Anatomy Highlighting in Lung Ultrasound Images via Deep Learning

Supervisor Oktay Karakus

James Tapp

September 2023

Acknowledgements

I am extremely grateful for the support Oktay Karakus has provided me with as supervisor, and whose encouragement and support made this project possible.

I would like to express my gratitude towards Intelligent Ultrasound for granting access to their data and GPU cluster throughout this project. A special thanks goes to Martin Benson, Thomas Hartey and Emile Belcourt. All of whom played an invaluable role in supporting my research.

I am also thankful for the support offered to me by Wanli Ma in my project development phase, for his ideas within the technical implementation.

I greatly appreciate the support offered by Emily Gaskell, John Tapp and Damaris Tapp. Both Emotionally and in the proofreading and grammatical advice given in support of my project.

Abstract

Intelligent Ultrasound is a company that deals with a large variety of ultrasound related products, aiming to make ultrasound simpler to use and easier to learn, while supporting clinicians in their practice. This is done through the use of cutting edge AI image analysis software. Intelligent Ultrasound have acquired a large dataset of labelled lung ultrasound images, and have allowed us access to the dataset along with their GPU cluster on which to train a deep learning model. This project will investigate the use of U-NET model architectures in performing a pixelwise image segmentation in near real time.

This area of medical ultrasound has experienced extra interest after the Covid-19 pandemic, where clinicians are more likely to encounter patients with lung issues. There are several solutions to similar computer vision tasks, and this project will utilise U-NET. This is one of the most established techniques in medical image segmentation, and we will focus on how this may be applied to lung ultrasound images. This project will attempt to segment the image in near real time on modest hardware in order to most effectively aid a clinician in identifying the critical ultrasound artefacts, and making an accurate diagnosis to maximise learning in the case of training, and patient care in the case of clinical support.

Through use of the provided dataset, several models were trained with the assistance of the Intelligent Ultrasound GPU cluster, in an attempt to produce a good classification model. Additional use of the Open Neural Network Exchange library ensured that any model could produce a classification in sufficient time in order to be potentially applied in a clinical setting on modest hardware.

The models trained showed an intersection over union value of over 0.8 across all classes on a randomly selected validation dataset. In particular, Effusions and Consolidations were classified with very high accuracy of over 0.95 in the best models. B-Lines and Pleura were segmented with accuracy of around 0.85, while Ribs and A-Lines obtained results of between 0.7 and 0.75 on the best performing models. However on a holdout set of selected patients, the model showed signs of overfitting with a worse overall performance. I have also identified areas in which further research is likely to yield to a better model performance, where I was not able to investigate during the course of this project due to time constraints.

Overall this project has succeeded in showing that U-NET can be applied successfully to segmenting lung ultrasound images by training on the Intelligent Ultrasound dataset. The inference time of the model is expected to be low enough to segment images in near real time, allowing it to be deployed in a clinical setting. However, the usefulness of the model to an experienced clinician or to one is training is still unknown, and may require further adjustments. is an area for future research.

Contents

Introduction

Image Segmentation

Image segmentation is one of the most complex areas within the field of computer vision, the processes by which a machine can identify and interpret meaningful information from complex images. Before the advent of deep learning techniques, this was almost an impossible task. The introduction of neural networks however, has revolutionised the field. Semantic image segmentation is the process of dividing a region into areas based on a predefined label, with each individual pixel belonging to a specific class.

Within a medical context this could be different organs, types of tumours, or other relevant features for patient diagnosis. These regions can then be considered for further analysis by a medical practitioner, to confirm the correctness of the segmentation and aid in the overall diagnosis. Overall this expediates the examination time, and reduces the risk of human error. Additionally, segmentation techniques can act as a powerful training tool. By enabling inexperienced clinicians to better understand anatomical structures and gain confidence in interpreting images, trainees experience a quicker uptake of knowledge.

With the introduction of neural networks, segmentation is now easier than ever before. Several different architectures offer promising results for segmenting a wide range of images. First proposed in 2015, U-NET is one of the most long standing and successful models within the field of medical image segmentation, and has been utilised across the field with great results. The structure consists of a contracting path and expanding path, giving it the distinctive 'U' shape. In general, the contracting path captures context in order to classify the image labels, whereas the expanding path give the segmentation the correct localisation of the classes. One of the key features of U-NET is in its skip connections, where different layers of the expanding and contracting paths are connected in order to preserve the spatial information, resulting in a more accurate segmentation. Overall this leaves U-NET as an excellent initial choice for segmenting lung ultrasounds.

Medical Practice

By making predictions and identifying risks, machine learning technologies have made a huge contribution to medical care. The potential to mitigate or even remove human error from decision making could vastly improve patient care. The most advanced medical diagnostic systems often perform on par with or better than experienced professionals, and could bring the same level of care to all patients, even those where medical expertise is not immediately available.

The medical field is currently experiencing a rapid transformation in delivering patient care, with the Covid-19 pandemic initiating a radical adoption of digital technologies. AI is no different, with 83% of healthcare leaders planning to invest in AI in the next three years, and 39% planning to directly invest in AI to support clinical decision making.

We can conclude that the role of machine learning within healthcare is evolving, and there is a significant increase in demand for tools to support medical analysis. In addition, there is a general belief that the use of tools such as the one proposed will lead to better health outcomes within the industry.

Motivation

Currently any lung ultrasounds must be manually interpreted by a trained medical professional. This is based on 'mastery learning', and is heavily dependent on the skill of the operator, as well as the level of training received. This skill level is not easily attainable, and there is no agreement on the most effective training methods. Incorrectly identifying anatomy through a scan can lead to patient misdiagnosis, and poorer health outcomes. A high quality model has the potential to reduce the difference between patient outcomes for more and less skilled clinicians by acting as a co-pilot, ensuring that a consistent standard of lung artefacts are identified by the machine for the clinician to examine.

By improving patient diagnoses, we can optimise treatment plans and improve the health of those with respiratory conditions. Overall this project could improve patient care for respiratory illnesses by improving the ability of clinicians to make an accurate diagnosis. Furthermore, an accurate tool would reduce the strain on the healthcare system by increasing the productivity of clinicians as well as reducing the number of serious illnesses that could be caused by misdiagnosis. Another result would be a reduction in the use of more invasive medical procedures such as MRI if we are able to obtain a good segmentation of the lung via only ultrasound.

Ultrasound Artefacts

Ultrasound images aim to capture a range of features that are useful in aiding patient care. This is done with the use of a small probe, which emits high frequency sound waves. When they encounter parts of the human body, they bounce around creating a unique echo based on the object encountered. This results in a greyscale image being produced, representing the internal bodily structure.

Compared to other types of medical scans, ultrasound offers numerous advantages over other medical imaging techniques. It is non-invasive and does not involve ionising radiation, making it safer for repeated use. Ultrasound imaging is highly versatile, and can capture a range of images in real time, providing an insight into dynamic bodily functions. Additionally, the portability of ultrasound probes allows for bedside or point of care applications. This can be especially useful if the patient is suffering from an infectious disease such as Covid-19, whereby moving them could increase potential exposure. Furthermore, ultrasound is cost-effective compared to alternatives like MRI or CT scans, making it an excellent choice for routine scans and diagnostics.

Lung ultrasounds is emerging as a powerful tool in respiratory medicine. With the onset of Covid-19, patients are more likely to present with lung related illnesses that require quick and inexpensive care in order to perform a diagnosis.

We must also understand what is most important to decision making. Although there are a range of artefacts that are useful in lung ultrasound analysis, there are four critical features: A-lines, B-lines, Pleura and Consolidations.

A-lines, B-lines and Pleura are identifiable as linear objects in lung ultrasound images. Consolidations appear as spotted marks on the ultrasound. Once these are correctly identified a method such as the BLUE protocol can be used to diagnose a range of respiratory conditions. Also identifiable are Effusions, which are caused by the accumulation of fluid in the lungs, giving them a distinctive appearance on the ultrasound scan. Other artefacts identifiable could be ribs, which cast a shadow across portions of the ultrasound image, as well as organs such as the spleen. These are generally of secondary importance to the clinician.

Objectives

The primary goal of this project is to produce a U-NET model that can be deployed in either a clinical or training environment alongside an ultrasound probe to perform a near real time segmentation of a patients lungs. The model will be trained on the Intelligent Ultrasound dataset, and as a secondary goal we will suggest whether more work within the dataset has the potential to produce a better model.

We must therefore achieve a good segmentation, successfully identifying image features at a high level. This is of particular importance for the aforementioned critical artefacts used in medical diagnosis.

The tool must have a high inference speed, so as to produce a segmentation in almost real time. A sufficient time should be less than 0.2 seconds using the Nvidia CUDA execution provider, or less than 0.5 seconds using the more classical CPU execution provider.

Due to a large variety of in image quality, patients, and ultrasound probes the model must be able to accurately segment a large variety of input images. It must be robust enough to handle a range of changes in input data, in order to effectively handle ultrasound images for all kinds of different patient scans. A tool that is suitable only for a specific probe, or a specific demographic of patients is not useful to a clinician within medical practice.

Questions

Within the field of Computer Vision, we can ask whether U-NET can be successfully applied to lung ultrasounds in order to produce a high quality segmentation. To what accuracy can our tool segment ultrasounds overall, and for each individual feature. We will also ask whether it is possible to produce a sufficient segmentation in near real time to be deployed in a clinical setting.

In conclusion, this project presents an opportunity to apply machine learning to healthcare in a novel way. It will provide a useful tool in building human experience in interpreting lung ultrasounds, and become a valuable safety net for experts who might otherwise be subject to human error. By building proficiency and supporting decision making, this project has the potential to significantly improve care for patients with respiratory conditions.

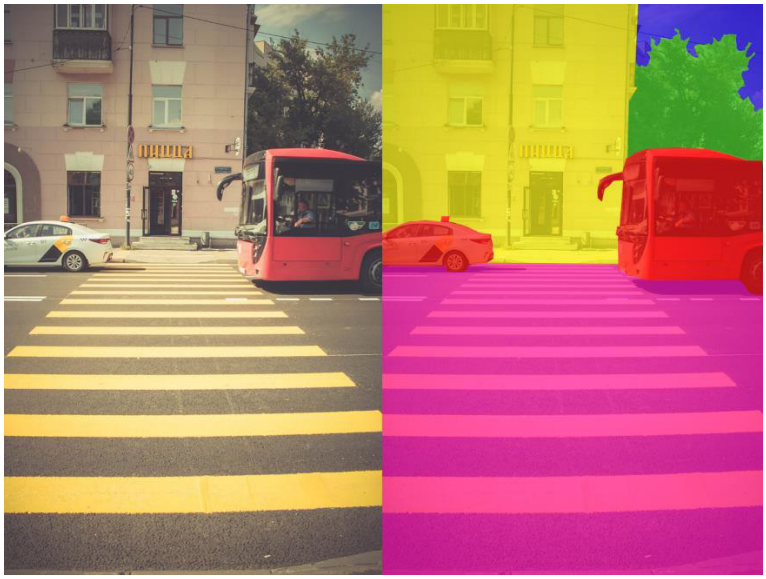

Figure 1: An image alongside its segmentation in class labels. (Palac, B. 2020, licensed under CC-BY-SA-4.0)

Existing Techniques

Algorithmic Approaches

An existing approach to the problem of segmenting lung ultrasound images has been the use of various image transformation algorithms. A-lines, B-lines and Pleura all present as straight lines within the image, and can be identified through a range of image transformation techniques. There are a range of different reasons why this may or may not be more suitable than a data driven approach.

One such advantage is in reducing the need to collect large amounts of personal medical data on which to train a model. Suitable datasets are hard to acquire due to the resources needed to scan patients, and medical expertise needed to label all the images which requires time and resources that could otherwise be assigned towards patient care. Additionally, datasets must be meticulously managed and maintained, and must ensure legal compliance with standards such as GDPR. Without a such a dataset, there is no risk of data breaches or leaks. A good dataset must also be collected in such a manner as to represent all subgroups within the population. Within medical imaging, this means that a diverse selection of age, gender and ethnicity must be considered within the patients who consent to providing us with their data, further complicating data collection.

Algorithmic segmentation benefits from additional transparency when attempting to understand why a prediction has been made. Often we can follow the logic given by the algorithm to a segmentation conclusion, and there is the opportunity to adjust this for a better result. Data driven methods however tend to be a black box, where there are simply too many different parameters interacting for us to fully understand why a prediction is made.

The resource cost for an algorithmic approach is also much lower than for a machine learning counterpart. With no training cost, there is no requirement to access large computational resources such as GPUs. Machine learning models must make sacrifices in terms of complexity in order to reduce the inference time for a classification.

However, when the segmentation class has a complex structure or ill-defined boundaries, it is difficult to develop an algorithm to accurately capture high level details. This is particularly the case for our effusion and consolidation classes within lung ultrasounds.

There is also an extra risk of variation of results, as algorithmic approaches are more vulnerable to changes in data. Different ultrasound probes, noise levels, and patient anatomy must be accounted for in developing such a method.

One significant algorithmic contribution came from Moshavegh et al (2019). In this approach, Moshavegh and colleagues developed a model-based method to identify specific lines, such as B-lines, in Lung Ultrasound (LUS) images. They began by creating a normalized grayscale map of the LUS image to distinguish different structures. The pleural line, a critical reference point, was located using a random walk algorithm. Subsequently, they excluded the upper pleural region and applied a series of filters to separate B-lines effectively. To precisely determine the position of B-lines, they utilized Gaussian models and overlaid the results onto the original images. Notably, this method offered an unsupervised framework for labelling LUS images when annotated data for machine learning was unavailable.

Another notable contribution was made by Anantrasirichai et al. (2017). Anantrasirichai and collaborators introduced an innovative inverse problem formulation for detecting various lines in LUS images. Their approach involved utilizing the Radon transform to convert LUS images into a space of lines. Line detection was treated as an optimization problem, minimized using the Alternating Direction Method of Multipliers (ADMM) algorithm. Initially, the focus was on identifying the pleural line to locate the lung space. Local peaks of the Radon transform were then detected, and line types, including B-, A-, and Z-lines, were successfully identified based on clinical definitions. An extension of this method combined the Radon transform with the Point Spread Function (PSF) of the ultrasound system, allowing simultaneous line detection and deconvolution. Additionally, penalty functions were employed to promote sparsity in the Radon space, enhancing line detection performance. This model-based approach holds promise for precise line artifact detection in LUS images, particularly in clinical contexts, such as the evaluation of Covid-19 patients.

Overall these methods are interesting and relevant to the segmentation of lung ultrasounds, however a data driven approach using the Intelligent Ultrasound dataset offers a range of different benefits.

Neural Networks

The main alternative to algorithmic approaches is to use a neural network model. This is a computational model, inspired by the structure of the human brain. It consists of many layers of interconnected artificial neurons, each with their own adjustable parameters. Through training these neurons we can process information in a manner similar to a human. By choosing the correct network structure and optimising the weights of each individual neuron, we can make predictions and interpretations using a range of complex patterns that humans are not able to comprehend.

The core components of a neural network are the input layer, output layer and hidden layers. The input layer is designed to take in a certain kind of data, which in our case will be a lung ultrasound image, and pass it to the hidden layers of the neural network. The hidden layers conduct a range of calculations via activating different neurons in these layers in order to disseminate the information from the input layer. Finally, once the hidden layers have processed the input it is transformed by the output layer in such a way as to make a valid prediction. In our case this will again be an image, but a classification map or mask for the inputted lung ultrasound.

Training a neural network consists of continually passing data forwards through the model, before then assessing the predicted output against the known ground truth. This is a form of supervised learning, whereby we adjust the models parameters in reference to the known correct prediction for our data. Within the scope of training a machine learning model there are many different parameters to consider, all of which could have an effect on the quality of its output once trained. It is therefore necessary to carry out many different training runs, often varying each parameter one at a time in order to accurately assess the affect that tuning a certain parameter has on the output.

Deep Learning Methods

> Deep Learning is a machine learning technique that constructs artificial neural networks to mimic the structure and function of the human brain. In practice, deep learning, also known as deep structured learning or hierarchical learning, uses a large number hidden layers - typically more than 6 but often much higher - of nonlinear processing to extract features from data and transform the data into different levels of abstraction.
>
> *(DeepAI 2019, https://deepai.org/machine-learning-glossary-and-terms/deep-learning)*

There are a number of neural network models that exist to solve a range of image segmentation problems. The basic method of neural networks trained on images with ground truth masks can effectively solve many segmentation problems, which will be the basis for my approach to segmenting lung ultrasound images.

Deep learning models have achieved great success in an array of medical image segmentation problems, with the best results outperforming expert human analysis. The sheer volume of neurons in such a model allows for the capture of details and connections that could otherwise elude a more conventional understanding, allowing it to outperform algorithmic models for complex imagery. Features can be learnt automatically as part of training, without having to be specified by the user. This allows a model to perform very well without much guidance.

The main drawback of this approach is the dependency on high quality data. The data must be labelled correctly, and represent an accurate section of the population it is drawn from. Otherwise any model trained from this data would be as limited as the data. In a medical context this is especially worrying, as it could prevent all patients from receiving the same level of care if a particular patient is not well represented in the training data.

Another risk is overfitting of the given dataset. A model must learn to distinguish features sufficiently in order to make a good classification. However if the model is trained for too long on too small a dataset, it simply learns what the dataset contains. Thus it performs poorly on any unseen data. This can be mitigated by ensuring the dataset is large and varied, as well as performing data augmentation to further diversify the dataset.

Training any model is computationally expensive, and must be performed over a long period of time on high performance computers. Intelligent Ultrasound have agreed to the use of their GPU cluster in order to support training a model using their lung ultrasound dataset.

One final danger is the lack of interpretability of these models. Due to the sheer complexity, any classification decision can be hard to interpret. In a medical setting this can cause complications by introducing uncertainty into the decision process, where understanding why a classification has been made can be important in diagnosis. Any tool produced should therefore not be seen as a replacement for an experienced clinician, only as an aid to assist with ultrasound interpretation.

In the field of automated lung ultrasound analysis, recent advancements have highlighted the potential of machine learning in order to improve the accuracy and efficiency of medical image interpretation. These data-driven methods have demonstrated their ability to capture complex patterns in lung ultrasound images, and proved their potential to form effective tools within medical analysis. Recent research has focused on fully supervised and weakly supervised learning approaches, where neural networks are trained to identify and classify lung abnormalities, including consolidations and B-lines, with high sensitivity and specificity. Moreover, these methods show promise in the context of Covid-19 diagnosis, where quantitative and automatic LUS scoring systems have been developed to aid in pneumonia evaluation. Despite challenges related to data availability and labelling, machine learning approaches are increasingly contributing to developing a more precise automated lung ultrasound analysis.

A comprehensive and strong segmentation of lung ultrasound images represents a significant step in this field, and a unique contribution to the field of computer vision. Previous approaches have generally focused on specific artefacts, leading to an ultimately limited or binary segmentation and analysis. This is generally of limited use to a clinician, whose analysis must identify many different artefacts in order to be effective. Several approaches have also focused on classification of artefacts,

such as the number of B-lines present and their severity. These have produced good results showing the potential for a full segmentation model to be developed. The use of Intelligent Ultrasound's large dataset could significantly progress these approaches in order to further assist clinicians in patient diagnosis.

CNNs

Convolutional neural networks (CNN's) are a type of deep learning neural network that specialises in image processing and classification. A digital image can be represented as a series of pixels in grid format, with each pixel possessing a value determining its own colour and brightness. In a biological vision system, each neuron works by responding to stimuli in its own restricted field of vision, identifying simple shapes and patterns within a small area. Then by connecting to other neurons in the system in the nearby field of vision, we can begin to recognise more complex structures in our overall sight. CNN's work similarly, with each pixel being connected only to those geographically nearby through the neural network. The neurons can then be trained to activate when certain patterns appear. Through the use of multiple network layers, a computer can pick up first on simple structures, which increase in complexity further along in the network. This is the starting point for allowing computers to see, and is the research field of computer vision.

The architecture of a CNN typically consists of three layers: a convolutional layer, a pooling layer, and a fully connected layer. Additionally, there must be a layer to input data and an output layer to provide a classification.

The convolutional layer is the core building block of a CNN, and carries the main computational load. This is the part of the model that is designed to extract the relevant features from an image. The main method of capturing image is by using a kernel. This is simply a matrix of shape $n \times n$, which slides across the height and width of our image to produce a new feature map. This is done by calculating the dot product of the kernel with the extracted image section.

The formula for the dimension of the new feature map is given below for a kernel size of length n, and an image of length l.

$$Dim = l - n + 1$$

An image may be padded out, with columns or rows added at the edge of the image in order to manipulate its shape. We call this padding, and the number of extra rows or columns added can be represented by an integer p.

We can move the kernel across an image with a varying step size. For example, assume our kernel captures the image indices i to $i + k$ for some constant k. If the subsequent kernel captures the indices $i + 1$ to $i + k + 1$, we can say that the kernel has moved by 1, or that it is of stride 1. In general we can say that if a kernel captures the indices i to $i + k$, then the subsequent kernel will capture the indices $i + s$ to $i + k + s$ for a stride of length s.

If we generalise the given feature map formula to include an image with padding p and stride s, we obtain the following equation.

$$Dim = \frac{l - n + 2p}{s} + 1$$

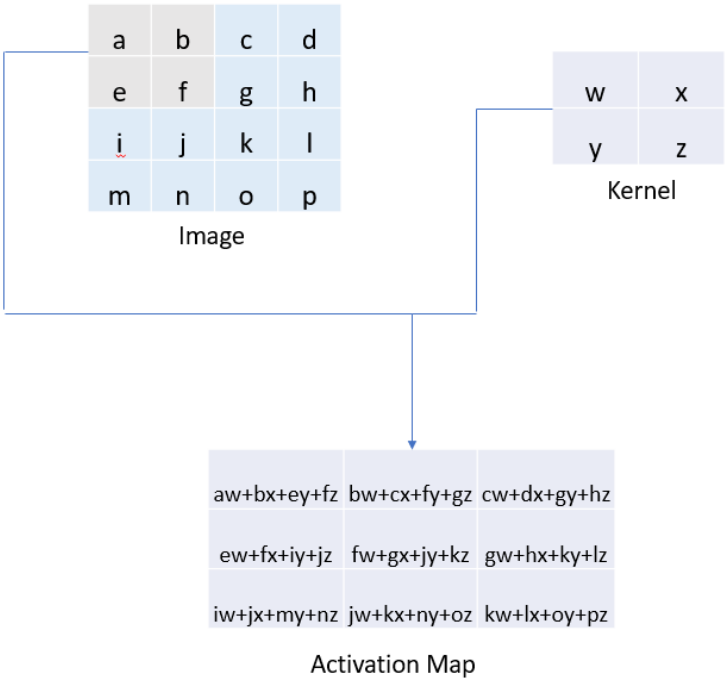

Figure 2: The construction of an image activation map

This structure allows us to reduce the interaction between disparate parts of the image by ensuring that only neighbouring pixels are directly interacting with each other. Trivial neural networks often have every input unit interacting with every output unit, which would be impossible in high resolution images due to the sheer computational complexity required. It also allows for the sharing of parameters, as we can logically conclude if we are trying to identify an image feature, its precise location is irrelevant and we can apply the same analytical methods to each individual image region. We also gain the property of equivariance to translation. This means that if we transform the input image in some way, the output will be similarly changed.

The next step is to process our new feature map through a pooling layer. This is done by applying a function to replace the output at certain locations in the image by deriving a summary statistic of nearby outputs. Similarly to the convolutional layer, this uses a kernel of size $n \times n$. There are several possible pooling functions, however the most popular is max pooling. This returns the maximum value in the kernel, and has the benefit of reducing noise within the image, focusing on the strongest features. Average pooling for example can suffer from this effect. An example max pooling operation is given below.

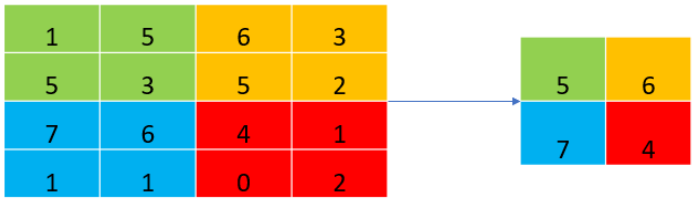

Figure 3: A max pooling operation with 2x2 filters and a stride of 2

These layers have the drawback that they are so far only able to capture linear relationships in the image, therefore a solution is to place a non-linear layer after every convolutional one. The rectified linear unit (ReLU) activation function is one method for this. The mathematical function is relatively simple, allowing it to be efficiently applied to every element in our feature map.

$$f(x) = \max(x, 0)$$

The final layer is a fully connected one where all neurons are connected to the preceding and succeeding layer. This is a traditional neural network layer, and can be used to make final predictions by summarising all the information in the model.

A final convolutional neural network could take the following form:

[INPUT LAYER]

-> [CONV1] -> [BATCH NORM] -> [ReLU ACTIVATION] -> [MAX POOL 1]

-> [CONV2] -> [BATCH NORM] -> [ReLU ACTIVATION] -> [MAX POOL 2]

-> [FULLY CONNECTED LAYER]

-> [OUTPUT LAYER]

U-NET

In 2015 Ronneberger 0., Fischer P. and Brox T. proposed a revolutionary new method for image segmentation. Previous methods had been limited by requiring the collection of large datasets. Furthermore these models were large, while U-NET could produce a segmentation in less than a second on their hardware. With training data often lacking for medical image segmentation, along with a quick classification time and good accuracy, this made it a big step forwards in medical image segmentation. It also produces a pixelwise segmentation of any image, the ideal result for a clinician to be able to quickly identify relevant features. Although initially used as a binary classification method to identify one class within an image, it can be easily generalised in order to produce a segmentation with any number of possible labels.

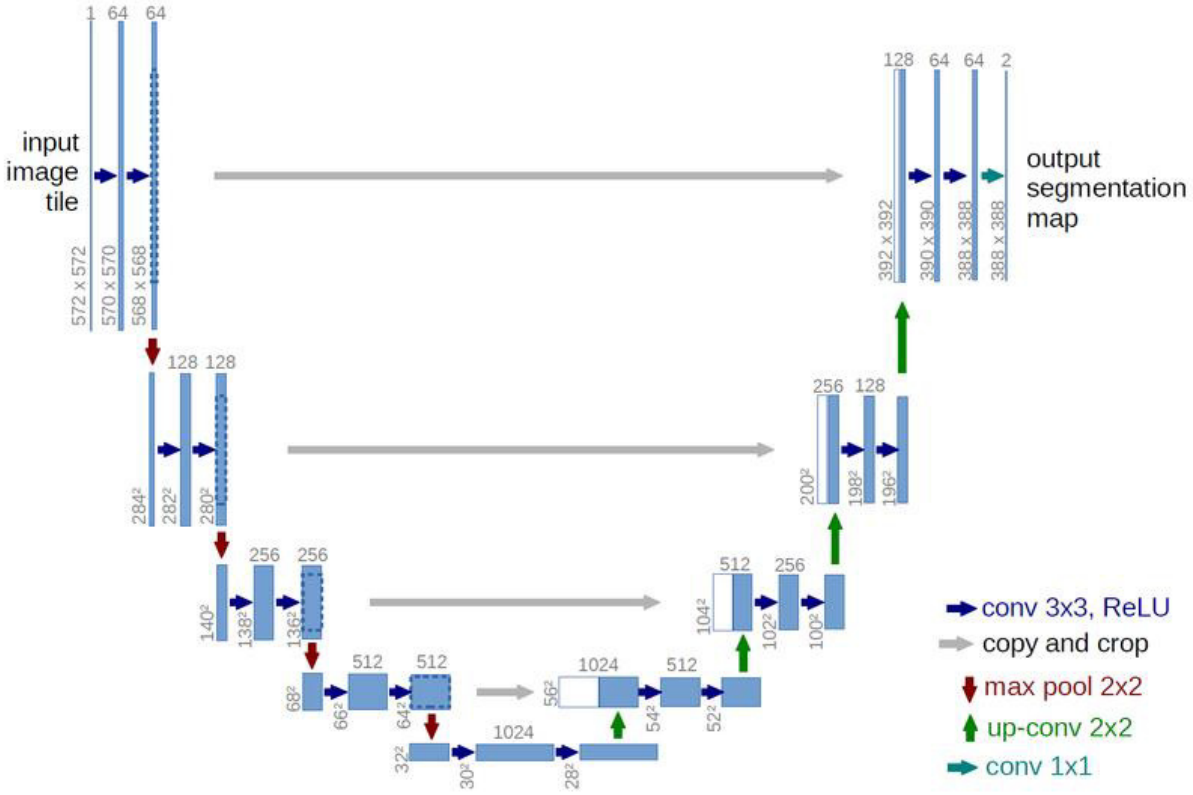

Figure 4: Architecture of a U-NET neural network (TheThinkingMan 2022, licensed under CC-BY-SA-4.0)

The underlying structure of U-NET uses many of the same features as CNNs, including convolutional layers. There is a contracting path which reduces the size of the image into smaller feature maps in order to capture detail in the images. Simultaneously, the number of feature channels are increased in order to capture a wider range of different patterns from the data. At the bottom of the contracting path, there exists a bottleneck layer which routes the input to the expanding path. This increases the feature map size, while decreasing the number of feature channels in the model. The final layer produces a classification for each pixel in the model, which can then be transformed into a full multilabel segmentation. A distinctive feature of U-NET is the presence of skip connections between the contracting and expanding path. At a high level these links help the model preserve the correct location of any features that are being segmented.

The contracting path acts similarly to how a typical CNN architecture would classify an image. It is a repeated application of two 3×3 convolutions, each followed by a ReLU activation. This is then followed by a 2×2 max pooling operation with a stride of 2, halving the size of the feature map. Each time this occurs, we double the number of feature channels in our network.

The expanding path starts with an up-convolution, working similarly to the convolutional layers in reverse. A kernel is repeatedly applied to sections of the image, with the output being a higher dimension matrix. Typically this is a 2×2 convolution in order to double the feature map's spatial dimension. Additionally the number of feature channels is halved at this stage. This is then followed by the same two 3×3 convolutions and ReLU activation functions as in the contracting path. The result of all this is concatenated with the feature map from the corresponding layer of the contracting path, resulting in our skip connections. The contracting feature map must however be cropped in order to account for any changes in dimensions from the various layers.

The final part of the model is a 1×1 convolution in order to map each component feature vector to the desired number of classes. The output will be a pixelwise probability distribution per class from which we can easily construct our predicted segmentation.

Part of training such a U-NET model is making use of data augmentation techniques. In order to ensure the network is robust, and when only a few training samples are available, we can increase the variety of our training data by performing various transformations on our images and masks respectively. Examples could include random cropping, vertical and horizontal reflection, rotations to varying degrees and varying brightness. By performing these operations randomly on our dataset we can ensure our model is taught to be sufficiently invariant.

The initial results of U-NET architectures were extremely good. On two separate cell tracking dataset segmentation challenges, U-NET's IOU was 0.09 and 0.31 higher than the second best models for the respective datasets, representing a significant step forward in computer vision.

U-NET++

As good as the original U-NET proved to be, it was still limited in some ways. In a medical setting, even small segmentation errors can lead to a poor user experience. The question then is how much can we improve upon the U-NET model in order to meet the demands of medical segmentation, and reduce any errors in computer generated or assisted diagnosis. U-NET++ addresses this through the use of improved skip connections and deep supervision. By increasing the effectiveness of the transfer of information from the contracting path to the expanding path, we will achieve a more accurate segmentation. Deep supervision works to ensure that the model learns hierarchical features at multiple scales, by introducing additional classifiers at intermediate layers of the network. These can be used to provide feedback during training, allowing the model to learn more fine-grained details in images. This multi-scale approach helps the model capture intricate structures and nuances, enhancing its ability to produce precise and reliable segmentations.

In U-NET the feature maps of the contracting path are directly received in the expanding path. However in U-NET++ they undergo a series of intermediate steps. The main idea is that information will be better preserved if the concatenated feature maps in the expanding path are semantically similar to each other. We can see below the original U-NET in black, and the addition of dense skip connections in green and blue.

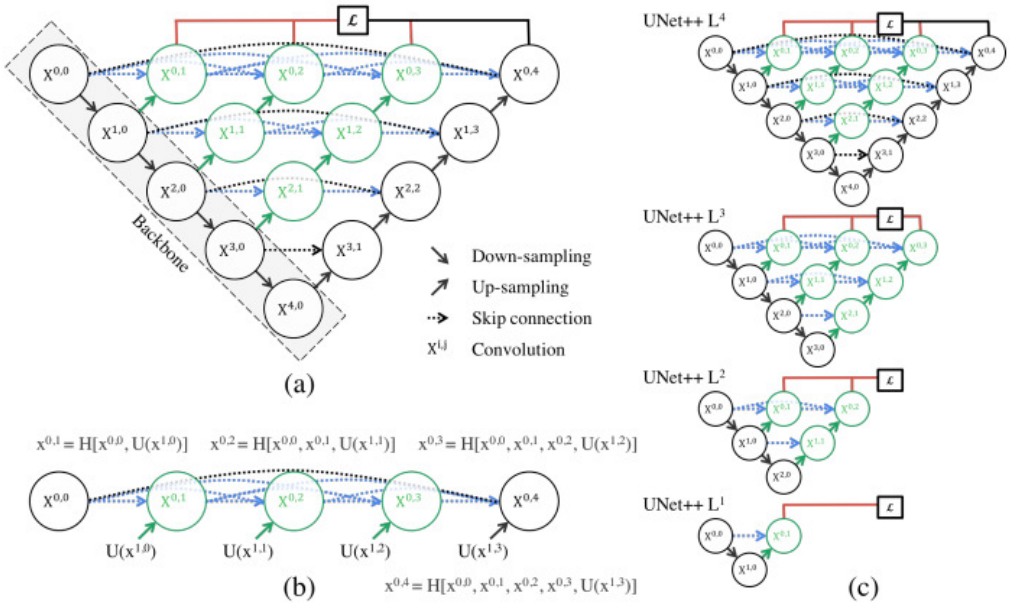

Figure 5: An architectural overview of U-NET++ (Liang, J. et al. 2018)

Each intermediate skip connection block can be expressed as $X^{i,j}$, with i representing the down sampling layer, and j representing the convolution layer of the block along the pathway. Each block is calculated through a convolution layer, followed by an activation function of the concatenation of all blocks at the same down-sampling layer, plus the up sampled block directly below it. For example, block $X^{1,3}$ is produced by a convolution and activation of the concatenated blocks $X^{1,0}, X^{1,1}, X^{1,2}$ and the up sampled $X^{2,2}$.

Formally we can express the skip pathways as follows. Let $\mathrm{H}(\cdot)$ represent a convolution operation followed by an activation function, $U(\cdot)$ represent an up sampling layer, and [] represent a concatenation layer. Therefore, the output $x^{i,j}$ of any node $X^{i,j}$ is computed as follows:

$$x^{i,j} = \begin{cases} H\left(x^{i-1,j}\right), & for\ j = 0 \\ H\left(\left[\left[x^{i,k}\right]_{k=0}^{j-1},\ U\left(x^{i+1,j-1}\right)\right]\right), & for\ j > 0 \end{cases}$$

Figure 5 part b further clarifies this equation by showing the transfer of feature maps across the top skip pathway of the model. This method means that all prior feature maps will accumulate before arriving at the current node, ensuring a higher transfer of information through the model.

Due to the modified architecture within the skip connections, U-NET++ produces a full feature map at multiple different points ($x^{0,j}, j \in \{1,2,3,4\}$). Each of these points is a possible prediction for the model, and each offers a chance for us to use a supervised approach. This is displayed in red on our diagram. A typical loss function such as cross-entropy or focal loss can be used at each of these levels to ensure our model is fully trained throughout.

Overall U-NET++ proved to be a noticeable improvement. Experiments on several medical segmentation datasets showed an averaged IOU gain of 3.9 points over U-NET, showing the importance of improved skip connections and deep supervision to the model.

U-NET 3+

In an attempt to further improve upon U-NET, a third version was proposed by Huang, H. et al. They acknowledged that U-NET was being widely used in medical image segmentation, however it was continuously experiencing the same flaws. U-NET++ had enhanced the sharing of information between different scales of the model, however this was still insufficient in many medical applications. Particularly in settings where organs could appear at varying scales, previous techniques fell short, and produced insufficient segmentations by failing to combine both high level and low level features effectively, limiting the network's ability to segment an image along its correct boundary. Furthermore there was an increased demand for computational efficiency in order to produce a good segmentation in near real time.

The first improvement introduced in U-NET 3+ is the use of full scale skip connections. Similarly to U-NET, the feature map from the same scale is directly received in the expanding path through a convolutional layer. In contrast, a set of interconnected skip connections delivers low level detailed information from the smaller scale contracting path layers by applying a non-overlapping max pooling operation. The higher level contracting path is put through a bilinear up-sampling layer in order to increase its spatial dimension. The feature maps are then merged via a convolutional layer, batch normalisation layer and ReLU activation.

The result is a set of feature maps that incorporate information from different scales, allowing the network to capture fine grained details from smaller scales, and high level semantics across the entire range of scales. It is also worth noting that these skip connections are more efficient with fewer parameters than a U-NET++ structure.

This process can be seen on the figure below, showing the process of constructing the full-scale aggregated feature map of the third layer of the expanding path.

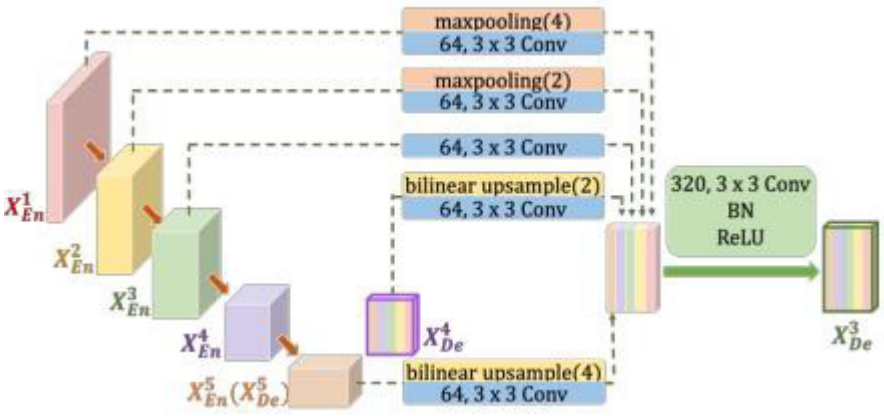

Figure 6: Construction of the full-scale aggregated feature map of third decoder layer X_{De}^3 (Huang, H. et al. 2020)

These skip connections can be formally expressed as follows. Let i index the layer of the contracting path and let N represent the total number of layers in the network. Let $H(\cdot)$ be the feature aggregation mechanism with a convolution followed by a batch normalisation and activation, and $C(\cdot)$ denotes a convolution operation. $U(\cdot)$ and $D(\cdot)$ are up sampling and down sampling functions respectively, and $[\cdot]$ is the concatenation function. The stack of feature maps represented by X_{De}^i is then given by the below equation:

$$X_{De}^{i} = \begin{cases} X_{En}^{i}, i = N \\ , \\ H\left(\left[\underbrace{C\left(D\left(X_{En}^{k}\right)\right)_{k=1}^{i-1}, C\left(X_{En}^{i}\right)}_{Scales: 1^{st}-i^{th}}, \underbrace{C\left(U\left(X_{De}^{k}\right)\right)_{k=i+1}^{N}}_{Scales: (i+1)^{th}-N^{th}}\right]\right), i = 1, \cdots, N-1 \end{cases}$$

The second improvement is the use of full scale deep supervision. The last layer of each decoder stage is fed into a plain convolutional layer, followed by a bilinear up-sampling and sigmoid function. This produces a valid segmentation prediction at each stage, which is supervised by the true segmentation mask.

To further improve the class boundaries, U-NET 3+ uses a hybrid loss function. This is made up of the MS-SSIM value, focal loss and intersection over union loss.

$$l_{seg} = l_{fl} + l_{ms-ssim} + l_{iou}$$

This forces our model to learn a range of structural boundaries by optimising multiple metrics, resulting in a better overall segmentation.

The final improvement is the introduction of a classification guided module. In most medical image problems, false positives are an inevitable consequence of automated segmentation. In order to further clarify the presence of a label class in an image, a new module is created at the bottom of the model, X_{En}^{5}. Benefiting from the richest information in the contracting path, this then outputs a 2 dimensional tensor, denoting with or without the class labels. Subsequently, we multiply the single classification output with the side segmentation outputs in order to reduce erroneous segmentations, and prevent segmentation of absent classes. However the work of the authors has only focused on binary segmentation, and it is currently unclear whether this can be successfully applied to our dataset of multiclass lung ultrasound images.

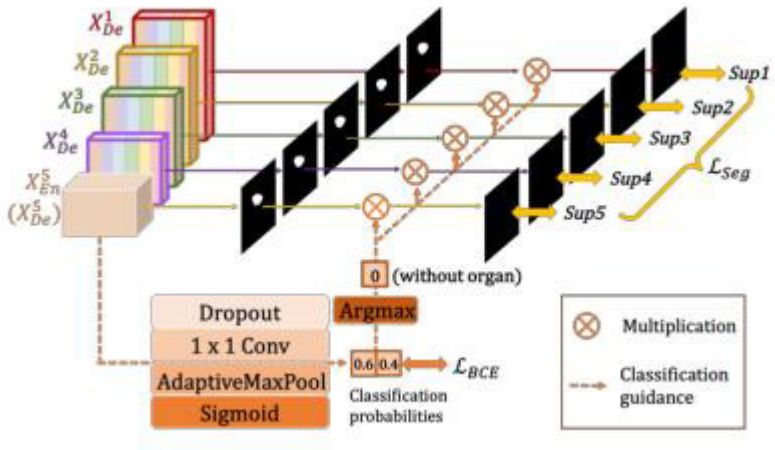

Figure 7: Architecture of a class guided module used in U-NET 3+ (Huang, H. et al. 2020)

The initial model was tested on two datasets of abdominal CT scans, segmenting for the liver and spleen. Overall the results were an improvement, with a Dice score of 0.9552 compared to a score of 0.9352 from U-NET++. The best results were all obtained using the full combination of hybrid loss function, class guided modules, and deep supervision. It is also worth noting that U-NET 3+ used

approximately two thirds of the parameters that U-NET++ used in its segmentation model, significantly improving the models performance time.

Conclusions

In conclusion, this section has provided a comprehensive overview of existing methods of image segmentation and their limitations. While numerous approaches have been developed over the years, it is evident that the U-Net architecture represents a significant advancement in the field of semantic segmentation. With its innovative design, skip connections, and robust feature extraction capabilities, U-Net has demonstrated superior performance in a wide range of medical image segmentation tasks.

However, it is crucial to recognize that no single segmentation method is universally superior, as the choice of approach often depends on the specific requirements and characteristics of the image data at hand. Therefore, it is essential for researchers and practitioners in the field to carefully consider the nuances of their segmentation task and select the most suitable method accordingly.

Moving forward, the integration of deep learning techniques, such as U-Net, with other advanced technologies like transfer learning, attention mechanisms, and multi-modal data fusion, holds great promise for further improving the accuracy and efficiency of image segmentation tasks. This ever-evolving landscape of image segmentation methods underscores the importance of ongoing research and development in the pursuit of more accurate and versatile solutions for various applications in computer vision and medical imaging.

Methods

Software

The choice of different programming languages and software libraries is a key part in developing a machine learning application. For this project I chose to work in python, specifically using the PyTorch library.

Python is a very natural choice, as the simplicity and readability of the language allows the developer to more easily focus on the machine learning approach, rather than coding. Additionally there is already a large and established community of Python developers creating and improving upon existing machine learning frameworks, some of which directly contributed to this project.

Using PyTorch was a more difficult decision, as it is roughly equivalent to other libraries such as TensorFlow in terms of support and standards. The final choice was determined by the expertise of my academic colleagues as well as those of Intelligent Ultrasound, all of whom preferred PyTorch. Any code produced will likely be maintained by another developer going forwards, so developing within a distinct library would dramatically reduce the maintainability of my project in the future.

Throughout the development process, I have focused on providing clear and concise comments in addition to focusing on readable Python code. The machine learning engineers at Intelligent Ultrasound also provided a basic template, intended to best make use of their distributed GPU cluster. This has provided the basis of my development in order to maximise the use of all available resources, while keeping a familiar structure for any other future developers.

Dataset

The first step in deciding how to approach the segmentation task was to conduct a thorough data analysis of the dataset. Intelligent Ultrasound provided some initial information about the class labels and features. These were specified as below, with an RGB colour value:

Class Feature	**Red Value**	**Green Value**	**Blue Value**
Pleura	255	0	255
A-Lines	0	255	255
B-Lines	255	255	0
Confluent B-Lines	255	100	0
Consolidations	0	255	0
Ribs	255	0	0

However, this was quickly shown to be false. The following distinct RGB colour values appear within the dataset, and must be accounted for when loading data.

Red Value	**Green Value**	**Blue Value**
0	255	255
255	0	0
255	1	1
1	0	1

1	1	0
255	100	1
255	1	0
255	0	255
0	1	255
1	255	1
100	50	0
0	0	0
0	255	0
1	0	0
150	0	75
255	100	0
150	0	255
255	101	1
101	51	0
0	150	255
0	1	1
255	0	1
1	255	0
100	51	1
255	1	255
0	0	255

Clearly some of these colours can be successfully merged into their obvious classes, such as the RGB (1, 0, 1) being part of the background class (0, 0, 0). Several colours needed a further investigation to decide which class they are part of, including (0, 150, 255), (0, 0, 255), (150, 0, 75), (150, 0, 255), (100, 50, 0). After sending several examples to medical professionals for confirmation, the following additional labels were decided.

RGB Colour Code	**Anatomical Feature**
(0, 0, 255)	Effusion
(0, 150, 255)	Liver
(100, 50, 0)	Effusion (also)
(150, 0 75)	Consolidation (additional to existing class)
(150, 0,255)	Spleen

Effusions are useful anatomical features, and we should capture them in any model that we build, and Consolidations can be added to the existing class. The liver and spleen however are entirely separate from the respiratory system and it makes the most sense to merge them into the background class.

It should be noted that there are no labelled B-Lines in the dataset. Despite the Intelligent Ultrasound segmentation instructions specifying that B-Lines and Confluent B-Lines should be segmented separately, it appears that this is not the case and the two classes are essentially merged in the dataset. Due to the sheer number of images, separating these classes is likely to be a large undertaking, however could be considered if it is of significant use for a medical practitioner. When loading the images into the model, these values were corrected into the following classes.

Class	Count
Ribs	13466
Pleura	15598
A-Lines	7422
Confluent B-Lines	5570
Consolidations	1944
Effusions	407
Total	**16052**

This dataset is extremely large, a rarity for most medical image datasets, however the class size is relatively imbalanced. Although the most useful features in lung ultrasounds are well represented, with a high presence of Pleura, A-Lines and B-Lines.

Despite this, deeper analysis reveals that the number of distinct patients examined is only 73. From each patient, at least one video has been recorded, with at least one image. In total there are 168 different videos recorded, with 16052 images produced. This could potentially lead to a lack of variety in the training data, due to the risk of recording similar structural patterns in the limited number of lungs being viewed. However the number of images will likely lead us to be able to recognise these structures from different angles. This leads us to conclude that we must be careful with overfitting our model and making assumptions about the true diversity of the dataset.

Finally, by checking the dimensions of the scanned images we find we have a large variety of sizes.

Dimension	Count
1048 x 656	424
1538 x 846	1370
1016 x 708	94
640 x 480	49
960 x 720	8275
736 x 1080	504
720 x 1080	233
1366 x 651	1855
1199 x 840	3127
1900 x 830	96
590 x 651	25

As there is no consensus between image dimensions or ratios, we must decide how to resize the images for processing. Image size is one of the main contributors to the time taken to produce a segmentation, however it is important to be able to decide on a shape ratio. Due to the vast array of dimensions, I decided to resize all images to a square shape before feeding them into any model for processing. This would ensure that a fair amount of information is captured from each image, however it is possible that the various ratios will interfere and lead to less information being captured from those images with higher height than width. It is also unclear without a very thorough examination of the data whether the images with a higher pixel count are a higher resolution, or whether they simply contain extra dead space with no class maps. A more detailed analysis and subsequent preprocessing may be required in order to capture a fair amount of information from each image for training.

Models

The next step in developing my tool was choosing an appropriate model architecture. Any model that I used had to produce an appropriate pixelwise classification of an image. Ideally the classification could be produced on a modern GPU in less than 0.1 seconds, in order to segment an input in near real time. A high accuracy segmentation was also needed due to the intended use of the tool in a clinical setting, where a suboptimal tool could lead to poorer patient outcomes.

I chose to explore the dataset with the use of a U-NET model, due to its impressive historical application to medical image segmentation problems. Other architectures were more often applied to different types of segmentation problems, whereas upon reading the existing literature I was extremely confident that I could produce a good segmentation model with U-NET. As discussed in detail in the previous chapter, U-NET and its variants are often state of the art models that perform at the highest level in segmentation tasks.

For this dataset in particular, I felt U-NET was a good choice. With 73 distinct patients in the dataset, there was little to no risk of a lack of diverse data affecting the training, as U-NET models are often trained on comparatively much smaller datasets than the 16052 present. By choosing an appropriate means of data augmentation, I could be confident that my model would correctly segment any unseen data.

Within the literature, U-NET is most often applied to binary segmentation problems or multiclass datasets with a small number of classes. However a clinician must analyse many different pieces of information in order to make a diagnosis, and a multiclass segmentation is a necessity for medical practice. Although U-NET can be easily generalised to produce a multiclass segmentation of any size, this is uncommon and could end up producing unexpected results from a model.

In general, due to the necessity of sharing spatial information at varying levels of the image to classify a lung ultrasound, U-NET is a good choice due to the built in skip connections. In lung ultrasounds the horizontal A-Lines appear as echoes of the pleural line. These are structurally similar, however the Pleural Line is distinguished by its placement at the highest part of the image. This requires a structure that can segment small features of an image to capture the horizontal artefacts, as well as a higher level path that can classify these artefacts into the Pleural Line or A-Line.

Overall, U-NET tends to underperform when attempting to segment complex structures, and struggles to effectively capture label boundaries. Lung ultrasounds produce mostly basic linear artefacts, which means that a U-NET model is unlikely to suffer from being able to segment complicated object boundaries effectively. Additionally within a medical context, a fine tuning of class boundaries is largely redundant. As a human medical practitioner is expected to review the results, this allows them to review the segmentation alongside the image in order to produce their own conclusions in such a way that small errors along the boundary would not matter.

Class imbalances can also be a problem within U-NET models, which is a problem we are extremely likely to encounter within this dataset. Ultimately, this can be minimised by the choice of an appropriate loss function but not reduced entirely. For this reason we will experiment with the use of focal loss functions in order to train our model.

Overall U-NET is a very natural choice for the first model produced to segment the lung ultrasound data, with several different versions available in order to maximise the segmentation results. My preferred model for this project is U-NET 3+, which achieved the best results on the dataset.

However there are likely to be even further improvements to the U-NET architecture in future, and a future exploration of the data could introduce different architectures in order to seek improvements.

Data Augmentation

Due to the potential lack of diversity in the dataset due to the limited number of distinct patients, it was necessary to consider methods to extend the variety and scope of the data available. Augmentation of image datasets is common practice for image classification and segmentation problems, and many methods exist. These can generally be broken down into spatial transformations, colour transformations, and noise transformations.

The spatial techniques I used were a random rotation of between 30 degrees either clockwise or counter clockwise, alongside a random reflection in the image's y axis. It was important to preserve the vertical structure of the image, due to the necessary sharing of information in the model to classify certain features. A-lines appear as horizontal lines, and B-lines appear as vertical lines, so a rotation of 90 degrees or more could significantly affect the model's ability to classify these structures. Furthermore A-lines often appear as echoes of the Pleura, presenting below the Pleural line. This means that a reflection in the image's x axis was not possible in order to effectively share the ability to classify these structures. Although I considered investigating a random cropping, I decided that this was too likely to have a negative effect, as the segmentation relies on whole scale details of the image.

The colour transformations I used were contrast and brightness variations. While brightness refers to the overall lightness of the image, contrast is the difference between light and dark parts of an image. When increasing brightness every image pixel as a whole gets lighter or darker, whereas upon increasing contrast the light parts of an image get lighter, and the dark parts get darker. Due to the greyscale nature of ultrasound images, available pixel colour augmentations are limited. It is unlikely for there to be many useful additional augmentations in this area.

When an image is taken, there are often imperfections in the capture resulting in distortions and defects in the picture. These can be caused by anything from sensor inaccuracies to random errors, and are to be expected in real world captured images. We can simulate this in our dataset in various ways, by randomly changing pixels to simulate this external interference. Gaussian noise is a noise added to the original image which has a normal (or Gaussian) probability distribution. For an image pixel with coordinates x and y, the pixel can be expressed as the sum of the original pixel $s(x, y)$ and the noise $n(x, y)$.

$$w(x, y) = s(x, y) + n(x, y)$$

The probability distribution of the added noise $p_G(z)$ is then given by the following formula.

$$p_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

This makes our model more robust to interference, and should enable us to produce a more accurate segmentation on real world data.

Inference Time

Producing a classification in a suitable time was not a worry while using a U-NET model due to its simple structure. The production of a model with a sufficiently low segmentation time was done with the assistance of the ONNX runtime python library, allowing us to produce a runtime for any model

that was independent from the machine it was run on. This also allowed for testing the model runtime without the use of a GPU. It was agreed with Intelligent Ultrasound that aiming for an average runtime of less than 0.5 seconds with the ONNX runtime CPU provider would suffice, as this would speed up appropriately when applied to a modern GPU device.

Evaluation

Although there are many different metrics for assessing the effectiveness of an image segmentation model, for this project I will use intersection over union (IOU). This is one of the most widely used evaluation methods, and makes it easy to quickly assess the usefulness of a potential prediction and to compare between similar models. Without a strong evaluation metric we cannot be certain how our model will perform on any unseen data, and whether improvements are being made to our model throughout the training process.

The number of pixels correctly classified (or accuracy) is generally a poor performance metric within image segmentation. This is due to the fact that it cannot capture any spatial information, where certain pixels may affect the usefulness of an image segmentation more than others. A large class imbalance could also skew the model. Consider for example a binary classification where 95% of the image is the background. A prediction of purely background pixels would produce a high accuracy score of 0.95, with absolutely zero segmentation. This will likely be an issue within our lung ultrasound images, and must therefore be avoided.

Intersection over union is calculated as the sum of the area of the correct predictions divided by the total area of the prediction and ground truth. Referencing figure 8, the true segmentation is displayed by the green circle, and the prediction by the red circle. The intersection over union is therefore calculated by the area in yellow divided by the area in green plus the area in red.

$$IOU = \frac{True\ Positive}{False\ Negative + False\ Positive + True\ Positive}$$

We can clearly see that $0 \leq IOU \leq 1$, and the better the segmentation the larger the value. Within this project we will measure the binary IOU for each individual class in order to analyse and compare the segmentation accuracy for each label.

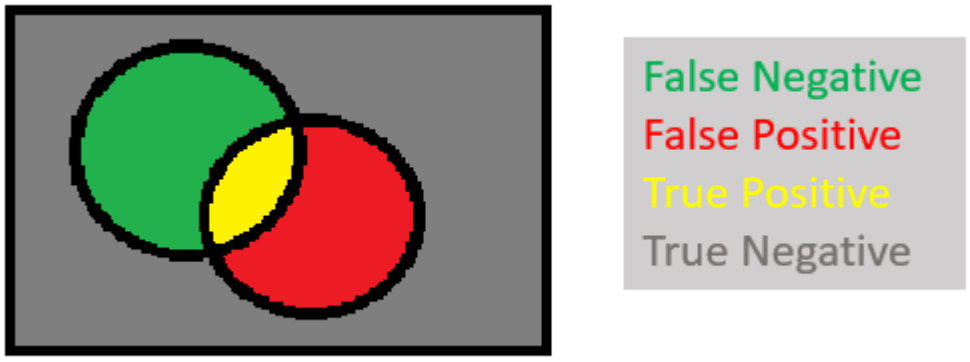

Figure 8: A binary image segmentation to represent intersection over union

Model Optimisation

Once our model architecture is decided, its parameters must be optimized to maximise the value of its output. Through effective training we can adjust the value of each neuron in the network to produce a continuously better model until we are happy with the overall output. The main method for this process is backpropagation.

The first step is to load a batch of data ready for processing. This can be of any size, although may be constrained by memory limitations. The most commonly used value is 2^n for the highest n that machine memory can take. This is then passed through the entirety of the model in order to obtain a prediction. Each neuron therefore performs a weighted sum and activation function in order to produce the final output.

This output is then compared to the true prediction via a loss function in order to produce an error value. There are a large number of choices for different losses depending on what is being optimised and the dataset available. Once an error is calculated, this is passed backwards through the network, producing a gradient for each parameter. Starting at the output layer, the model weights are adjusted through the use of this gradient in order to minimise the final loss function error so that it is as low as possible. Once this is completed for a batch of data, it is then repeated until the entire dataset has been passed through the model. Finally, the entire dataset is passed through the model again in order to optimise it further. Each pass of the dataset is referred to as an epoch. Finally once the model is sufficiently optimised, the training is stopped and the model can be used for predictions.

Loss Functions

The choice of an appropriate loss function is critical to fully optimising a model, and different functions exist for all manner of different problems. It ultimately guides the model during training and has the potential to emphasise certain classes, characteristics and anomalies that the model may come across. Additionally, it can influence the sensitivity of a model to outliers, the handling of noisy data and the speed of convergence when training.

Within image segmentation, an intersection over union loss attempts to maximise the overlap between the predicted class label and the true class label for each class. It measures the similarity by computing the ratio of the overlap between the predicted class label and the ground truth plus the predicted class labels. For any given class, the IOU is given as described in the evaluation section. The IOU loss is therefore given as follows, for all classes c in the set of classes C. Let $|C|$ be the number of classes.

$$IOU\ loss = \left(\sum_c^C \frac{True\ Positive(c)}{False\ Negative(c) + False\ Positive(c) + True\ Positive(c)}\right) \Big/ |C|$$

Cross entropy loss is another function to minimise the prediction error of a machine learning model. It penalises incorrect predictions with high confidence more heavily, encouraging them to assign higher probabilities to the correct classes. For example, assuming the class is present, a prediction of 0.05 would have a much higher loss than 0.7. Assume we have C classes, a binary prediction of value {0,1} $y_{o,c}$ as well as a probability $p_{o,c}$ in the range (0,1) for class c and observation o. Then the formula for a multilabel cross entropy loss function can be expressed as follows.

$$CE\ loss = -\sum_{c=0}^{C} y_{o,c} \ln(p_{o,c})$$

Focal loss is designed to address class imbalances when training machine learning models. It works by reducing the contribution of easy to classify examples in a similar manner to cross entropy loss. However it has the additional option of adjusting how much the model focuses on hard to classify examples (γ), and a class specific weighting factor to determine how much the model focuses on each class (α). Let p_t be the predicted probability of the correct class. Then we define the focal loss for each p_t as follows.

$$FL = -\alpha_t(1-p_t)^{\gamma}\ln(p_t)$$

Typically γ is set between 2 and 5 depending on how fast the model should converge. The list of α values is generally set to be between 0 and 1, depending on the prevalence and the classification difficulty for each respective class. Multiple runs are often needed to fine tune these parameters for an optimal loss function.

Multi scale structural similarity index (MS-SSIM) is another loss function useful for processing images. The value of this metric, is that it measures the similarity between two images at multiple different scales. Whereas other loss functions assign a similar weighting to the boundary pixels between class labels, MS-SSIM weights it higher. This means that by measuring the loss between the true mask and the predicted mask of an image at smaller scales, we can encourage our model to more accurately segment the boundaries of our classes. Depending on what we are trying to segment, correctly identifying the class edges could be of high importance.

Let us take $x = \{x_i | i = 1,2,\dots,N\}$ and $y = \{y_i | i = 1,2,\dots,N\}$ as the pixel values in two image distributions. We then further define μ_x and μ_y as the respective means of x and y respectively, along with σ_x and σ_y as their variance. Then the single scale structural similarity index is given by the following formula.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

The final MS-SSIM is derived from a weighted average of this formula, for continuously cropped sections of the segmentation and ground truth mask.

Optimisation Function

When it comes to algorithms for updating the weights of a machine learning model, there are several choices. For this project I have chosen to employ the Adam optimisation function, which is short for adaptive moment estimation. This is a common choice in many machine learning problems, including image segmentation. The main benefit is that Adam adapts learning rates individually for each parameter weight, leading to a faster convergence than other algorithms.

Learning Rate

A learning rate is one of the most important parameters within machine learning, and determines the step size at which the model updates its parameters during training. The choice of an appropriate learning rate directly affects the convergence and stability of a model. If the rate is too high it can cause the model to overshoot an optimal solution, while if it is too low the convergence may not happen quickly enough leading to a subpar solution. Therefore, finding the right learning rate is essential to ensure that a machine learning model converges efficiently and effectively learns from the training data, ultimately leading to better performance on unseen data.

To determine the most effective learning rate, I will be using the OneCycleLR PyTorch function. This takes in several parameters, and produces a dynamic learning rate to use during training in a single cycle, initially increasing followed by a consistent reduction. This allows for faster convergence to an optimal solution, meaning less training time and a lower use of computational resources.

Logging Runs

For this project I will be making use of the wandb (Weights & Biases) platform, which allows for easy logging of results and creates graphs for each metric from each run. The main metrics to be measured will be the total loss for each subsequent forwards and backwards pass, alongside intersection over union for each full pass for both the training and validation set. This will allow for a quick and easy comparison between runs, and a quick identification of any issues in the training of my models.

Results

During the course of this project I attempted many different training runs in order to fully optimise my model. This included changing many different variables which I will discuss in this chapter. While changing one parameter, it was critical to keep all other variables the same so as not to introduce other influences. The main metrics to measure were the intersection over union values for each class, and the overall average value in accordance with our evaluation metrics.

Inference Time

The time taken to produce a segmentation of a given ultrasound image must be close enough to real time to run alongside an ultrasound scanner on modest hardware. Although deploying this application was beyond the scope of the project, a projected target was a runtime of under 0.5 seconds using the open neural network exchange library and the CPU execution provider. The following runs were all measured to the nearest millisecond for the average timings, on models with 7 output channels to mirror the number of classes

Using the model code for the U-NET 3+ introductory paper, we obtain the following results with a RGB channel input image of size 256 by 256 pixels. Each run was done 1000 times to ensure an accurate measurement.

Model	**Mean Milliseconds Per Segmentation**	**Standard Deviation**
U-NET	838	277
U-NET 3+	1917	562
U-NET 3+ Deep Supervision	2056	159
U-NET 3+ Deep Supervision Class Guided Module	1944	201

Even with accounting for the unpredictable nature of running a program on a multipurpose device, these versions proved to be consistently too slow to be useful. Therefore it was necessary to make modifications to the model class.

Each model contains a preset variable to determine the number of filters for each layer in the U-NET model. These are essentially channels in the network, and extract features from the ultrasound scan by sliding over each part of the image. Reducing these will significantly reduce the number of calculations involved in the model and will therefore lower the run time. It will also reduce the amount of memory needed, which could be important depending on the specifications of the device it is run on.

Reducing the number of filters was a very natural choice to improve model performance. The features we are extracting are in general geometrically simple, and therefore fewer filters would be needed to capture enough information for a good segmentation. However, it is important to note that such a change may decrease the accuracy of our model. Once the filters were reduced by half at each level of the U-NET model, we obtained the following timing results.

Model with reduced filter sizes	Mean Milliseconds Per Segmentation	Standard Deviation
U-NET	141	12
U-NET 3+	441	47
U-NET 3+ Deep Supervision	474	28
U-NET 3+ Deep Supervision Class Guided Module	513	32

These results were generally satisfactory, showing an average performance under the goal of 0.5 seconds or 500 milliseconds. However, these results were generated using an input for RGB images. This means that each pixel is represented by a tuple of three distinct values in the range 0-255. Ultrasound images being entirely greyscale, this seemed unnecessary. The following results were generated with a single input channel, versus a model with three input channels.

Model with reduced filter sizes and grayscale input	Mean Milliseconds Per Segmentation	Standard Deviation
U-NET	120	15
U-NET 3+	408	37
U-NET 3+ Deep Supervision	419	38
U-NET 3+ Deep Supervision Class Guided Module	478	24

I believe this is the ideal format of a U-NET 3+ model for attempting near real time segmentation of lung ultrasound images, with the best results consistently under 0.5 seconds on a CPU execution provider.

U-NET Versions

One of the largest components of each training run is the model architecture. In order to fully explore different structures, I tested several different models trained on different U-NET versions in order to assess the segmentation accuracy. All runs were done using the halved filter architecture and the same loss function.

Model	Class Intersection Over Union						
	A-Lines	B-Lines	Consolidations	Effusions	Pleura	Ribs	Average
U-NET	0.538	0.669	0.881	0.976	0.03	0.156	0.542
U-NET 3+	0.537	0.654	0.881	0.976	0.699	0.676	0.737
U-NET 3+ Deep Sup	0.808	0.828	0.961	0.976	0.733	0.705	0.835

There is a very clear difference between the different U-NET version, with each model improvement carrying an overall improvement in IOU. Although U-NET 3+ has a slight dip in A-Lines and B-Lines IOU, the increased segmentation of both the Pleura and Ribs more than make up for this. This could be explained by the necessity of knowing the other image features in order to correctly classify a pixel as Pleura and Ribs, and the improved skip connections in U-NET 3+ could be providing this.

The deeply supervised approach shows improvements across all classes, suggesting that it has worked well and is a useful addition to the model. This seems to have the greatest effect on the classes that need the least amount of whole image information to make a classification. Considering that the deeply supervised approach works by updating model weights further down the model path where more fine grained features are present, this is working as expected.

Loss Functions

The next logical test is to ask which loss function will produce the best results. Choosing the correct loss function determines how the model evaluates its own performance, and which features it learns best. Huang, H. et al proposed that U-NET 3+ should be trained on a hybrid loss function, consisting of intersection over union loss, focal loss and MS-SSIM loss. I felt it necessary to explore this during my project.

Loss	**Class Intersection Over Union**						
	A-Lines	**B-Lines**	**Consolidations**	**Effusions**	**Pleura**	**Ribs**	**Average**
Intersection Over Union (IOU)	0.538	0.788	0.859	0.975	0.03	0.162	0.559
Cross Entropy (CE)	0.538	0.794	0.881	0.976	0.03	0.156	0.505
Multi Scale Structural Similarity (MS-SIM)	0.796	0.824	0.957	0.976	0.734	0.703	0.831
Jaccard	0.788	0.784	0.961	0.987	0.682	0.656	0.795
Focal $\alpha = (0.1, 0.5 \dots)$ $\gamma = 2$	0.769	0.784	0.965	0.978	0.659	0.657	0.788
Focal $\alpha = (0.1, 0.5 \dots)$ $\gamma = 3$	0.786	0.781	0.965	0.977	0.662	0.651	0.789
Focal $\alpha =$(0.1, 0.5 …) $\gamma = 4$	0.792	0.781	0.963	0.984	0.657	0.652	0.804
Focal $\alpha =$(0.1, 0.8, 0.8, 0.7, 0.7, 0.5, 0.5,) $\gamma =$2	0.796	0.768	0.964	0.974	0.622	0.619	0.790
Focal $\alpha =$(0.1, 0.3, 0.3, 0.5, 0.5, 0.7, 0.7) $\gamma =$2	0.786	0.782	0.964	0.984	0.681	0.657	0.809
IOU, CE and MS-SSIM	0.761	0.812	0.881	0.976	0.711	0.706	0.793
IOU, Focal, and MS-SSIM	0.805	0.831	0.957	0.976	0.734	0.709	0.835

Jaccard, Focal and MS-SSIM	0.809	0.825	0.967	0.985	0.728	0.705	0.837

From looking at the results, clearly IOU and Cross Entropy loss are subpar functions, and completely fail to learn the Ribs and Pleura classes. Although Jaccard loss is theoretically calculated by the same method as IOU, we can see that the implementation by Silva, J. is much improved. This produces a sufficient, but not optimised result by itself. Multi scale structural similarity loss however produces an extremely good result, strongly justifying its place as a component of a hybrid loss function.

More interesting results come from the Focal loss runs. We can see that the variation of the gamma variable has very little effect on the overall IOU values. This suggests that the majority of images can be classified easily, and our model is already performing generally well. However we can see some minute overall improvement for a higher gamma value of 4.

Changing the alpha variable also leads to a minimal change in evaluation. From our results so far, we can generally say that the models are struggling to segment the Ribs and Pleura classes the most, while the Consolidations and Effusions are the easiest, with A-Lines and B-Lines somewhere in between. However, there are the most examples by far of the Ribs and Pleura classes, followed by A-Lines and B-Lines, and the fewest examples of Consolidations and Effusions. The difficulty of classifying the more common classes seems to be roughly balanced out by their prevalence in the dataset, at least within the focal loss function. The best results were with an alpha value of (0.1, 0.3, 0.3, 0.5, 0.5, 0.7, 0.7), which aims to underemphasise the more common examples in the dataset. However, this was only a marginal improvement of 0.02.

Within the tested hybrid loss functions, the lowest scoring was the combination of cross entropy, intersection over union and multi scale structural similarity. This is unsurprising as focal loss is in almost all cases an improved version of cross entropy loss. The proposed hybrid function for U-NET 3+ of focal, intersection over union and MS-SSIM works better. However the best results are achieved with Jaccard, focal and MS-SSIM. Although this is only a 0.002 increase, it suggests that there is further optimisation possible by improving upon the IOU loss function. There are several alternatives within the literature, and this is an interesting point of investigation for future research.

Colour and Grayscale

With an improved processing time for grayscale images over colour images, it is necessary to confirm whether there is a difference between grayscale and colour segmentation results.

Colour	**Class Intersection Over Union**						
	A-Lines	**B-Lines**	**Consolidations**	**Effusions**	**Pleura**	**Ribs**	**Average**
Grayscale	0.791	0.812	0.958	0.971	0.730	0.714	0.829
RGB Colour	0.805	0.831	0.957	0.976	0.734	0.709	0.835

The RGB colour scale has a very small average improvement of 0.006 IOU. Individually each class is segmented to a similar quality, and can be concluded that there is little to no change in overall segmentation results.

Data Augmentations

I experimented with several different data augmentations, with the goal of improving the robustness of the model once trained.

Reflection was done only along the vertical y axis, in order to preserve the relationship between the Pleural Line and the A-Lines, and rotation was applied to a maximum of 35 degrees in each direction, in order to further preserve the vertical relationship between lung artefacts.

Contrast and brightness is applied with a probability of 1 to all pixels. Both have factor limits of 0.2 in order to preserve the image information.

The Gaussian noise added has a mean of 0, and a variance limit of 20 for a pixel brightness value. This is applied with probability 1 to all pixels.

Augmentations	**Class Intersection Over Union**						
	A-Lines	**B-Lines**	**Consolidations**	**Effusions**	**Pleura**	**Ribs**	**Average**
None	0.891	0.865	0.983	0.976	0.860	0.867	0.907
Rotation and Vertical Reflection	0.795	0.823	0.957	0.976	0.731	0.707	0.837
Rotation, Vertical Reflection, Brightness and Contrast	0.732	0.734	0.880	0.976	0.580	0.564	0.762
Rotation, Vertical Reflection, Brightness, Contrast and Gaussian Noise	0.640	0.659	0.881	0.976	0.327	0.323	0.634

Overall, we can see that the model performs significantly worse depending on the augmentations applied. While Consolidations are still easily identified after all data augmentations, the average suffers greatly. Rotation and reflection are failing to improve the model's performance, although are mainly causing harm to the identification of the Ribs, Pleura and A-Lines. As these are all generally vertical artefacts, it would be logical to suggest that this is caused by over rotation rather than reflection.

Additionally brightness and contrast affects all classes other than effusions, reducing the IOU value by around 0.1 each. This seems to show that the images lose too much information by reducing the brightness and contrast, and therefore the model struggles to learn a good segmentation.

Gaussian noise again has no effect on the effusion class, and even slightly improves the consolidation segmentation. However, this is easily counteracted by the significant IOU reduction in all the remaining segmentation labels. The improvement of the Consolidations class is a promising sign however, being the only direct improvement in segmentation results from my applied augmentations.

Image Size

Due to the possibility of increasing or reducing the image size during preprocessing, it is important to examine the direct impact of changing the size of the image fed into the network in order to determine whether any potential improvements can be made. It is also possible during testing that we may see either a higher or lower inference time than expected. If higher, then the simplest fix may be to simply reduce this image size, and if lower we may be able to increase the image dimensions in order to improve the segmentation.

Number of Image Pixels (Height and Width)	Class Intersection Over Union						
	A-Lines	B-Lines	Consolidations	Effusions	Pleura	Ribs	Average
128 by 128	0.633	0.717	0.899	0.909	0.518	0.443	0.635
256 by 256	0.795	0.823	0.957	0.976	0.731	0.707	0.837
384 by 384	0.647	0.768	0.905	0.976	0.509	0.383	0.698
512 by 512	0.517	0.515	0.877	0.976	0.449	0.156	0.582

These results suggest that the previously proposed 256 by 256 dimension image is close to the optimal. The lower model trained on a 128 square pixel image does not appear to capture enough details from the source image, producing a worse segmentation. Conversely the increase in image size does not appear to increase the segmentation significantly.

Class Guided Module

I also experimented with an implementation of the class guided module model for multi label segmentation, and compared it against the same model with no such module.

Model Type	Class Intersection Over Union						
	A-Lines	B-Lines	Consolidations	Effusions	Pleura	Ribs	Average
Deep Supervision	0.798	0.822	0.947	0.976	0.735	0.705	0.831
Deep Supervision and Class-Guided Module	0.509	0.710	0.894	0.980	0.716	0.557	0.728

These results were very interesting. On average, the segmentation results dropped by 0.103 IOU. However, the Effusions class received a boost, with the Pleura and Consolidations class remaining roughly equivalent. However A-Lines and Ribs saw very significant decreases.

Train Test Split

During this work I used a randomised split function on my dataset to separate them into a training and validation set. Therefore the validation data would likely have some instances of very similar images within the training data due to the same patients and videos being present in each dataset.

To test for the model overfitting, I selected several patients to be removed entirely from the training dataset. These patients were selected to provide coverage of all different lung ultrasound artefacts, in order to provide a measured estimate of the models ability to segment each class on a completely unseen patient.

The patients chosen were segPatient00000, segPatient00004, segPatient00006, segPatient00008 and segPatient00051.

Dataset	**Class Intersection Over Union**						
	A-Lines	**B-Lines**	**Consolidations**	**Effusions**	**Pleura**	**Ribs**	**Average**
Training	0.839	0.821	0.955	0.973	0.733	0.715	0.839
Randomised Validation	0.830	0.815	0.954	0.971	0.728	0.711	0.835
Selected Holdout	0.392	0.847	0.978	0.986	0.475	0.320	0.666

The training set and randomised validation set have a very similar accuracy, different only by 0.004 IOU. The differences in the selected holdout set are drastically different, with the Consolidation, B-Lines and Effusions classes performing similarly well to the randomised validation and training sets. The A-Lines, Pleura and Ribs however drastically underperform, with an average difference between the three classes of 0.361 IOU compared to the randomised validation set.

Conclusion

Final Evaluation

Overall the models I have trained over the course of this project have performed well, although show signs for possible improvements. The best models were trained with the U-NET 3+ Deeply Supervised model, without the use of the class-guided module. The model took in a grayscale input, and resized the input images to 256 by 256 pixels before processing. This made use of a loss function comprising of focal loss, Jaccard loss and multi scale structural similarity loss, all with equal weighting. The focal loss parameters had a gamma value of 2, and an alpha value of 0.1 for the background class, with 0.3 for Ribs and Pleura, 0.5 for A-Lines and B-Lines, and 0.7 for Effusions and Consolidations. The best results were obtained with no augmentations, and were trained using a batch size of 4 images through 25 epochs. Due to time and resource constraints on the Intelligent Ultrasound GPU cluster, such a model could not be trained for this project. However, from the results gathered such a model could be expected to have an average IOU segmentation of close to 0.9 over all classes on a randomised validation dataset. The evaluations for the Consolidation and Effusion classes are consistently over 0.95, while the results for the B-Lines and Pleura are consistently over 0.8 on such a model. While the Ribs and A-Lines classes are more difficult for the model to segment, they still show over 0.7 and 0.75 consistently on the best trained models.

One large question is how such a model will extend to a completely unseen patient. The tests run show that we can be confident in the segmentation of the Consolidation, Effusion and B-Lines classes still, however the Ribs, Pleura and A-Lines show significantly worse results with signs of the model overfitting the dataset. There is still significant work to be done in ensuring that good results are possible on unseen data, and a smart approach to data augmentation is necessary keeping in mind the preservation of information necessary to segment these classes in the original image. The developer should also be willing to sacrifice some overall IOU results in order to ensure a better fit to unseen data. The patient specific holdout dataset mentioned in the test section should also be checked and expanded in order to ensure that it represents the overall dataset well, and does not contain any unusual patterns or data points that may affect the segmentation results.

The final model using U-NET 3+ Deeply Supervised architecture, with a single channel input and seven channel output can be expected to segment images in near real time on modest hardware. Confirmation of this must be acquired through more thorough testing of a deployed model, which is beyond the scope of this project.

This project also makes no assumptions about the medical value of each class label that is being segmented for. Moreover, the value of each lung ultrasound artefact may even change depending on prior medical information extracted by a non-ultrasound medical assessment. For example, a clinician will in most cases be able to identify the patient's ribs through an external inspection of the torso, without necessitating seeing them on the ultrasound scan. Through the use of loss function adjustments and merging unnecessary classes into the background, the proposed model can be easily modified with professional medical feedback.

Another unfortunate loss of a static image approach to lung ultrasound segmentation is the loss of a dynamic view of the lungs in motion. As the dataset and trained models only lend themselves to individual image segmentations, there is little to no information gained from the continuous processing of consecutive lung images. The clinicians view of the dynamic motion may be enhanced

by viewing an ultrasound scan with real time segmentation, but this is also another avenue for further exploration.

Ultimately, however effective the model may be it should be further stressed that it cannot replace a trained medical practitioner. Such a tool does not possess the ability to diagnose medical conditions, which must be made by a professional. This tool will however be able to aid a clinician in making such decisions, easing training and improving decision making, leading to better patient care.

Future Work

Throughout the course of this project, I was able to identify a range of potential areas of investigation for my work that due to time constraints was unable to be completed. There are likely to be methods that would further increase the accuracy of my final segmentation, as well as make it more useful for medical practitioners in potentially everyday use. Additionally there are some research questions that are unanswered, and warrant investigation.

The Intelligent Ultrasound dataset contains a very large selection of images, however not all of these are labelled with an approved mask. It may be possible to use an unsupervised machine learning approach that does not rely on labelled data in order to develop a segmentation model. Through the combination of my supervised approach and an unsupervised method, a further improved semi supervised deep learning model could make use of all the data available and has the potential to improve upon my results.

Additionally, further expert approved image segmentation masks could be created for the unlabelled data. I expect incorporating new data would further improve my model, however this is likely to be a case of diminishing returns. The dataset is already quite large, and to see a significant improvement in segmentation accuracy it is likely that the dataset would have to be greatly expanded. If this was to be done, priority should be given to a select number of images from different patients and videos. This would increase the data diversity at the lowest effort to segment the data, having the greatest potential effect on the model.

A further improvement of the dataset could be an effort to discern between Confluent B-Lines and B-Lines as distinct classes. It is unclear the amount of effort versus the potential clinical reward for this task, which should be put to healthcare professionals before being decided upon. Investigating the images with a greater height than width could also improve the dataset if they can be modified into a roughly uniform proportion with the rest of the data. This would allow us to train the model in a more uniform fashion, and extract the same amount of information from each image by warping the original scan less. Further to this point, identifying how to get the maximum amount of information from images with great variations in pixel numbers is an important task within such a varied dataset. There exist a range of augmentations to adjust image quality which could be useful. however if the absolute number of pixels dedicated to displaying the lungs in these images are all similar, then a more sophisticated approach may be required for cropping out certain sections of the background class.

The use of the Jaccard function in place of intersection over union in the hybrid loss function suggests that there could be a more optimal loss function. A quick search of the literature offers some similar IOU based alternatives, similar to the improvement of using focal loss over cross entropy loss. Furthermore, the standalone loss functions in the hybrid loss produce very dissimilar accuracies. This suggests that a weighted hybrid loss function could produce even better segmentations. Using some values α, β, γ say we could define our hybrid loss as below, giving more

weight to seemingly better loss functions. This would require a large number of runs in order to find the optimal ratios and best results.

$$l_{hybrid} = \alpha l_{iou} + \beta l_{focal} + \gamma l_{ms-ssim}$$

Additionally, there may be further optimisations achievable by configuring the alpha and gamma parameters of the focal loss function. However it may be required to emphasise certain classes for practical reasons, including how critical to patient care misclassifying a certain feature could be.

The class guided module showed an improvement for some classes, and may still be useful at a further point in the project. In general, the module aim to solve the problem of more aggressive segmentation that incorrectly classifies the images. However, the results suggest that most classes are experiencing the opposite under segmentation problem. If the model was improved to such an extent that it was over segmenting images, then this could be a useful tool in improving results. In the meantime, it may be possible to adjust the model architecture to only provide the classification module for the classes where results are shown to have been improved by reducing the false positive classification rate.

A wide range of additional data augmentations are available and suitable to use for lung ultrasound, and those used within this project could be expanded and improved upon. There are various avenues to explore within the transforms used in this project, and expanding this to include other augmentation techniques.

The maximum rotation values and the probabilities of applying our existing transforms could be optimised. It seems that a maximum rotation of 35 degrees is too high, and is leading to reduced segmentation results. Brightness and contrast changes showed no improvement, however reducing the probability and magnitude of these changes is still worth investigating for improvements. Introducing Gaussian noise into the image shows the potential to increase the model's robustness, however the amount of noise likely needs reducing. There are also other methods available to add noise into an image, which are worth experimenting with.

There are also many other augmentation techniques which exist in the literature for lung image structures and ultrasounds which should be more thoroughly investigated to expand the available data. A random cropping technique could allow us to further expand our dataset for example. However this would necessitate that the entire segmentation mask is preserved in the new image to allow for full scale information, which is required in order to properly classify parts of the image. In general we can conclude that for our dataset we must be very careful to preserve all the necessary information needed to segment the image.

As U-NET is only one model within a broad field of image segmentation, it is important to consider a wide range of possible neural network structures for our model. U-NET itself has many different variants, and a thorough exploration of the literature may show other improvements that can be incorporated into the model. Further enhancements to the U-NET architecture could come from improved skip connections and a faster processing time. The dataset should also be examined with respect to models other than U-NET in order to fully understand the advantages and disadvantages of using different architectures by comparing results, and U-NET should not be assumed to be the model type best suited to lung ultrasound segmentation without a full exploration of different architectures on the dataset.

When it comes to assessing the model's performance, intersection over union is a good but limited metric. Using other methods of evaluation can expose deficiencies in segmentation predictions, and

a combination of functions such as dice coefficient and precision can provide a more comprehensive overview of the evaluation, and any future work should investigate this alongside IOU.

The practical purpose of this project must also be kept in mind. In order to fully measure the success of the project, it should be deployed alongside a standard ultrasound probe in order to properly assess the inference time of the model. If it is performing more quickly than expected, it may be possible to make the model more complex by increasing the image size or number of filters. Likewise, if it is underperforming then the model must be made simpler by reducing these features.

Once an appropriate model with a sufficient inference time is deployed, then the next step is to examine its usefulness in medical practice. By assessing the tool in clinical diagnosis and training, this could provide useful directions about how the model may be optimised for practical use by clinicians of varying experience. Within patient care, we may decide to emphasise a class with more critical care implications, while sacrificing others to improve patient diagnosis. Similarly, a training model may choose to focus on more commonly seen classes in order to guide the trainee's learning. This could be done by merging certain classes into the background if they are deemed not useful, or by training our model to focus on certain areas more via the use of focal loss parameters. Expert feedback and practice should be sought for these decisions in order to ensure that any changes have a positive overall effect.

Once a good tool is available and approved by medical professionals, it must then be more thoroughly tested in patient care. For example, we should seek to answer whether the tool can successfully segment certain classes in practical clinical settings. We should also ask whether the tool segments all patients to a similar level, or whether it falls short for certain gender, ethnic or age demographics to name a few. If the model is not confirmed to perform well across all societal groups, then certain patients will receive a different level of care when using the tool. Clinical testing will also be helpful for providing feedback as to whether the model fails to segment any common scenarios, and could provide insights of potentially common misclassifications or gaps in the dataset. The overall robustness of the model to new and unseen data could be very confidently evaluated by high quality clinical feedback.

Research Contribution

This project has shown that it is possible to create a good deep learning model in order to segment lung ultrasound images using U-NET. We show a high IOU result for each class simultaneously, with the potential to segment in near real time.

Additionally this project has built upon the results proposed for U-NET 3+, showing the a class guided module has limitations, and must be applied smartly to a suitable segmentation problem.

This project has also provided a first exploration of the labelled portion of the lung ultrasound dataset. Due to the sheer number of image samples, this is a unique collection of images and evaluating the dataset critically is important in order to extract maximum value from it. Over the course of the project, I have sought to critically examine the data in order to identify both its strengths and weaknesses. Any future users of this dataset should be able to examine this work in order to build upon my progress, and gain a better insight into how to use the data in order to train machine learning models.

As there are no publicly available datasets of lung ultrasound images, there is an absence of data driven approaches for segmenting these images in the literature. This work has shown that such an approach is possible, and furthermore can segment such an image in close to real time. The

development of machine learning approaches for segmenting a lung ultrasound is therefore a novel step forwards in the literature, and my project suggests that there is even further progress to be made in producing an optimal segmentation. In addition, this approach has focused on segmenting multiple different image features, where prior work has focused on a minimal number of distinct features. Certainly the Intelligent Ultrasound dataset is worth exploring further in order to develop an unsupervised or semi supervised approach. This underscores the transformative nature of artificial intelligence tools and the potential of this project to directly transform patient care, by increasing the effectiveness and accuracy of diagnosis via lung ultrasounds.

References

Achim, A., Allinovi, M., Anantrasirichai, N., Bull, D., Hayes, W. 2017. Line Detection as an Inverse Problem: Application to Lung Ultrasound Imaging. *IEEE Trans Med Imaging* 36(10), pp. 2045-2056. doi: 10.1109/TMI.2017.2715880.

Ahmad, M., Ahmed, I., Asif, M. and Khan, F.A. 2020. Comparison of Deep-Learning-Based Segmentation Models: Using Top View Person Images. *IEEE Access* 99. doi: 10.1109/ACCESS.2020.3011406.

Albumentations. 2023. *Transforms.* Available at: https://albumentations.ai/docs/api_reference/augmentations/transforms/ [Accessed: 30 August 2023].

Arora, A. 2020. *U-Net A PyTorch Implementation in 60 lines of Code*. Available at: https://amaarora.github.io/posts/2020-09-13-unet.html [Accessed 17 September 2023].

Brownlee, J. 2020*. A Gentle Introduction to Cross-Entropy for Machine Learning.* Available at: https://machinelearningmastery.com/cross-entropy-for-machine-learning/ [Accessed: 12 August 2023].

Brownlee, J. 2021. *How to Choose an Activation Function for Deep Learning.* Available at: https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/ [Accessed: 17 September 2023].

Brownlee, J. 2021. *How to Choose an Optimization Algorithm.* Available at: https://machinelearningmastery.com/tour-of-optimization-algorithms/ [Accessed: 20 September 2023].

Brox, T., Fischer, P. and Ronneberger, O. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv*. arXiv:1505.04597v1. doi: 10.48550/arXiv.1505.04597.

Chandra, P. 2023. *IoU Loss Functions for Faster & More Accurate Object Detection.* Available at: https://learnopencv.com/iou-loss-functions-object-detection/ [Accessed: 9 August 2023].

Cheng, D. and Lam, E.Y. 2021. Transfer Learning U-Net Deep Learning for Lung Ultrasound Segmentation. *arXiv.* arXiv:2110.02196. doi: 10.48550/arXiv.2110.02196.

Chen, G., Dai, Y., Li, L. and Zhang, J. 2023. Rethinking the unpretentious U-net for medical ultrasound image segmentation. *Pattern Recognition* 142. doi: 10.1016/j.patcog.2023.109728. Springer, Cham.

Datagen. 2023. *Image Segmentation: The Basics and 5 Key Techniques.* Available at: https://datagen.tech/guides/image-annotation/image-segmentation/ [Accessed: 10 July 2023].

Dollár, P., Girshick, R., Goyal, P., He, K. and Lin, T.Y. 2017. Focal loss for dense object detection. In *Proceedings of IEEE international conference on computer vision.* Venice, Italy, 22-29 October 2017. (pp. 2980-2988). doi: 10.48550/arXiv.1708.02002

Draelos, R. 2020. *Segmentation: U-Net, Mask R-CNN, and Medical Applications.* Available at: https://glassboxmedicine.com/2020/01/21/segmentation-u-net-mask-r-cnn-and-medical-applications/ [Accessed: 30 July 2023].

Fortuner, B. 2022. *Loss Functions.* Available at: https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html [Accessed: 18 September 2023].

Gao, Y., Lee, L.H. and Nobel, J.A. 2021. Principled Ultrasound Data Augmentation for Classification of Standard Planes. *arXiv.* arXiv:2103.07895v1. doi: 10.48550/arXiv.2103.07895.

Hassan, A. 2022. *Multi-class Focal Loss.* [Loss Function]. Available at: https://github.com/AdeelH/pytorch-multi-class-focal-loss [Accessed: 3 September 2023]

Hasty. 2023. *Gaussian Noise.* Available at: https://hasty.ai/docs/mp-wiki/augmentations/gaussian-noise [Accessed: 25 September 2023].

Hu, C., Huo, L., Jiao, L. and Tang, P. 2021. Refined UNet v3: Efficient end-to-end patch-wise network for cloud and shadow segmentation with multi-channel spectral features. *Neural Networks* 143, pp. 767-782. doi: 10.1016/j.neunet.2021.08.008.

Huang, H. et al. 2020. UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation. *arXiv.* arXiv:2004.08790. doi:10.48550/arXiv.2004.08790.

Huang, H. et al. 2020. *Figure 2, Illustration of how to construct the full-scale aggregated feature map of third decoder layer* X_{De}^{3}. UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation. *arXiv.* arXiv:2004.08790. doi:10.48550/arXiv.2004.08790.

Huang, H. et al. 2020. *Figure 3, Illustration of classification-guided module (CGM).* UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation. *arXiv.* arXiv:2004.08790. doi:10.48550/arXiv.2004.08790.

Huang, H. et al. 2020. *UNet-Version.* [Models and Loss Functions]. Available at: https://github.com/ZJUGiveLab/UNet-Version/tree/master [Accessed: August 5 2023].

Huang, S. and Tsai, T. 2022. Refined U-net: A new semantic technique on hand segmentation. *Neurocomputing* 495, pp. 1-10. doi: 10.1016/j.neucom.2022.04.079.

Image Segmentation. 2010. *A figure of an original image, and a segmented image.* Available at: https://commons.wikimedia.org/wiki/File:Image_segmentation.png [Accessed: 14 September 2023].

Kalim, A.R. 2020. *Logging with Weights & Biases.* Available at: https://towardsdatascience.com/logging-with-weights-biases-da048e3cbc8b [Accessed: 2 August 2023].

Kaur, A., Kaur, L. and Singh, A. 2021. GA-UNet: UNet-based framework for segmentation of 2D and 3D medical images applicable on heterogeneous datasets. *Neural Computing and Applications* 33, pp. 14991-15025. doi: 10.1007/s00521-021-06134-z.

Khandelwal, R. 2021. *Different IoU Losses for Faster and Accurate Object Detection.* Available at: https://medium.com/analytics-vidhya/different-iou-losses-for-faster-and-accurate-object-detection-3345781e0bf [Accessed: 9 August 2023].

Klingler, N. 2023. *Image Segmentation with Deep Learning (Guide).* Available at: https://viso.ai/deep-learning/image-segmentation-using-deep-learning/ [Accessed: 5 July 2023].

Liang, J., Siddiquee, M.M.R, Tajbakhsh, N. and Zhou, Z. 2018. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. *arXiv.* arXiv:1807.10165. doi: 10.48550/arXiv.1807.10165.

Liang, J., Siddiquee, M.M.R, Tajbakhsh, N. and Zhou, Z. 2018. *Figure 1, UNet++: A Nested U-Net Architecture.* UNet++: A Nested U-Net Architecture for Medical Image Segmentation. *arXiv.* arXiv:1807.10165. doi: 10.48550/arXiv.1807.10165.

Mishra, M. 2020. *Convolutional Neural Networks, Explained.* Available at: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939 [Accessed: 12 July 2023].

Moshavegh, R., Hansen, K., Møller-Sørensen, H., Nielsen, M. and Jensen, J. 2019. Automatic Detection of B-Lines in In Vivo Lung Ultrasound. *IEEE TRANSACTIONS ON ULTRASONICS, FERROELECTRICS, AND FREQUENCY CONTROL.* 66(2), pp. 309-317. doi: 10.1109/TUFFC.2018.2885955.

NHS. 2021. *Ultrasound scan.* Available at: https://www.nhs.uk/conditions/ultrasound-scan/ [Accessed: 11 July 2023].

O'Sullivan, C. 2023. *U-Net Explained: Understanding its Image Segmentation Architecture.* Available at: https://towardsdatascience.com/u-net-explained-understanding-its-image-segmentation-architecture-56e4842e313a [Accessed: 27 August 2023].

Persson, A. 2021. *Machine-Learning-Collection.* [Machine Learning and Deep Learning Tutorials] Available at: https://github.com/aladdinpersson/Machine-Learning-Collection [Accessed: 31 July 2023].

Persson, A. 2021. *PyTorch Image Segmentation Tutorial with U-NET: everything from scratch baby.* Available at: https://www.youtube.com/watch?v=IHq1t7NxS8k [Accessed: 30 July 2023].

PyTorch. 2023. *PYTORCH DOCUMENTATION.* Available at: https://pytorch.org/docs/stable/index.htm [Accessed: 30 September 2023].

PyTorch. 2023. *SOURCE CODE FOR TORCH.UTILS.DATA.DATASET*. [Random split of dataset function]. Available at: https://pytorch.org/docs/stable/_modules/torch/utils/data/dataset.html#random_split [Accessed: 4 August 2023].

Quora. 2017. *What is the difference between Contrast and Brightness?* Available at: https://www.quora.com/What-is-the-difference-between-Contrast-and-Brightness [Accessed: 25 September 2023].

Raju, G. and Shereena, V.B. 2022. Medical Ultrasound Image Segmentation Using U-Net Architecture (from the Advances in Computing and Data Sciences, Kolkata, India 27-28 April, 2022). *Communications in Computer and Information Science* 1613. doi: 10.1007/978-3-031-12638-3_30.

Secherla, S. 2021. *Understanding Optimization Algorithms in Machine Learning.* Available at: https://towardsdatascience.com/understanding-optimization-algorithms-in-machine-learning-edfdb4df766b [Accessed: 19 September 2023].

Sharma, P. 2019. *Computer Vision Tutorial: A Step-by-Step Introduction to Image Segmentation Techniques (Part 1).* Available at: https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/ [Accessed: 10 August 2023].

Shereena, V., Raju, G. 2022. Medical Ultrasound Image Segmentation Using U-Net Architecture. In: Singh, M., Tyagi, V., Gupta, P.K., Flusser, J., Ören, T. eds. *ICACDS.* Kumool, India, 22-23 April 2022. Springer, Charm. doi: 10.1007/978-3-031-12638-3_30

Silva, J. 2021. *Segmentation Models Pytorch.* [Functions]. Available at: https://github.com/jlcsilva/segmentation_models.pytorch/blob/master/segmentation_models_pytorch/losses/jaccard.py [Accessed: 20 September 2023].

Simonelli, J. 2020. *Data Augmentation with Albumentations.* Available at: https://jss367.github.io/data-augmentation-with-albumentations.html [Accessed: 25 September 2023].

Tas, S. 2020. *How To Evaluate Image Segmentation Models?* Available at: https://towardsdatascience.com/how-accurate-is-image-segmentation-dd448f896388 [Accessed: 3 July 2023].

TheThinkingMan. 2022. *U-net-architecture.* Available at: https://commons.wikimedia.org/w/index.php?curid=123267582 [Accessed: 15 September 2023].

Ultrasound Solutions Corp. 2022. *What Are The Benefits Of Ultrasound – 13 Advantages of Using It.* Available at: https://www.uscultrasound.com/ultrasound-benefits/ [Accessed: 14 July 2023].

Wang, L. 2022. *UNet 3+ Fully Explained – Next UNet Generation.* Available at: https://medium.com/@mlquest0/unet-3-fully-explained-next-generation-unet-2a8e204e4cf9 [Accessed: 2 August 2023].

Wang, Z. 2020. Deep Learning in Medical Ultrasound Image Segmentation: a Review. *arXiv.* arXiv:2002.07703. doi: 10.48550/arXiv.2002.07703.

Wang, Z., Simoncelli, E. and Bovk, A. 2004. Multi-Scale Structural Similarity For Image Quality Assessment. (from The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, Pacific Grove, CA, USA. 9-12 November 2003). *IEEE.* pp. 1398-1402 Vol.2, doi: 10.1109/ACSSC.2003.1292216.

Wikipedia. 2023. *Strucutral similarity.* Available at: https://en.wikipedia.org/wiki/Structural_similarity [Accessed: 7 September 2023].

Wikipedia. 2023. *U-Net*. Available at: https://en.wikipedia.org/wiki/U-Net [Accessed: 3 July 2023].