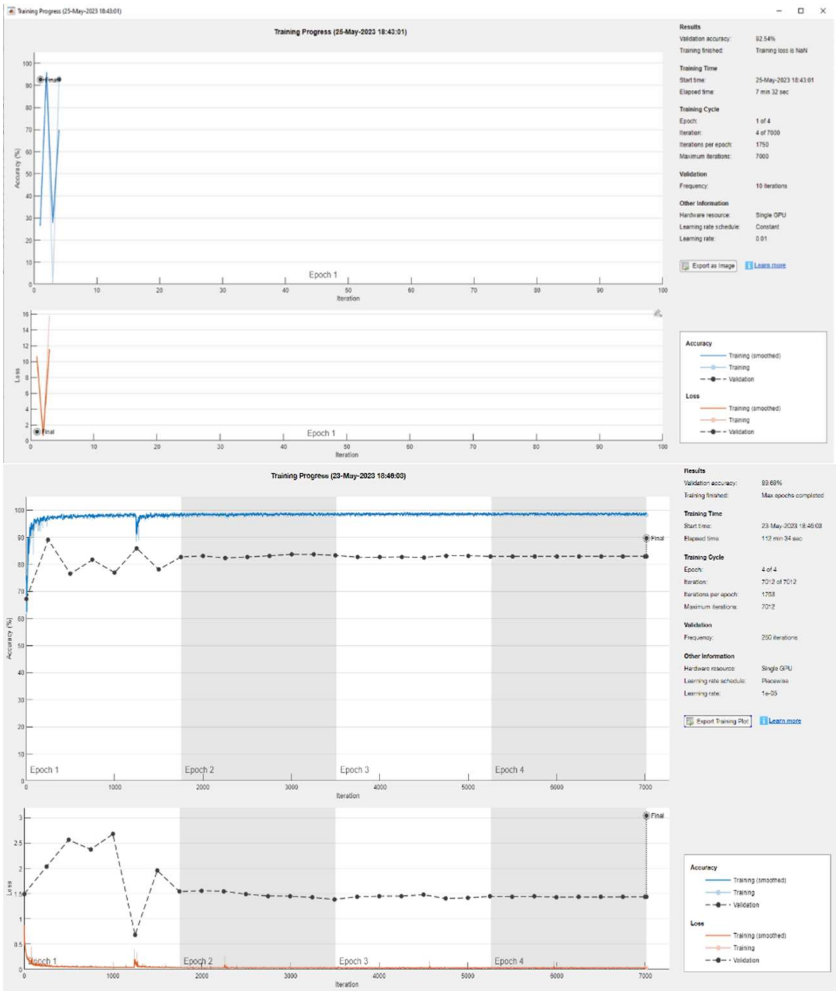


Image 21: Interrupted training plot of network 10 (superior), training plot of network 11 (inferior).

As it has been explained before, Grad-CAM is a useful tool that will be used also to compare different networks. On this case, this comparison will not be possible because in one of the two nets the training process was finished prematurely.

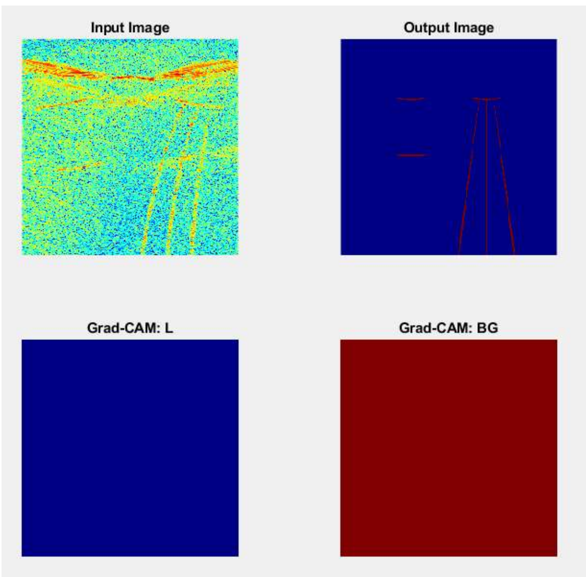


Image 22: Grad-CAM analysis of network 11 output layer

As it can be seen the result of BG (background) is completely red, this means that the network detects that all the pixels are considered as background. And the L mask is completely blue, this means that the network detects that there is not a single pixel of GT mask.

Despite having a high training accuracy, the network is not as good as it would appear only analysing the accuracy values.

Secondly, analysing two networks that have the same Initial Learning Rate, MaxEpoch, MinBatchSize and solver (adam), but have different weighting process and mask (net 12 has a GT mask while net 15 has a B-line mask) some differences can be found.

The first difference that was present was the training time, net 12 spent much more time than net 15 training and the second one was related to the Accuracy values obtained where on net 15 are above 97%.

Analysing at the Grad-CAM of the last output layer it can be seen that when a different mask is used the result doesn't change and when the weighting is used the result obtained is a complete background mask (see on Image 23).

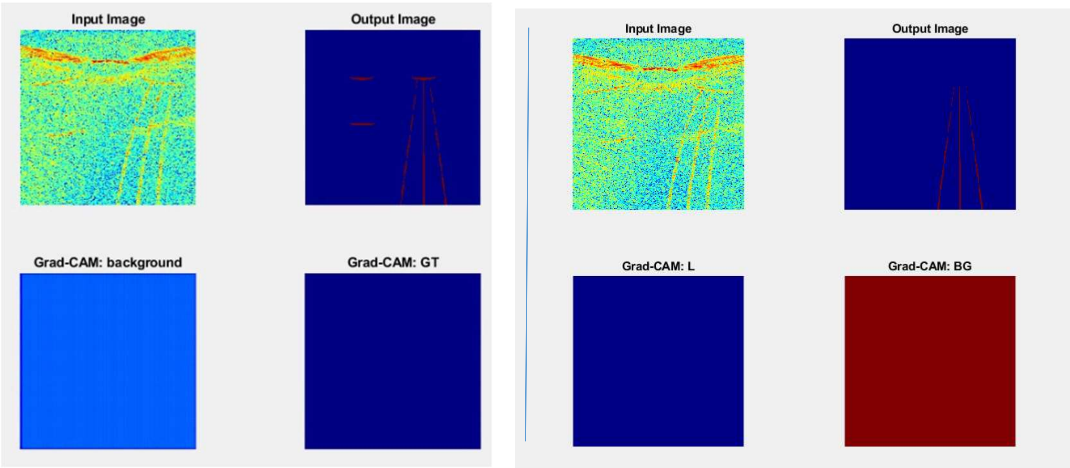


Image 23: Grad-CAM of last output layer of net 12 (left) and net 15 (right)

This phenomenon can be also found analysing the different layers found in the middle of the trained network. As it can be seen on Image 24, the layers of the net 12 (for both classes) are all completely blue and on net 15 tend only the background class tends to become red in order to finish with a complete background mask (as it was shown on Image 23).

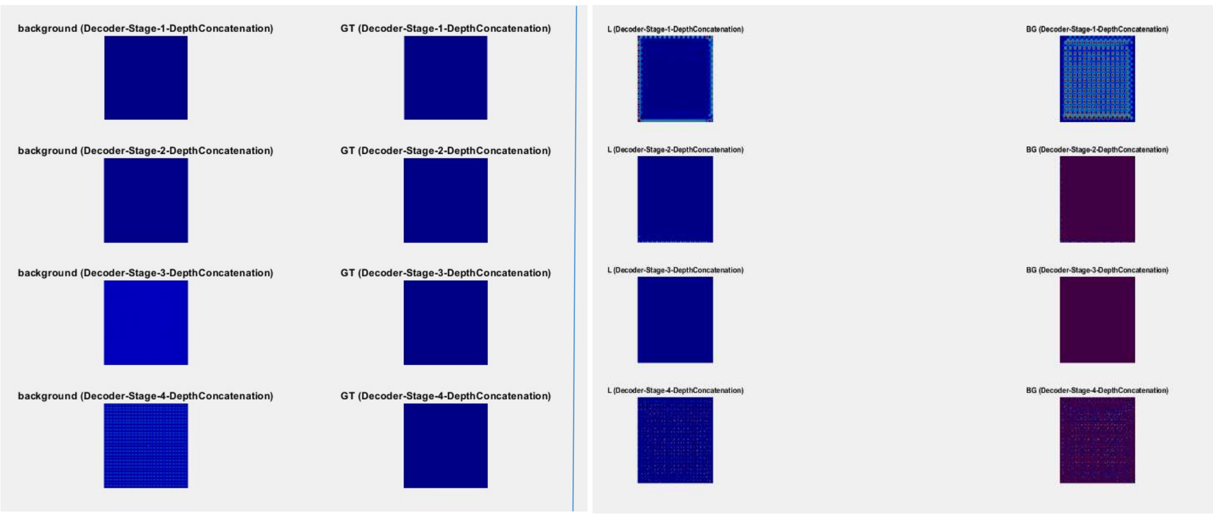


Image 24: Grad-CAM of middle high interest layers of net 12 (left) and net 15 (right)

Thirdly, analysing two networks that have the same Initial Learning Rate, MaxEpoch, MinBatchSize, solver (sgdm) and weighting process but have different masks (net 11 has a GT mask while while net 16 has a B-line mask).

As a hypothesis, the team thought that changing the mask could be a factor that incremented the accuracy. The idea behind was that if a B-line mask was used, it should be more specific than using a GT mask containing 3 different types of lines.

From this single example it cannot be affirmed nor denied that using one mask or another makes an impact sufficient to be taken in account. Both nets have a training accuracy of 98.1455 and 98.6894 respectively. As it can be seen on Image 25, the only difference between the last layers of both nets is the output image that has been used for training.

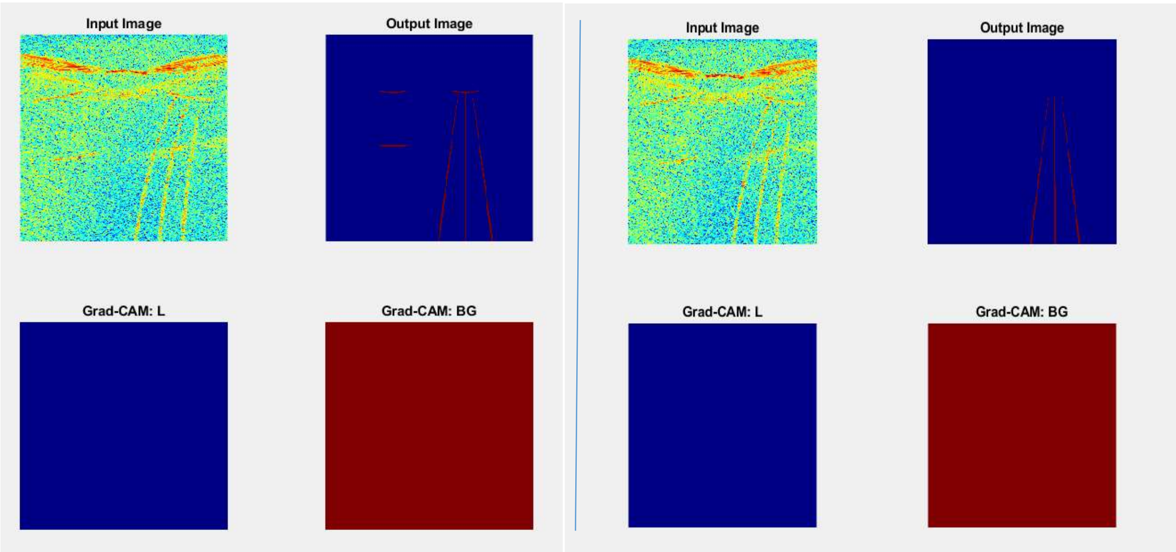


Image 25: Grad-CAM of last output layer of net 11 (left) and net 16 (right)

Talking about the rest of interesting layers, when Grad-CAM is used, the differences are minimal (see Image 26).

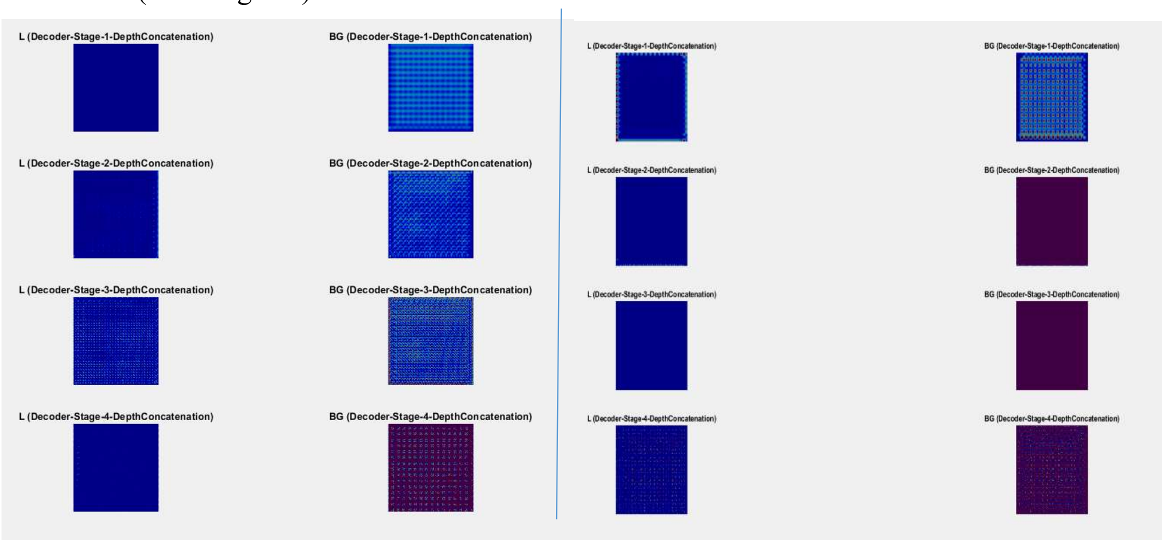


Image 26: Grad-CAM of middle high interest layers of net 11 (left) and net 16 (right)

Finally, talking about the worst outcomes. Some of the networks shown on the Table 8 are coloured as red. As it has been explained with the net 10, this means that while they were training, something happened that made the training stop.

As it can be seen, after comparing the different high training accuracy values with the respective low values it can be concluded one thing. Despite having a very high accuracy, on these experiments, these values don't correspond to the reality.

When the weighting was done, the network was trained taking in account that the background mask was more present, and considering it as more important than GT/B mask. The high training and validation accuracy values that appear are mostly because all of the predicted masks of these nets are considered as background and it has just been explained, the network understood that these class is the most important one.

8 Discussion

8.1 Limitations

After all these months of hard work, not all the main objectives of this project have been accomplished. After training with different hyperparameters it has been seen that finding good options for the training can be quite tricky. Despite not acquiring a real good accuracy result of the predicted masks, with this project it has been learned how to train a U-Net network with new data and how to adapt it to the problem that has been studied.

The main issue has been the time for training the nets and try to find the better hyperparameters configuration. This could have been prevented by having a better system with high GPU earlier, given that I'm not a student of the UPC this administrative procedure took more time than it should.

Working with a thesis and a master student provided me with much experience but it also gave me more difficult problems to solve, and sometimes they took more time for me to solve them. During the thesis the project has been forcedly simplified in order to finish it before the dead-line (for example, changing the training data, changing the division method for training, changing what kind of network should be used…) With these changes many time invested was "lost" at the end of the project because many efforts focused in one concrete issue, weren't reflected on the final methodology of this project.

8.2 SDO Contribution

Inside the 17 Sustainable Development Goals (SDG) from the UN, this project would accomplish the objectives mentioned in point number 3, Good Health and Well-Being.

This point talks about ensuring healthy lives and promoting well-being at all ages, being essential to a sustainable development. This project could help the scientific community by showing what hyperparameters don't allow a U-net network to obtain high accuracy values while segmenting LUS images. As it has been said on the introduction, these results are going to be helpful on a PhD thesis that has the objective to find what is the best approach in order to find a mask of A-lines, B-lines and Pleural line.

By combining a good method of detecting these mentioned patterns and a ultrasound, many people could be diagnosed faster and people from underdeveloped countries could have better access to detecting lung diseases with cheaper machinery and with less formation.

9 Conclusion

Taking in account the initial objectives, that where being able to differentiate background and the mask for each frame extracted from an ultrasound video, retrain a U-Net network and determine if the combination of Deep Learning and Lung Ultrasound Segmentation is a good procedure with high accuracy, I could say that not all of them have been accomplished.

It hasn't been possible to find a method, using Deep Learning and Lung Ultrasound Segmentation, which obtains a real high accuracy value. What has been found was a group of networks that are able to create a predicted mask containing almost 100% of pixels determined as background.

After understanding how the training networks had gone and why the high accuracy values can't be taken in account as good results, it could not be affirmed that the networks can segment properly a GT/B mask from the background.

In order to determine if Deep Learning combined with Semantic Segmentation of Lung Ultrasound could be a good technique, more investigation should be done with different datasets, hyperparameters and other factors.

As an improvement, more tests should have been performed with different hyperparameters and datasets. Another improvement could have been, despite having used Grad-CAM, finding more methods to analyse and understand the algorithm that the neural network followed.

The results of this project will be useful for Maria Farahi's thesis where she will continue exploring different paths in order to obtain the optimum method for Lung Ultrasound Segmentation to, in further phases of the investigation, assist and help diagnose lung pathologies such as Pneumothorax, Pneumonia or Covid-19.

10 References

[1] Cleveland Clinic. Lungs: Location, Anatomy, Function & Complications [Internet]. Cleveland Clinic. 2022. Available at: https://my.clevelandclinic.org/health/body/8960-lungs

[2] Lung Health & Diseases | American Lung Association [Internet]. www.lung.org. Available at: https://www.lung.org/lung-health-diseases/lung-disease-lookup

[3] Lung Disease A - Z | the lung association [Internet]. Lung.ca. 2012. Available at: https://www.lung.ca/lung-health/lung-disease

[4] World Health Organization. WHO COVID-19 dashboard [Internet]. World Health Organization. 2023. Available at: https://covid19.who.int/

[5] American Lung Association. Preventing Pneumonia | American Lung Association [Internet]. www.lung.org. 2020. Available at: https://www.lung.org/lung-health-diseases/lung-disease-lookup/pneumonia/preventing-pneumonia

[6] COVID-19 | American Lung Association [Internet]. www.lung.org. Available at: https://www.lung.org/lung-health-diseases/lung-disease-lookup/covid-19

[7] CDC. Coronavirus Disease 2019 (COVID-19) – Symptoms [Internet]. Centers for Disease Control and Prevention. CDC; 2022. Available at: https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html

[8] Yu C. What Does COVID-19 Do to Your Lungs? [Internet]. WebMD. 2023. Available at: https://www.webmd.com/covid/what-does-covid-do-to-your-lungs

[9] Pneumonia | American Lung Association | American Lung Association [Internet]. www.lung.org. Available t: https://www.lung.org/lung-health-diseases/lung-disease-lookup/pneumonia

[10] Pneumonia | the lung association [Internet]. Lung.ca. 2014. Available at: https://www.lung.ca/lung-health/lung-disease/pneumonia

[11] Association AL. Pneumothorax (Collapsed Lungs) [Internet]. www.lung.org. Available at: https://www.lung.org/lung-health-diseases/lung-disease-lookup/pneumothorax

[12] Pneumothorax: Practice Essentials, Background, Anatomy [Internet]. Medscape.com. 2019. Available at: https://emedicine.medscape.com/article/424547-overview

[13] Moshavegh R, Hansen KL, Moller-Sorensen H, Nielsen MB, Jensen JA. Automatic detection of B-lines in in vivo lung ultrasound. IEEE Trans Ultrason Ferroelectr Freq Control [Internet]. 2019;66(2):309–17. Available at: http://dx.doi.org/10.1109/TUFFC.2018.2885955

[14] Lichtenstein DA. Lung ultrasound in the critically ill. Annals of Intensive Care. 2014;4(1):1.

[15] Segmentación semántica [Internet]. es.mathworks.com. Available from: https://es.mathworks.com/solutions/image-video-processing/semantic-segmentation.html

[16] Introducción a la transferencia del aprendizaje [Internet]. Mathworks.com. Available from: https://es.mathworks.com/help/deeplearning/gs/get-started-with-transfer-learning.html

[17] Transfer Learning [Internet]. Mathworks.com. Available at: https://es.mathworks.com/discovery/transfer-learning.html

[18] TrainingOptions [Internet]. Mathworks.com. Available at: https://es.mathworks.com/help/deeplearning/ref/trainingoptions.html

[19] ¿Qué es una red neuronal convolucional? [Internet]. Mathworks.com. Available at: https://es.mathworks.com/discovery/convolutional-neural-network-matlab.html

[20] Social Events. U-Net: Convolutional Networks for Biomedical Image Segmentation [Internet]. Uni-freiburg.de. Available at: https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

[21] Researchgate.net. Available at: https://www.researchgate.net/figure/Illustration-of-Max-Pooling-and-Average-Pooling-Figure-2-above-shows-an-example-of-max_fig2_333593451

[22] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. En: Lecture Notes in Computer Science. Cham: Springer International Publishing; 2015. p. 234–41.

[23] Researchgate.net. Available at: https://www.researchgate.net/figure/Confusion-matrix-and-performance-evaluation-metrics_fig5_340939027

[24] Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual explanations from deep networks via Gradient-based localization [Internet]. arXiv [cs.CV]. 2016. Available at: http://arxiv.org/abs/1610.02391

[24] Liu S, Wang Y, Yang X, Lei B, Liu L, Li SX, et al. Deep Learning in Medical Ultrasound Analysis: A Review. Engineering. 2019 Apr;5(2):261–75.

[25] Roy S, Menapace W, Oei S, Luijten B, Fini E, Saltori C, et al. Deep learning for classification and localization of COVID-19 markers in point-of-care lung ultrasound. IEEE

Trans Med Imaging [Internet]. 2020;39(8):2676–87. Available at: http://dx.doi.org/10.1109/TMI.2020.2994459

[26] Born J, Brändle G, Cossio M, Disdier M, Goulet J, Roulin J, et al. POCOVID-Net: Automatic detection of COVID-19 from a new lung ultrasound imaging dataset (POCUS) [Internet]. arXiv [eess.IV]. 2020. Available at: http://arxiv.org/abs/2004.12084

[27] Horry MJ, Chakraborty S, Paul M, Ulhaq A, Pradhan B, Saha M, et al. COVID-19 detection through transfer learning using multimodal imaging data. IEEE Access [Internet]. 2020;8:149808–24. Available at: http://dx.doi.org/10.1109/ACCESS.2020.3016780

[28] Born J, Wiedemann N, Cossio M, Buhre C, Brändle G, Leidermann K, et al. Accelerating detection of lung pathologies with explainable ultrasound image analysis. Appl Sci (Basel) [Internet]. 2021; 11(2):672. Available at: https://www.mdpi.com/2076-3417/11/2/672

[29] Arntfield R, VanBerlo B, Alaifan T, Phelps N, White M, Chaudhary R, et al. Development of a convolutional neural network to differentiate among the etiology of similar appearing pathological B lines on lung ultrasound: a deep learning study. BMJ Open [Internet]. 2021;11(3):e045120. Available at: http://dx.doi.org/10.1136/bmjopen-2020-045120

[30] Baum ZMC, Bonmati E, Cristoni L, Walden A, Prados F, Kanber B, et al. Image quality assessment for closed-loop computer-assisted lung ultrasound [Internet]. arXiv [eess.IV]. 2020. Available at: http://arxiv.org/abs/2008.08840

[31] Huang S-Y, Hsu W-L, Hsu R-J, Liu D-W. Fully convolutional network for the semantic segmentation of medical images: A survey. Diagnostics (Basel) [Internet]. 2022;12(11):2765. Available at: https://www.mdpi.com/2075-4418/12/11/2765/pdf

[33] Born J. covid19_ultrasound: Open source lung ultrasound (LUS) data collection initiative for COVID-19. Available at: https://github.com/jannisborn/covid19_ultrasound

[34] Ketelaars R, Gülpinar E, Roes T, Kuut M, van Geffen GJ. Which ultrasound transducer type is best for diagnosing pneumothorax? Crit Ultrasound J [Internet]. 2018;10(1):27. Available at: http://dx.doi.org/10.1186/s13089-018-0109-0

[35] Zhao L, Fong TC, Bell MAL. COVID-19 feature detection with deep neural networks trained on simulated lung ultrasound B-mode images. En: 2022 IEEE International Ultrasonics Symposium (IUS). IEEE; 2022.

[36] unetLayers [Internet]. Mathworks.com. Available at: https://es.mathworks.com/help/vision/ref/unetlayers.html

[37] trainNetwork [Internet]. Mathworks.com. Available at: https://es.mathworks.com/help/deeplearning/ref/trainnetwork.html

[38] Datastore for pixel label data - MATLAB - MathWorks España [Internet]. Mathworks.com. Available at: https://es.mathworks.com/help/vision/ref/pixellabeldatastore.html

[39] Researchgate.net. [citado el 28 de mayo de 2023]. Disponible en: https://www.researchgate.net/figure/Calculation-of-Precision-Recall-and-Accuracy-in-the-confusion-matrix_fig3_336402347

[40] Create pixel classification layer for semantic segmentation - MATLAB - MathWorks España [Internet]. Mathworks.com. Available at: https://es.mathworks.com/help/vision/ref/nnet.cnn.layer.pixelclassificationlayer.html

[41] analyzeNetwork [Internet]. Mathworks.com. Available at: https://es.mathworks.com/help/deeplearning/ref/analyzenetwork.html

ANNEX A: Planning

Step 1: Project Preparation

Approximated Duration: 2 weeks

On this part of the project I tried with my UdG tutor to stay in contact with the UPC department where I wanted to work with. After many e-mails I managed to enter into it.

Step 2: Data preparation

Approximated Duration: 7 weeks

When I managed to enter to the group I had to learn many things in the less time possible. First of all I had to start learning what A-lines, B-lines and Pleural line are in order to start preparing the simulated data.

Step 3: Search generic information about Deep Learning uses in Matlab

Approximated Duration: 8 weeks

I started writing reports about what I could find in order to present them on the weekly meetings I had with Maria Farahi. Here I found information about how to use different MATLAB functions used in order to retrain a network, how to freeze layers…

Step 4: Dividing data strategies

Approximated Duration: 4 weeks

Despite having many problems with this part, I finally managed to divide the data and start training test nets with 7 images as train and 3 as validation. Finally this task didn't get to be used for this final degree project.

Step 5: Testing the nets with less data

Approximated Duration: 4 weeks

Tests trying to make the network work properly in order to obtain further good results.

Step 6: Training the net

Approximated Duration: 4 weeks

After having many problems in order to train properly due to the poor access to a good system. UPC finally provided me with a system that could work properly and the training could be done. This was one of the longest part because it consisted in changing and modifying the hyperparameters in order to find the optimum values to have the highest accuracy possible.

Step 7: Analyse results

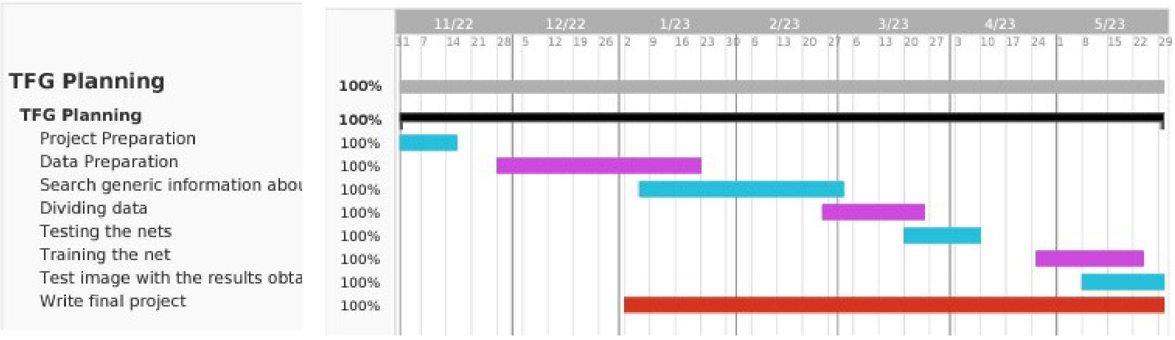
Approximated Duration: 2 weeks

With the firsts results obtained while training the networks, I started analysing the results through different techniques such as Deep Learning.

Step 8: Write memory

Approximated Duration : 3 months

While all the other tasks, the memory was being written at the same time.



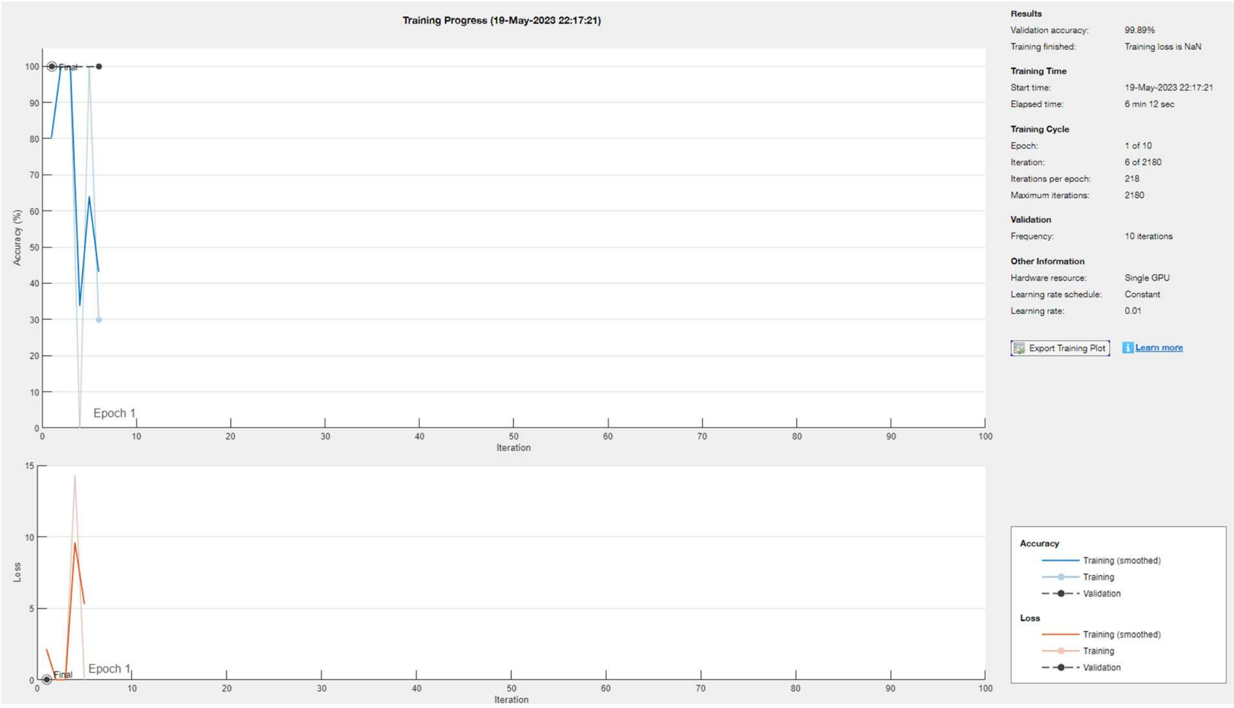
ANNEX B: Code

This project contains some code that can be found on a GitHub repository if is needed in the future by anyone.

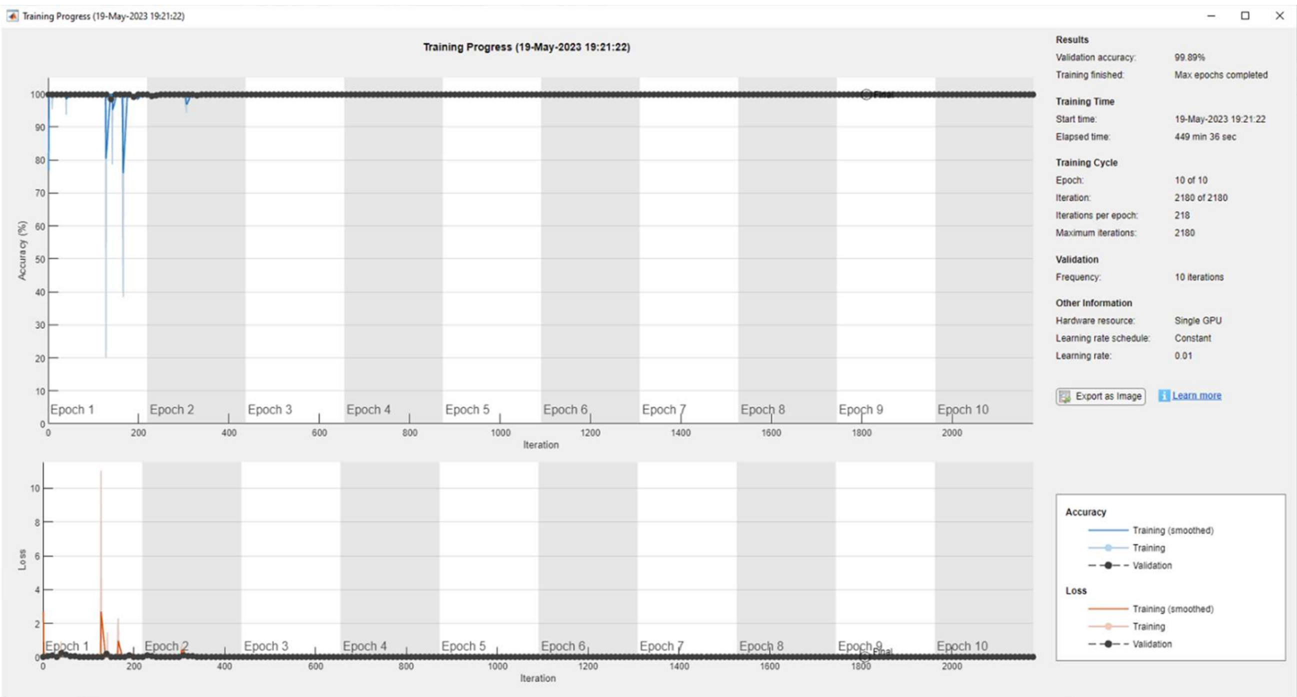
Link: https://github.com/FerranFrancoiMoral/TFG

ANNEX C: Network Training Plots

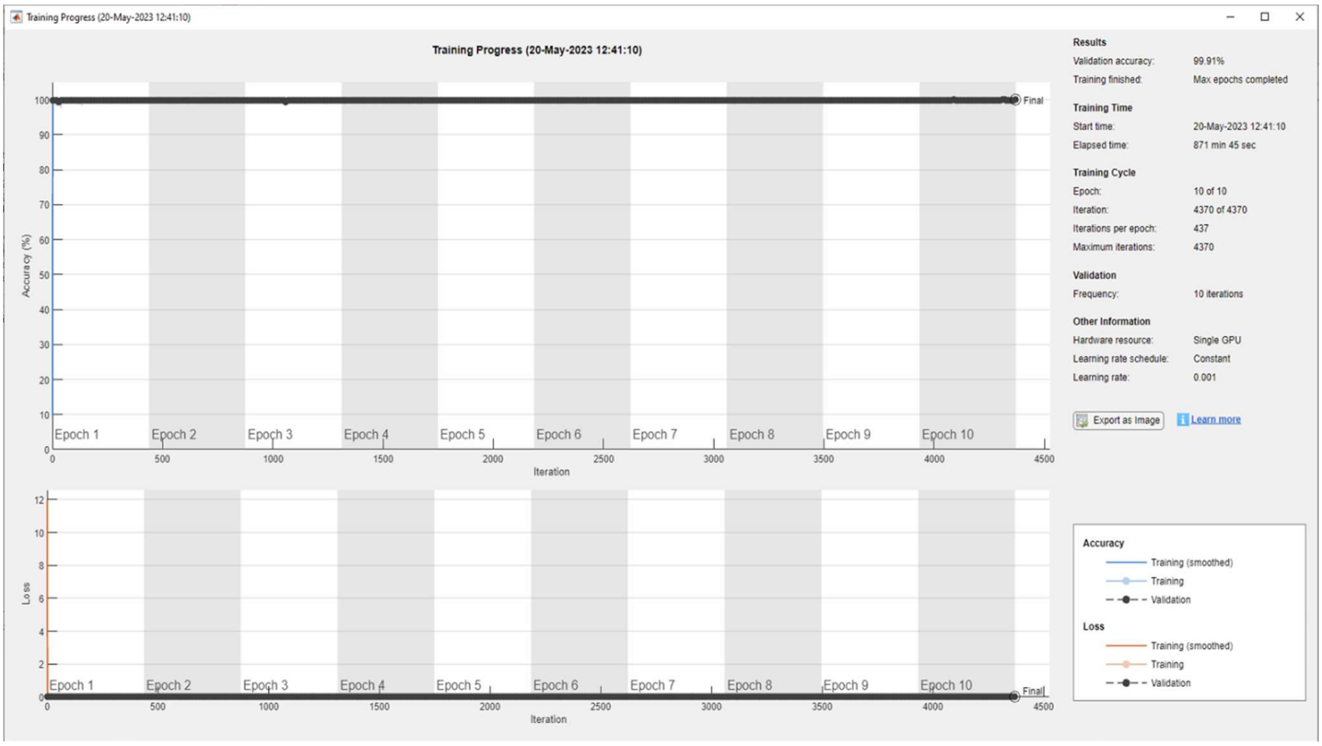
Training on the System b (see references on section 6.4) with an Initial Learning Rate of 0.01, using sgdm as the optimizer, a batch size of 32, a MaxEpoch value of 10 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). No weighting applied.



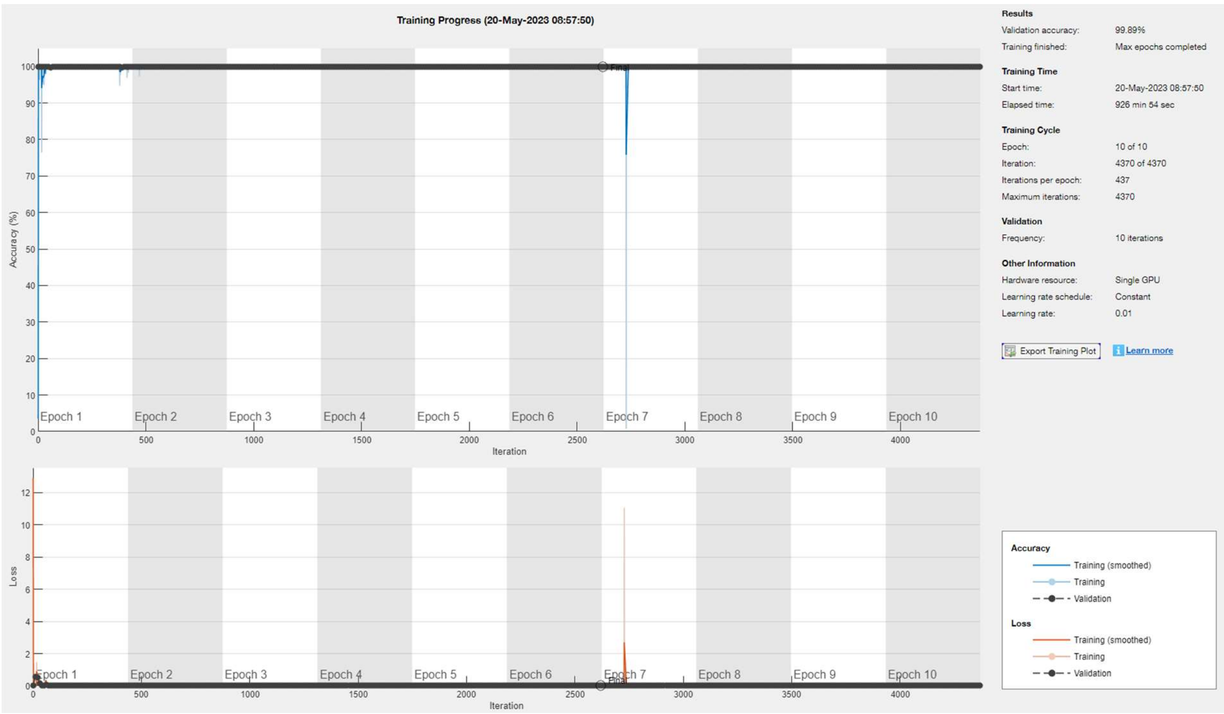
Training on the System a (see references on section 6.4) with an Initial Learning Rate of 0.01, using adam as the optimizer, a batch size of 32, a MaxEpoch value of 10 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). No weighting applied.



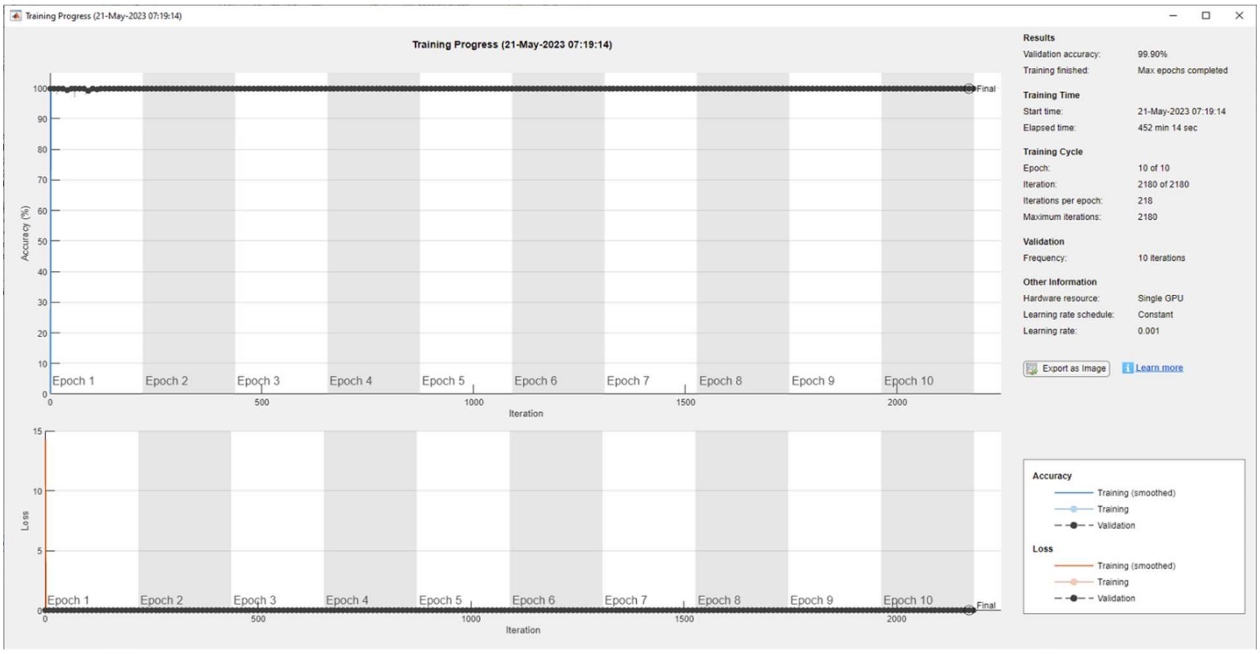
Training on the System a (see references on section 6.4) with an Initial Learning Rate of 0.001, using adam as the optimizer, a batch size of 16, a MaxEpoch value of 10 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). No weighting applied.



Training on the System b (see references on section 6.4) with an Initial Learning Rate of 0.01, using adam as the optimizer, a batch size of 16, a MaxEpoch value of 10 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). No weighting applied.

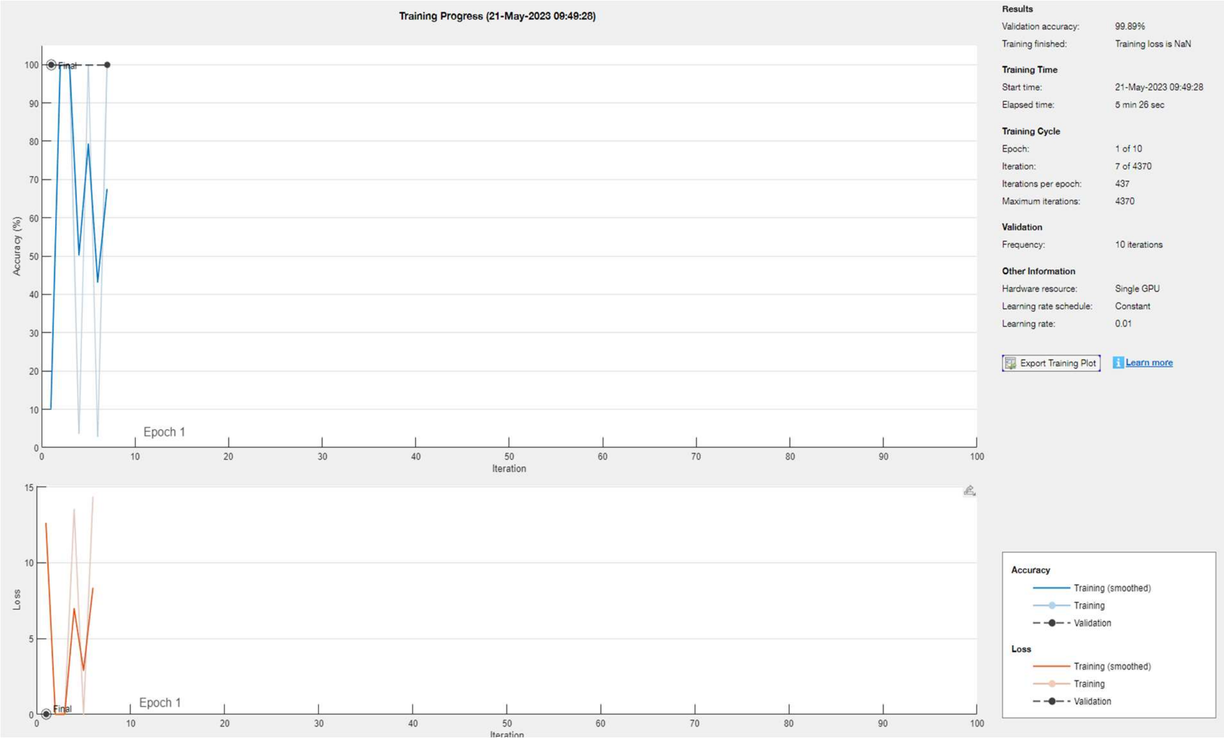


Training on the System a (see references on section 6.4) with an Initial Learning Rate of 0.001, using adam as the optimizer, a batch size of 32, a MaxEpoch value of 10 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). No weighting applied.

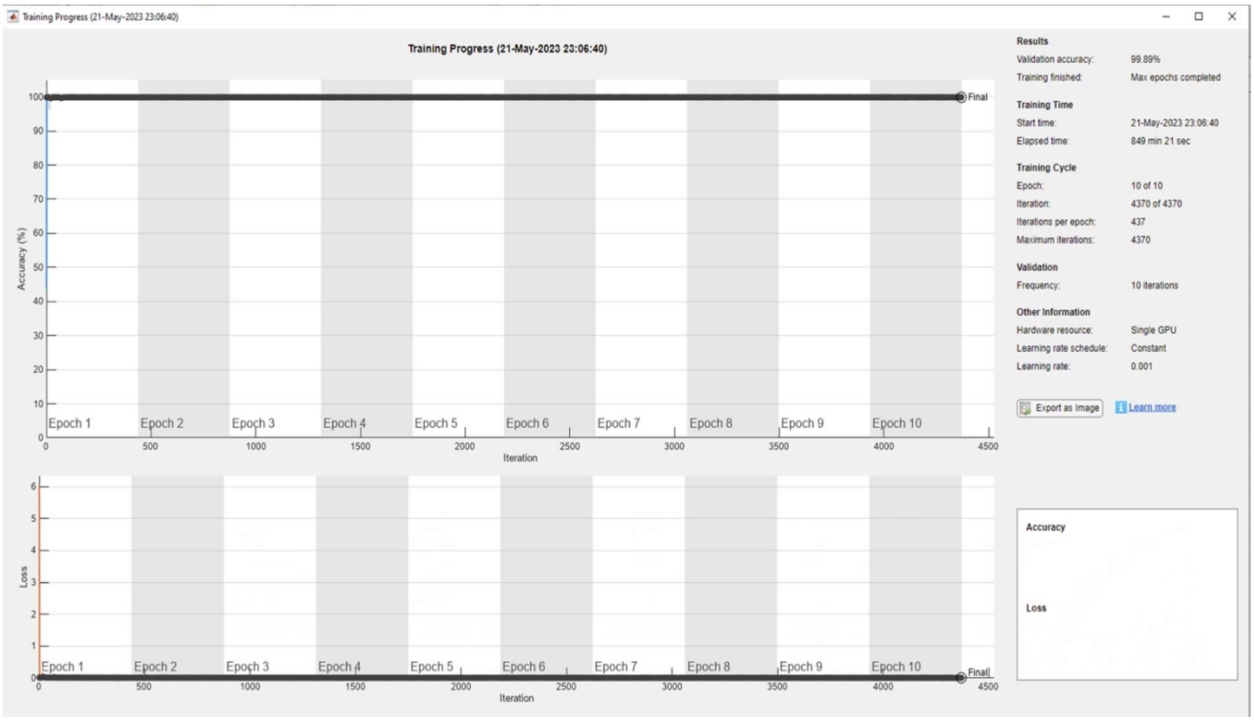


Training on the System a (see references on section 6.4) with an Initial Learning Rate of 0.001, using sgdm as the optimizer, a batch size of 32, a MaxEpoch value of 10 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). No weighting applied.

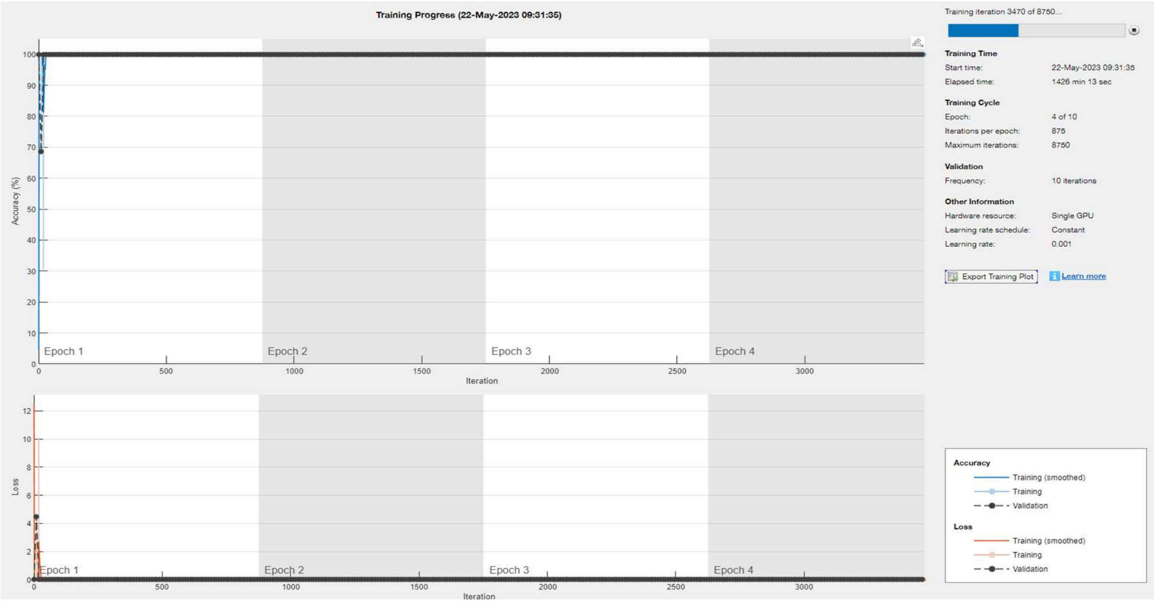
Training on the System b (see references on section 6.4) with an Initial Learning Rate of 0.01, using sgdm as the optimizer, a batch size of 16, a MaxEpoch value of 10 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). No weighting applied.



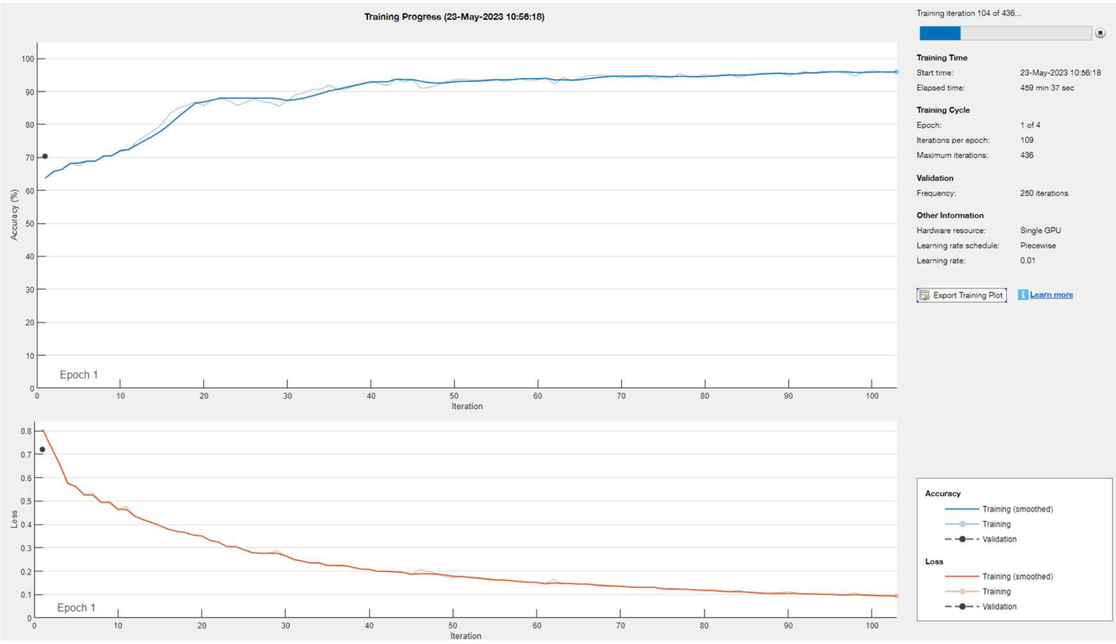
Training on the System a (see references on section 6.4) with an Initial Learning Rate of 0.001, using sgdm as the optimizer, a batch size of 16, a MaxEpoch value of 10 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). No weighting applied.



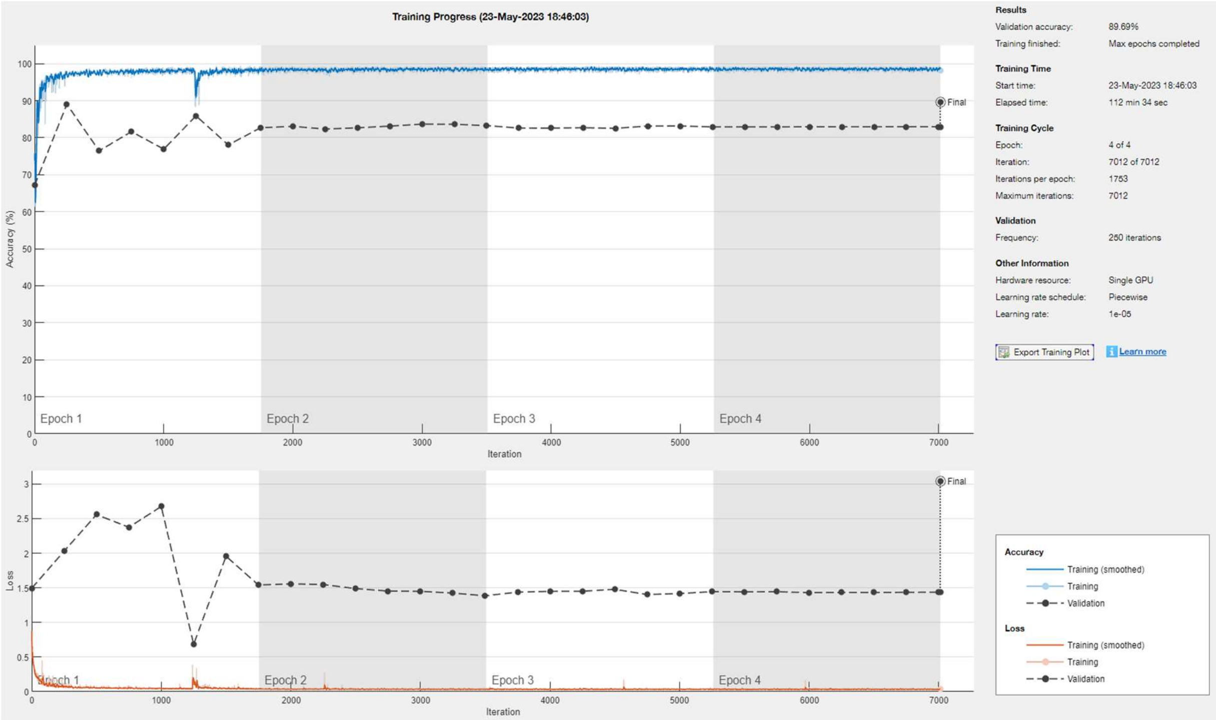
Training on the System b (see references on section 6.4) with an Initial Learning Rate of 0.001, using sgdm as the optimizer, a batch size of 8, a MaxEpoch value of 10 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). Stopped because it was highly time consuming and was following the same pattern as the ones with lowest results already obtained. No weighting applied.



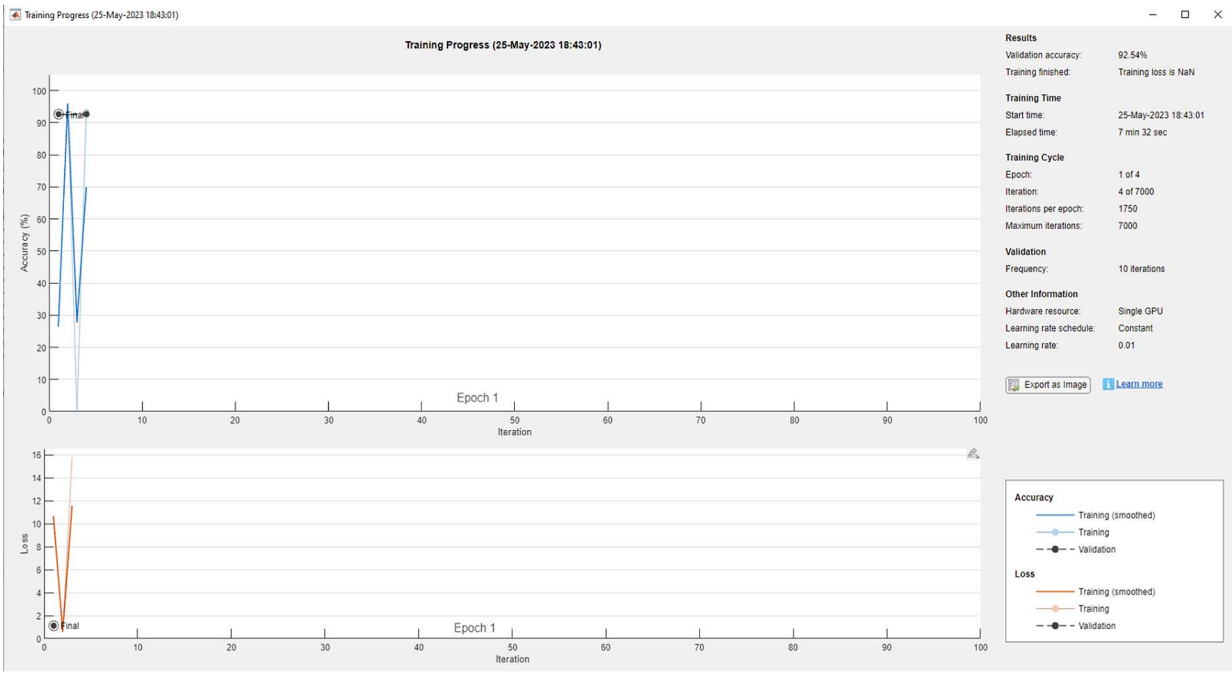
Training on the System b (see references on section 6.4) with an Initial Learning Rate of 0.01, using sgdm as the optimizer, a batch size of 64, a MaxEpoch value of 4 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). Stopped because it was highly time consuming and was following the same pattern as the ones with lowest results already obtained. No weighting applied.



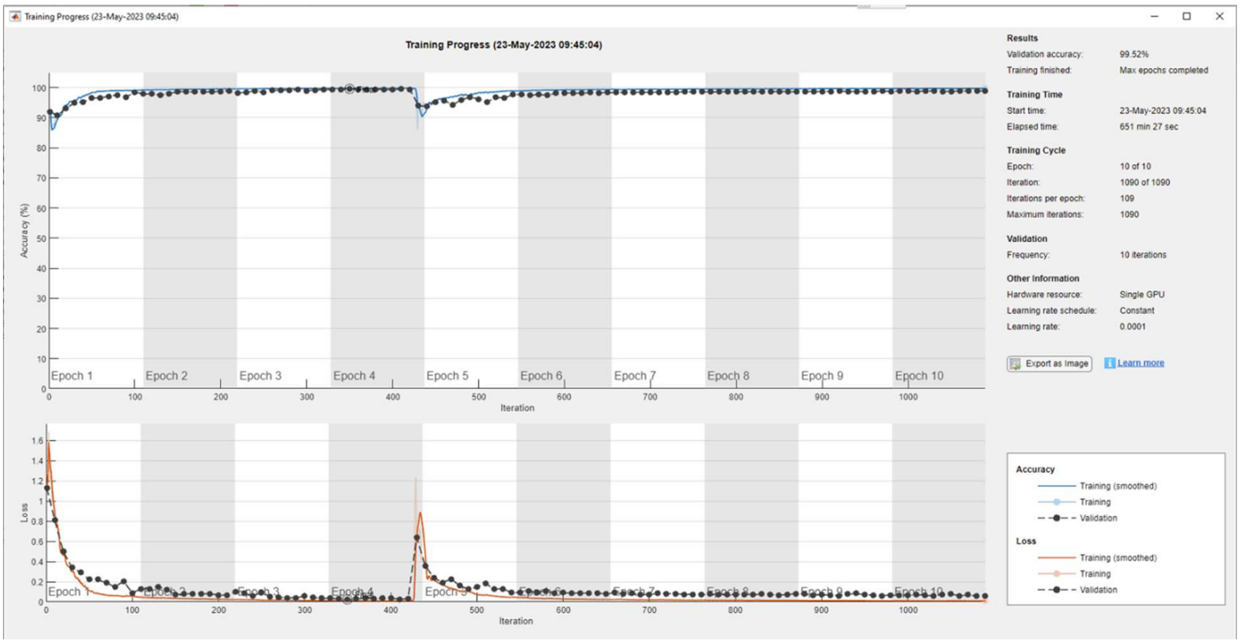
Training on the System b (see references on section 6.4) with an Initial Learning Rate of 0.01, using sgdm as the optimizer, a batch size of 4, a MaxEpoch value of 4 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). Weighting applied.



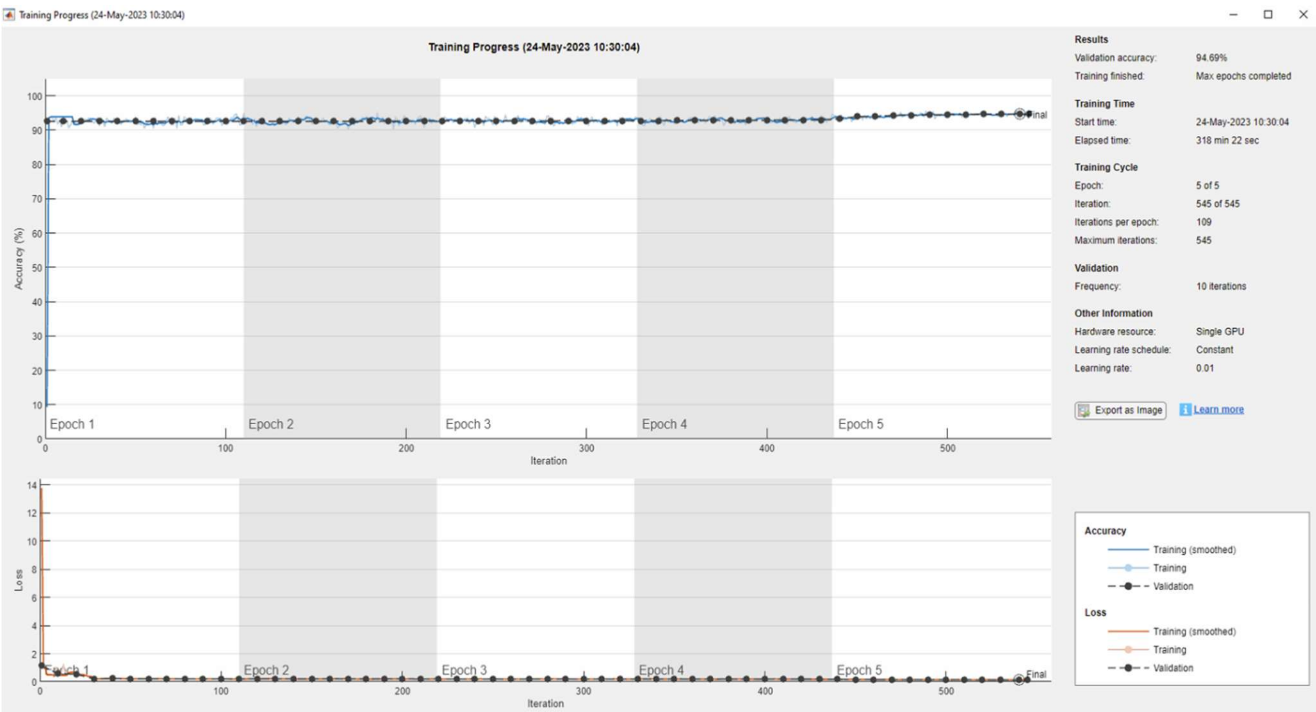
Training on the System a (see references on section 6.4) with an Initial Learning Rate of 0.01, using sgdm as the optimizer, a batch size of 4, a MaxEpoch value of 4 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). No weighting applied.



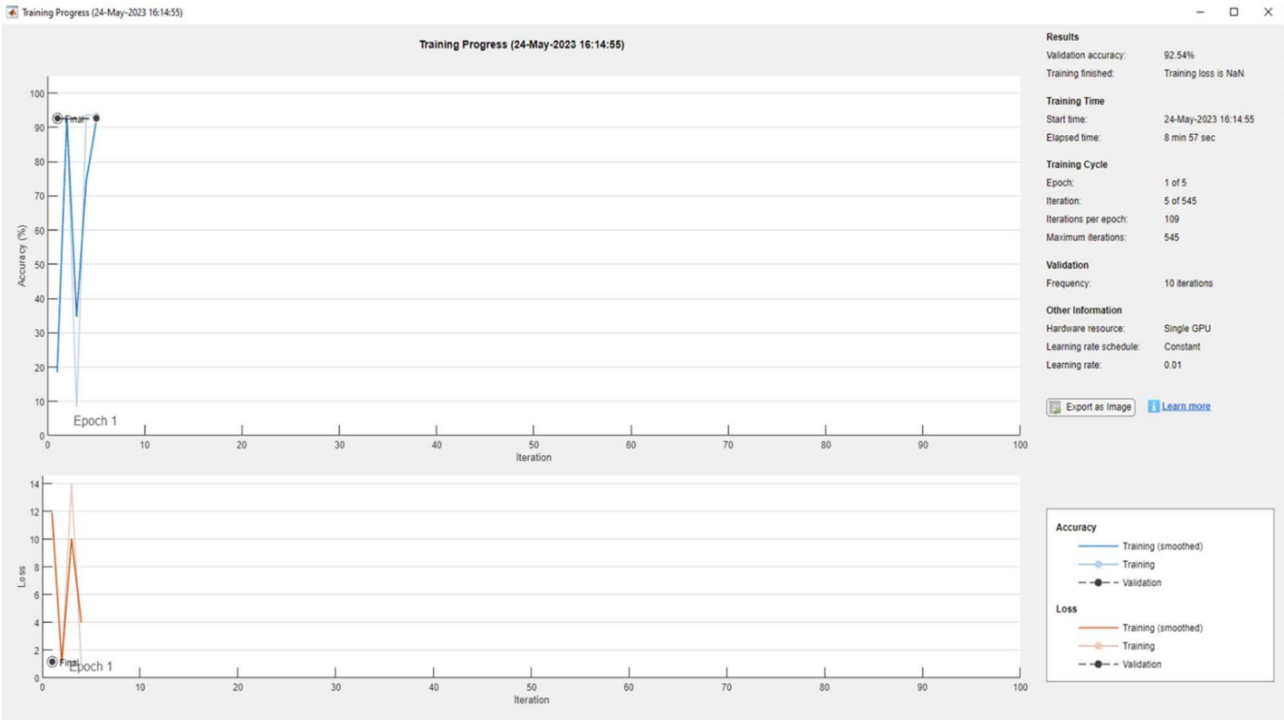
Training on the System a (see references on section 6.4) with an Initial Learning Rate of 0.0001, using adam as the optimizer, a batch size of 64, a MaxEpoch value of 10 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). No weighting applied.



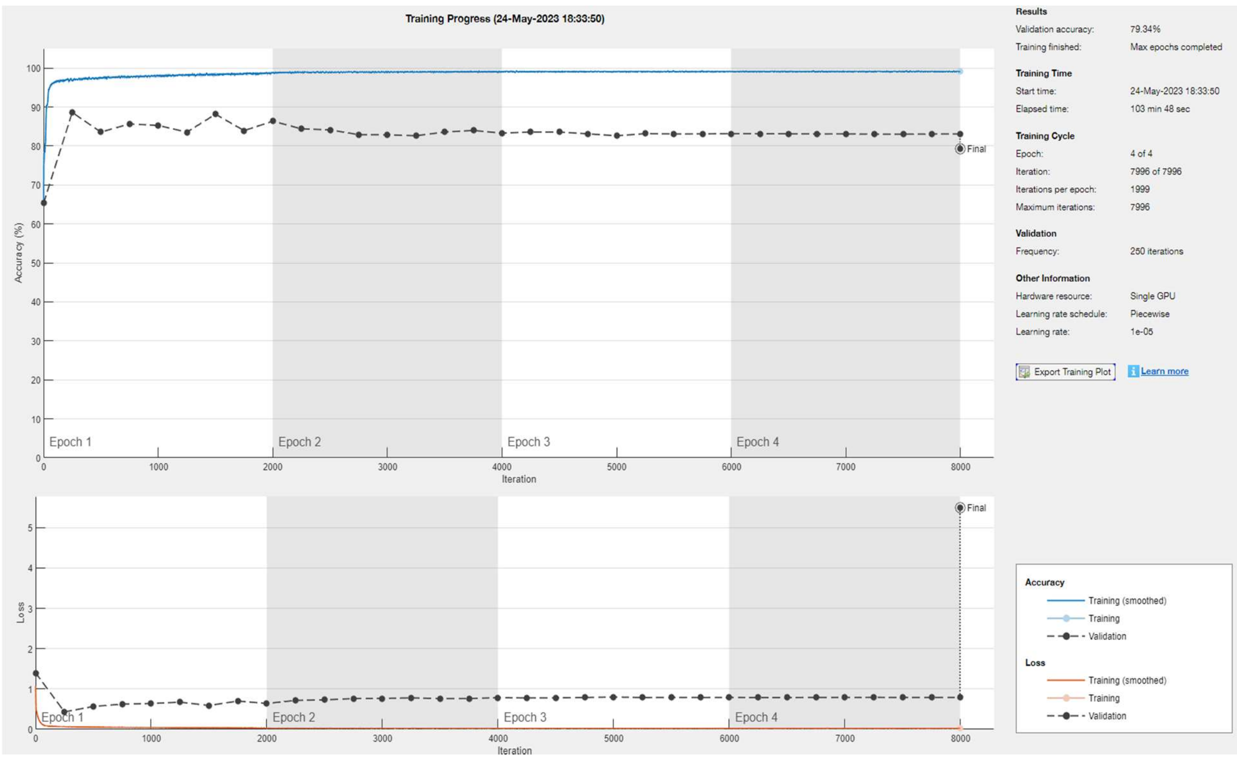
Training on the System a (see references on section 6.4) with an Initial Learning Rate of 0.01, using adam as the optimizer, a batch size of 64, a MaxEpoch value of 5 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). No weighting applied.



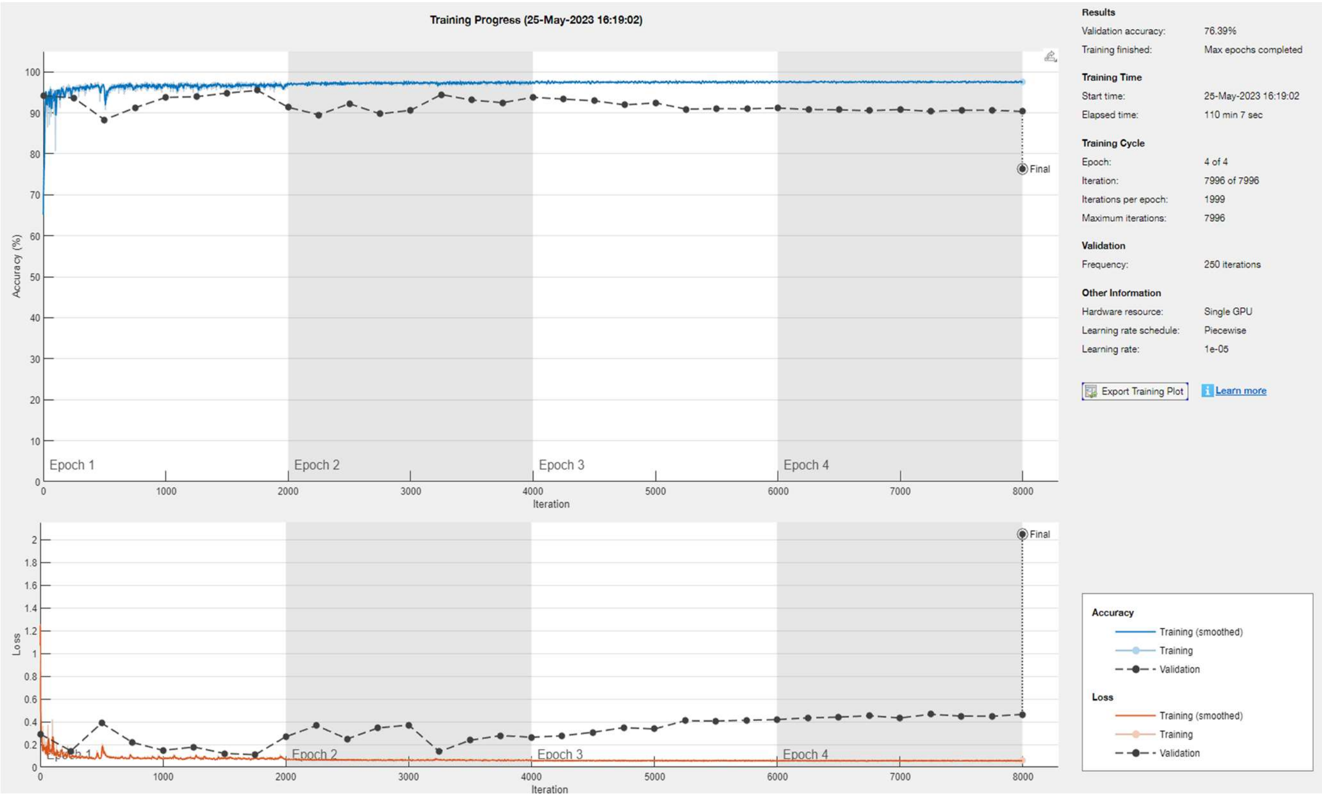
Training on the System a (see references on section 6.4) with an Initial Learning Rate of 0.01, using adam as the optimizer, a batch size of 64, a MaxEpoch value of 5 and an output mask containing A-lines, B-lines and Pleural lines (GT mask). No weighting applied.



Training on the System b (see references on section 6.4) with an Initial Learning Rate of 0.01, using sgdm as the optimizer, a batch size of 4, a MaxEpoch value of 4 and an output mask containing B-lines (B-line mask). Weighting applied.



Training on the System b (see references on section 6.4) with an Initial Learning Rate of 0.01, using adam as the optimizer, a batch size of 4, a MaxEpoch value of 4 and an output mask containing B-lines (B-line mask). Weighting applied.



ANNEX D: Budget

On this Annex C it will appear the estimated budged required for this project. It has to be taken account that, being a final degree project done inside two institutions, software license and high GPU systems have been provided.

D.1. Handwork

Taking account that this final degree project has taken about 400 working hours approximately, and an Artificial Intelligence Engineer salary is approximately 40€/hour:

Personal	Salary / hour	Total
Artificial Intelligence Engineer	40 €	16.000 €

D.2. Resources

Despite having the software license and a system provided by both universities, an estimation of the price will be added on this Annex C.

Resource	Quantity	Price
MATLAB_R2022b	1	2.000€
UPC computer and electricity	2 computers + electricity	5.500€

D.3. Final Budget

On the following table it will be shown the total estimated budged of this project:

Description	Hours	Salary / hour	Quantity	Cost
Artificial Intelligence Engineer	400	40€	1	16.000€
MATLAB_R2022b	-	-	1	2.000€
UPC computer and electricity			2 computers + electricity	5.500€

Price	23.500

ANNEX E: Ethics Committee

As it has been said on section 6.1, this data comes from different public datasets generated from the following sources:

Data related to Covid-19 and Pneumonia: https://github.com/jannisborn/covid19_ultrasound.

Data related to Pneumothorax: [22]

Because this data is public and anonymous and comes from different health centers, they have passed their own Ethics Committee.