ABSTRACT

As indicated by the title of the Thesis, "Brain Magnetic Resonance Imaging Segmentation System in MS with Deep Neural Networks" the aim was to develop a system using deep neural networks, to analyze medical images.

The main purpose was to create a system for automated analysis of magnetic resonance imaging of type T2 and the accurate identification of areas indicating the presence of Multiple Sclerosis. To achieve this, it was necessary to create a prediction model that was previously required to acquire knowledge from suitably formatted information such as medical images accompanied by corresponding masks and to be able to accurately predict the outcome for new data.

The tools used to develop the prediction model were the Python programming language and deep learning libraries TensorFlow and image processing library OpenCV. The web application was implemented using Angular framework and Flask.

In conclusion, the objective of automated detection of the aforementioned areas was achieved with an effectiveness of 70%. This effectiveness is quite satisfactory; however, it could be increased even further with the provision of additional data and the application of specialized techniques concerning data preprocessing.

Giorgos Adamides – University of Cyprus, 2024

Brain Magnetic Resonance Imaging Segmentation System in MS with Deep Neural Networks

Giorgos Adamides

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

at the

University of Cyprus

Recommended for Acceptance

by the Department of Computer Science

June, 2024

APPROVAL PAGE

Master of Science Thesis

BRAIN MAGNETIC RESONANCE IMAGING SEGMENTATION SYSTEM IN MS WITH DEEP NEURAL NETWORKS

Presented by

Giorgos Adamides

Research Supervisor

Prof. Constantinos Pattichis

Committee Member

Prof. Andreas Aristidou

Committee Member

Prof. Andreas Panayides

University of Cyprus

June, 2024

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to the staff of the Electronic Health Laboratory of the Department of Computer Science at the University of Cyprus for their immediate assistance with any issues that arose during the length of the diploma thesis.

In particular, I would like to make special mention of Professor Constantinos Pattichis and Professor Christos Loizou. The assistance provided by 3aHealth, both for the technical and theoretical specifications of the thesis, was also invaluable and enabled me to successfully conduct the work. I am thankful for their expertise and willingness to share their knowledge, which greatly enriched the quality of this work.

CREDITS

Special thanks are owed to the eHealth Laboratory of the Department of Computer Science at the University of Cyprus. The lab's meticulous efforts in collecting and preparing the dataset were crucial. Their detailed attention to ensuring the dataset was comprehensive, well-annotated, and meticulously prepared not only facilitated my analysis but also enhanced the reliability and validity of my research findings.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

Medicine has made tremendous progress over the years from the beginning of modern history to the present. More specifically, with the rapid technological development of the 20th century, medical science has benefited to the greatest extent possible. Accordingly, the approach to diagnosis, monitoring, and treatment of various diseases has radically changed.

The development of innovative machines such as CT scanners, MRI scanners, and ultrasound has contributed to precise diagnosis and monitoring. The evolution that medical technology experienced through technological progress was equally seen in pharmaceutical technology, having the ability to develop original formulations that contribute not only to the treatment of the patient but also to maintaining or even improving the patient's quality of life. Naturally, at this stage, there are no applicable solutions for all diseases and conditions. However, for the existing ones, if not applied in the appropriate timeframe, the situation appears bleak and no one can guarantee their success.

Here lies the motivation for my Individual Dissertation titled 'Brain MRI Image Segmentation System with Deep Neural Networks', which aims to solve the problem of detecting areas in magnetic resonance images indicating the presence of Multiple Sclerosis, with the goal of implementing an automated solution to simplify the diagnostic process through the doctor and to address any problems that may arise in a timely manner.

1.2 Challenges

It is a given that the challenges that exist in such an endeavor are numerous and vary in form. Initially, solving such problems involves the importance of artificial intelligence, which in simpler words means providing knowledge to the computer, shaping this knowledge to be understood by the computer, and then making effective predictions based on the knowledge previously acquired.

One of the challenges that had to be addressed was the excessive computational resources needed to train such a system, as the process is very demanding. Another challenge is the small volume of data. To train a model without any initial knowledge, large amounts of data are required, which, however, are not readily available and require a great effort to create, as will be explained later. Another obstacle in implementing these issues, due to the nature of the problem, is the percentage of effectiveness that is though as satisfactory. In the case of medical problems, it is not enough for the results to function with mediocre effectiveness, as human lives are at risk and the slightest detail can be crucial both for the progression of the disease and for the patient's health trajectory. Health by definition is not just the absence of disease or the elimination of the illness, but it is physical, mental, and motor well-being.

1.3 Strategic Objectives

Concluding the short-term goal of the Individual Dissertation, it was to study and become familiar with using deep neural networks, so as to be prepared for the implementation of the long-term goal of automated detection of areas of interest in brain magnetic resonance imaging for the evaluation of Multiple Sclerosis. In addition, different models and techniques had to be used in order to conclude to the most appropriate handling of data.

1.4 Contents

In the next chapter, number two, titled "Convolutional Neural Networks," the field of deep learning is described, which is the fundamental basis for the use of

Convolutional Neural Networks. This chapter outlines the operation of these networks and analyzes in depth the functioning of the U-Net Convolutional Neural Network, which will be later tested for detecting multiple sclerosis lesions. Other variations of the U-Net architecture like Attention U-Net, ResUNet-a, U-Net++ and many more will be used to compare the methodologies and results. Their operations include the preprocessing of the data they receive as input and analysis. At the end of the chapter, it is understood that to have a proper evaluation of performance, appropriate metrics must be established as a standardized measure for comparing the results.

Moving forward, a detailed explanation of the available input and output data is provided. In between, the application of the Convolutional Neural Network U-Net to solve the problem is presented, and the final results achieved using it are shown.

In chapter "Multiple Sclerosis Segmentation Framework” the problem of segmenting areas of interest in Brain Magnetic Resonance Imaging for the evaluation of Multiple Sclerosis is presented, and the purpose, which is the automated detection of these areas, is outlined. Then, an in-depth analysis of the data in their initial form is conducted, as well as the process to transform these data into a uniform, understandable form for training. At this point, normalization is applied, and the first prediction model is created, which does not provide sufficient effectiveness. Subsequently, the steps of adding Batch Normalization layers, Data Augmentation, increasing the training data, modifying the learning rate, and ultimately restoring the original clarity of the data as regards improving effectiveness are described. At each point, a comparison with the previous attempt is made, and the differences are highlighted. Last, an overview of the framework that encapsulates the automated model is shown, together with the capabilities that such an online responsive tool has to offer.

In the last chapter, number six, titled "Conclusion" the evaluation of the final results is presented, clarifying their significance. A comparison is also presented with research that aimed to solve an identical problem. Finally, a reference is made to a possible future research and how it could improve the existing results of the current research.

Chapter 2

Convolutional Neural Networks

2.1 Introduction

Initially, to understand the logic on which convolutional neural networks are based, basic knowledge of biology and more specifically the functioning of the human brain must be learned. As indicated by the name Convolutional Neural Networks [31], these are networks consisting of layers, with each layer containing neurons, aiming to process the input as it transitions between neurons and ultimately understand and interpret the initial input. A neuron is a node where the input is processed, and various features extracted from the analysis are considered to direct it to the corresponding neuron of the next layer and ultimately lead to a conclusion.

In the case of Convolutional Neural Networks, there is full connectivity, meaning there is the ability to transition from any node to another node of the next layer of the network. The layers that make up a Convolutional Neural Network are defined as the architecture of that specific network. The way Convolutional Neural Networks operate makes them suitable for machine learning with inputs of images. Some advantages of Convolutional Neural Networks over other methods of machine learning, include the relatively low preprocessing of input data and the adaptation of processing during the learning [1]. The term 'deep neural networks' refers to the multiple layers that these networks contain and by extension, the size of the processing intended for the input.

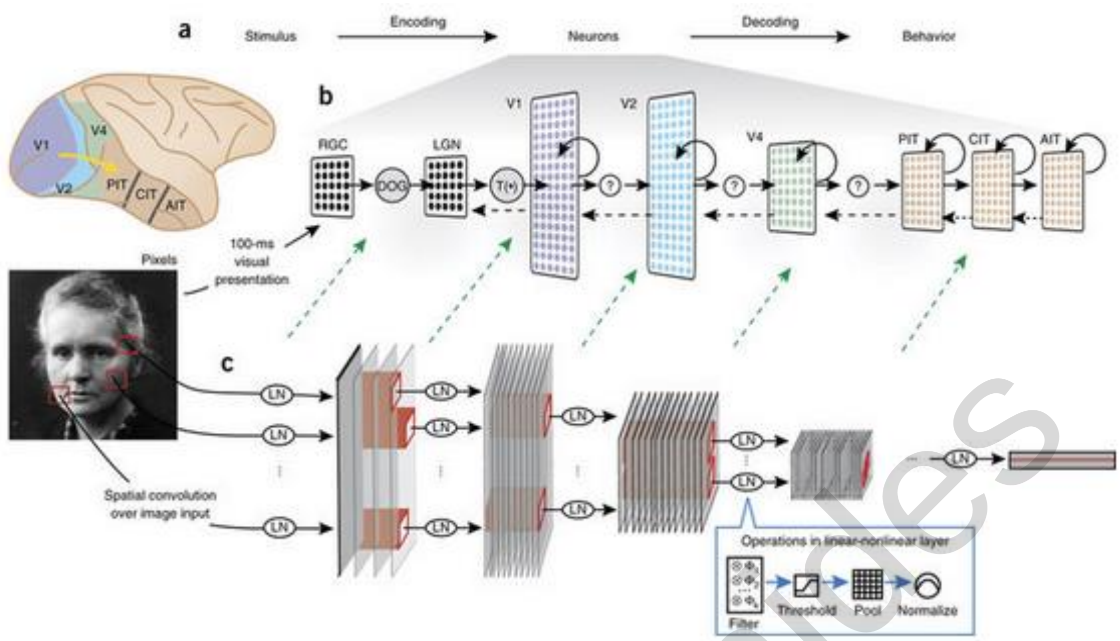


Figure 2.1 Comparison of the function of the human brain with a Convolutional Neural Network in understanding a visual image (Source: https://medium.com/nerd-for-tech/understanding-convolutional-neural-network-cnn-9f5ec8a308ac, Last accessed 01/05/24)

The image above contrasts the function of the human brain and Convolutional Neural Networks in understanding visual information. The human brain receives information through the eyes and directs it through the neurons to derive a conclusion. Convolutional Neural Networks attempt to simulate the same process.

2.2 U-Net

U-Net [2] is a convolutional neural network structured to facilitate the training of a model using images as input, with the ultimate goal of enabling the model to predict the contents of another image provided as input, based on the knowledge with which it was already been trained on.

Focusing on the operation and architecture of this particular architecture, we can easily observe that it is divided into two main parts: encoder and decoder. The first part, the encoder, receives the input, which in the context of the Thesis, consists of

MRI images of the brain and attempts to analyze the content by applying Convolutional 3x3 ReLU, Max Pooling 2x2, Up-Conv 2x2, and Conv 1x1 layers in order to extract feature maps and conclude to a prediction. Regarding the data provided as input and the training of the system, it is characterized as Supervised Learning because along with the input image, a mask is also provided as input for each of the images, which depicts the information indicating that in specific areas there is a positive indication for the presence of multiple sclerosis, so the model can be trained to predict those and calculate its' performance. The output image is a black and white mask that shows the pixels that were verified as positive in white and the negative ones in black. The process of splitting the areas of the original input image into positive and negative through the extraction of the mask is called Semantic Segmentation [3].

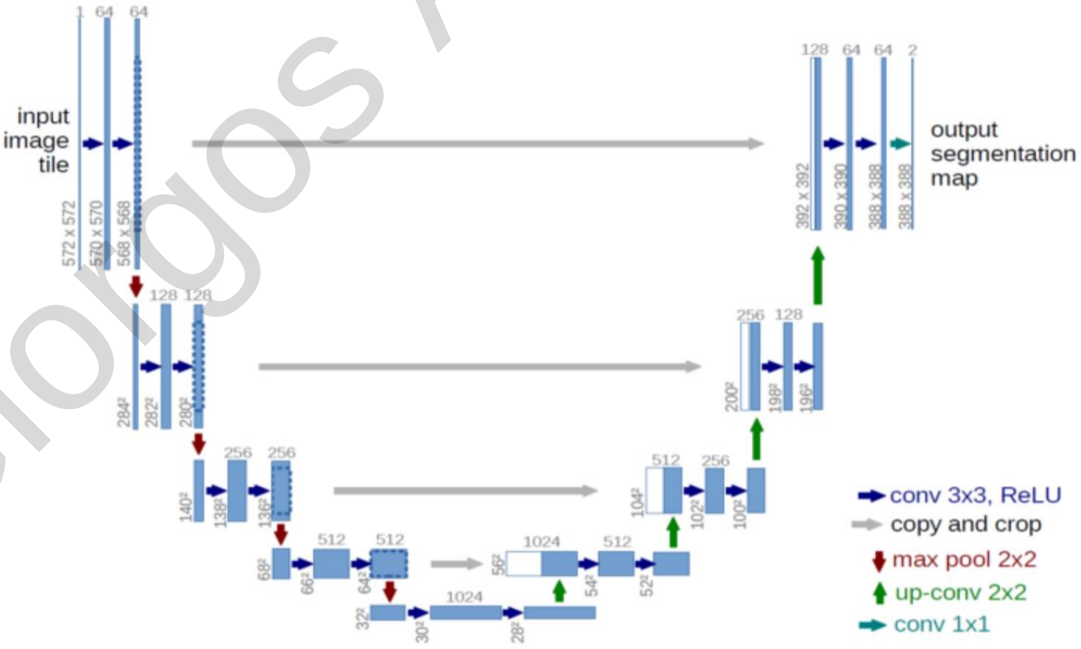


Figure 2.2 U-Net architecture (Source: https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/, Last accessed 02/05/2024)

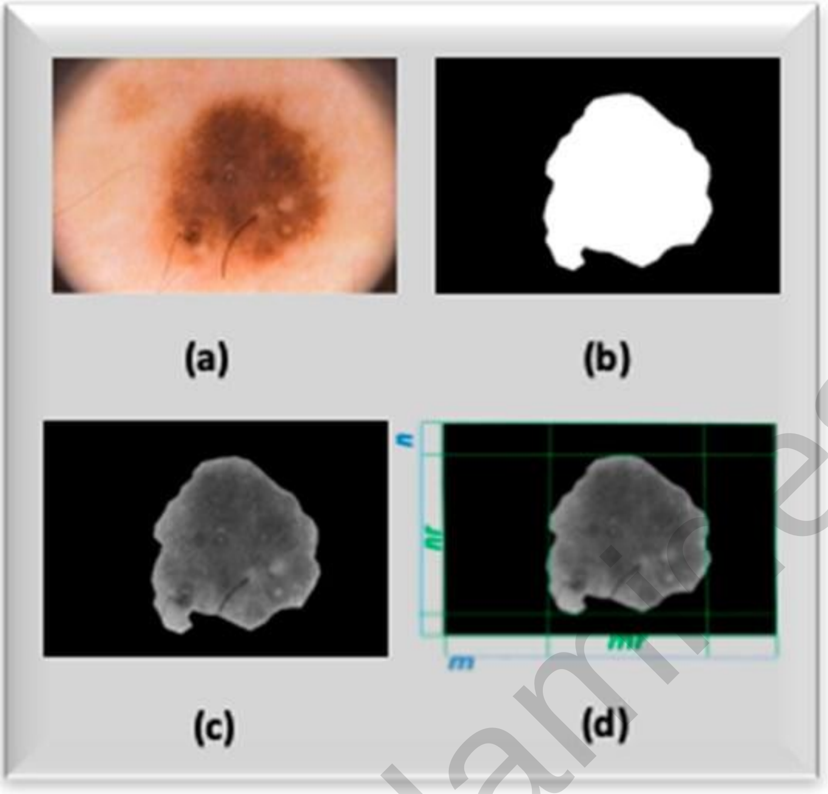


Figure 2.3 Image (a) contains an area of skin cancer and image (b) is the mask that indicates the positive in cancer presence areas as white and negative areas as black (Source: https://www.mdpi.com/2079-3197/7/3/44, Last accessed: 02/05/2024)

2.3 U-Net Variations

The U-Net architecture has become a standard model for medical image segmentation because of its effectiveness in handling a wide range of imaging tasks. Several iterations of the original U-Net design have been built in recognition of its success in order to solve certain issues and enhance the performance of medical image processing. In this section, some of the most important U-Net variants are covered. These models made several adjustments to the conventional U-Net architecture in an effort to increase the model's generalizability across various medical imaging datasets and enhance segmentation accuracy.

Adding attention gates (AGs) into the U-Net design, the Attention U-Net [23] assists the model in focusing on target structures of various sizes and forms. These attention processes selectively suppress the less significant characteristics while emphasizing the prominent ones that are essential to specific tasks. This is especially helpful in medical imaging, where it's crucial to precisely locate things like tumors or lesions. Attention U-Net is flexible enough to handle the complexities of medical image segmentation because its attention mechanisms may change dynamically based on the input picture.

ResUNet [24] is an architecture for medical image segmentation that extends on the U-Net architecture that makes use of residual learning blocks. By include these additional blocks, the issue of the vanishing/exploding [34] gradient is mitigated, enabling the training of deeper networks with more efficiency. ResUNet's ability to allow gradients to flow across the network allows it to support tasks demanding deep anatomical segmentation while permitting the learning of increasingly complicated features without sacrificing efficiency.

U-Net++, or Nested U-Net as it is also called [22], presents a series of nested, dense skip paths that increase information flow across the network by allowing more accurate localization when combined with contextual data from the contracted path. This architecture minimizes the semantic gap between the feature maps of the encoder and decoder modules in while improving feature representation at multiple resolution scales. The goal of U-Net++ is to offer better segmentation with enhanced fine-grained detail acquisition, which is crucial for medical imaging diagnosis.

TransUNet [25] combines the powerful self-attention mechanisms of transformers with the solid structure of CNNs. The hybrid model accurately processes the global dependencies inside the picture by utilizing the advantages of both systems. When tasks like organ segmentation need precise prediction and contextual awareness, TransUNet typically performs exceptionally well because the transformer part examines the picture as a whole and helps to clarify the complex anatomical components seen in medical pictures, while the other part operates as the classic U-Net architecture.

2.4 Image Preprocessing

For the analysis of medical images and the proper training of our model, it is essential to apply some form of processing [4] to the images that are given as input into the network. The structure of the network determines the processing that the images undergo, and the corresponding filters are used to implement this processing. But what is typically done, is to apply some processing to the images prior to giving them as input to the network, in order to ease the network's feature identification and extraction.

Initially, filters serve multiple purposes such as smoothing, enhancing features, reducing, enlarging, and many more. To accomplish those, what is known as a Kernel is used. The Kernel is essentially a matrix that determines the weight of neighboring pixels. To make this more understandable, a kernel can have any size and shape that seems appropriate, with different weights for each cell. The kernel is then applied to each pixel of the original image, and results in a new value for the corresponding pixel which is assigned to another empty as of now image, which is finally turned into the final processed image.

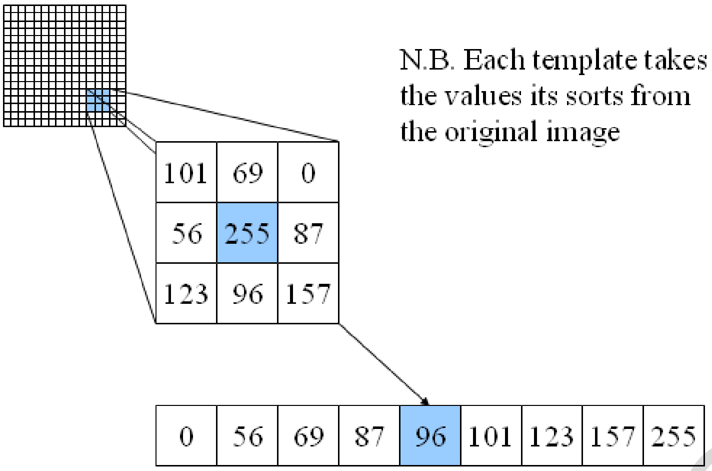

N.B. Each template takes the values its sorts from the original image
101
69
0
56
255
87
123
96
157
0
56
69
87
96
101
123
157
255

Figure 2.4 Application of Median filter to a single pixel (Source: https://www.southampton.ac.uk/~msn/book/new_demo/median/, Last accessed: 02/05/2024)

2.5 Image Resolution

The processing of images provided to the network utilizes the following layers. The Convolutional Layer applies a 3x3 square kernel to the input image to produce the processed image. The resulting image is reduced by 2 pixels in both the horizontal and vertical axes each time the Convolutional Layer is applied, because there are no neighboring cell values for the first 2 and last 2 rows and columns since they are on the perimeter of the image, and thus the network chooses to ignore these values because it does not use any padding. In the Max Pooling Layer, we also have a 2x2 square kernel from which we draw the maximum value of the square and designate this particular value as the new value of the corresponding element. Consequently, the resulting image becomes that half the size of the input image.

The second part of the U-Net primarily aims for local accuracy using Transposed Convolutions (Up-Conv 2x2). The Transposed Convolution uses a 2x2 kernel, which is applied to the original image and the result is of higher clarity as it contains double the information. Finally, the Convolution Layer is used again, but with a different kernel size of 1x1. All the images intended for training undergo this process to produce the model that we will use in the future for prediction.

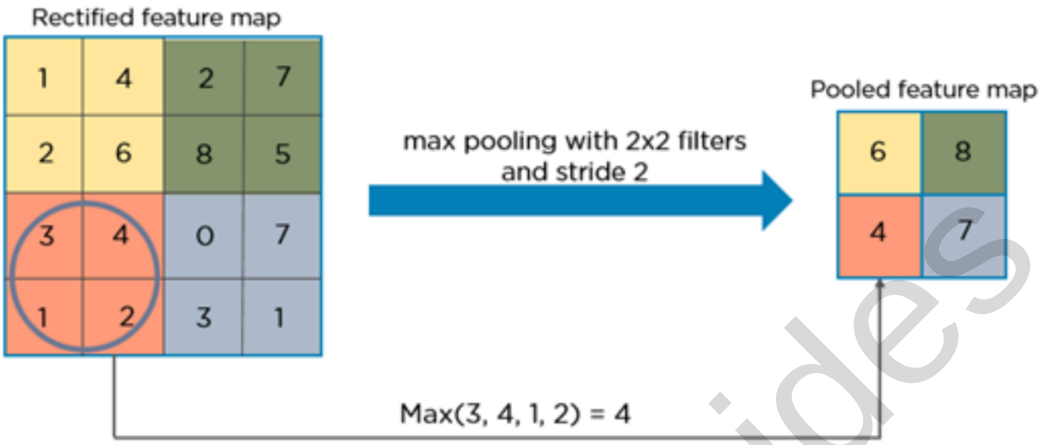


Figure 2.5 Max Pool filter with a 2x2 kernel (Source: https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47, Last accessed: 02/05/2024)

2.6 Evaluation Metrics

To evaluate the effectiveness of the prediction model that has trained, it is imperative to set evaluation metrics so that the model can be compared with other various available prediction models or instances of the same model with different configurations. Since the prediction involves images and particularly the identification of regions in images, the metrics that are easily visible are the percentages where the prediction model and the real assessment agree and those where they disagree.

For example, in a medical image concerning the detection of tumors, the trained prediction model may identify certain areas as positive, and the doctor will also identify his areas of interest. The determination of the areas by the doctor is considered absolutely correct, and thus the evaluation metrics [5] emerge as follows:

Metric	Description
True **P**ositive (TP)	Areas for which both the prediction model and the doctor agree are positive regarding the presence.
True **N**egative (TN)	Areas for which both the prediction model and the doctor agree are negative regarding the presencE.
False **P**ositive (FP)	Areas which the prediction model has identified as a positive and the doctor as a negative in presence area.
False **N**egative (FN)	Areas which the prediction model has identified as a negative and the doctor as a positive in presence area.

Table 1 Evaluation Metrics description

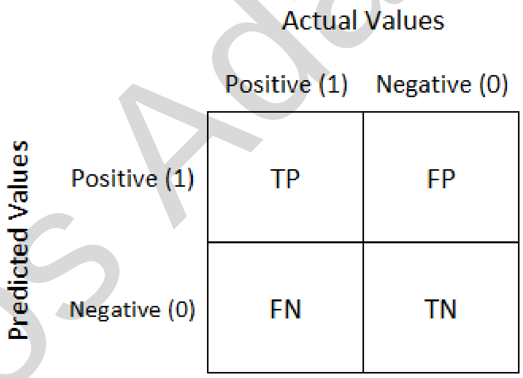
The above table separates the cases when both the true value and the prediction agree on the positive value of the area, then the area is marked as True Positive. When both agree on the negative value of the area, then the area is marked as True Negative. In cases where they disagree, when the true value is positive, then the area is a False Negative. When the two disagree, and the value of the true value is negative, then the area is a False Positive.

These specific metrics form the basis for producing additional metrics in case we want to measure differently the importance of some of these metrics. Some of these are:

Metric	Equation
DSC (Dice Similarity Coefficient)	$\frac{2\ \mathrm{x}TP}{2\ \mathrm{x}\,TP\ +\ FP\ +FN}$
$Precision$	$\frac{\mathrm{TP}}{\mathrm{TP}\ +\ FP}$
$TPR\ (True\ Postitive\ Rate)$	$\frac{TP}{TP\ +\ FN}$
$FPR\ (False\ Positive\ Rate)$	$\frac{FP}{FP\ +\ TN}$

Table 2 Additional evaluation metrics

The above table outlines the derivation of additional evaluation metrics based on the metrics that were previously mentioned. These additional metrics consist of the Dice Similarity Coefficient [6], which is in simpler words the percentage of overlap, the accuracy, and the ratios of positive and negative areas.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 2.6 Confusion Matrix (Source: https://towardsdatascience.com/taking-the-confusion-out-of-confusion-matrices-c1ce054b3d3e, Last accessed: 02/05/2024)

The confusion matrix visually represents how the basic evaluation metrics arise based on the comparison of the values of the prediction model and the ground truth.

Chapter 3

Multiple Sclerosis Segmentation Framework

3.1 Introduction

Magnetic Resonance Imaging (MRI) refers to the method of visualizing the interior of an individual using a magnetic resonance scanner. Initially, a specific part of the body is chosen for the magnetic resonance scanner to image, based on the doctor's assessment of the area that needs further examination. The result obtained using this method is two-dimensional representations of the casing and the interior, which, when stacked, produce a three-dimensional representation of the specific area.

For the identification of areas indicative of Multiple Sclerosis, it is necessary for the patient to undergo an MRI [30] of the brain to visualize its internal part. At this stage, it is essential to clarify that Multiple Sclerosis is a chronic, autoimmune, demyelinating disease affecting the central nervous system and subsequently the various functionalities it performs. Some aspects include speech, vision, coordination of various body parts, multiple brain functions, as well as its contribution to the functioning of the muscular and digestive systems [32]. Multiple Sclerosis is characterized as the condition where the outer shield of nerves, called myelin, shows deterioration, which in turn causes instability in the transmission of messages both from the body to the brain and the other way around from the brain to the body [3].

It also must be clarified that this disease is characterized by relapses, each of which can have consequences that finally add up to form the overall clinical picture of the patient. All the above converge on the fact that early diagnosis and immediate treatment of any changes can be proven beneficial for the patient. This is because the symptoms of each relapse can be treated with specific medication, which can prevent problems and avoid permanent and irreversible effects. Hence, there is the need for automated detection of these specific areas with as much accuracy as possible. To accomplish this, a predictive model must be created that accepts as input an MRI image of the human brain and produces a corresponding output image that highlights areas where the damaged myelin is observed, if such areas with the corresponding characteristics exist [33].

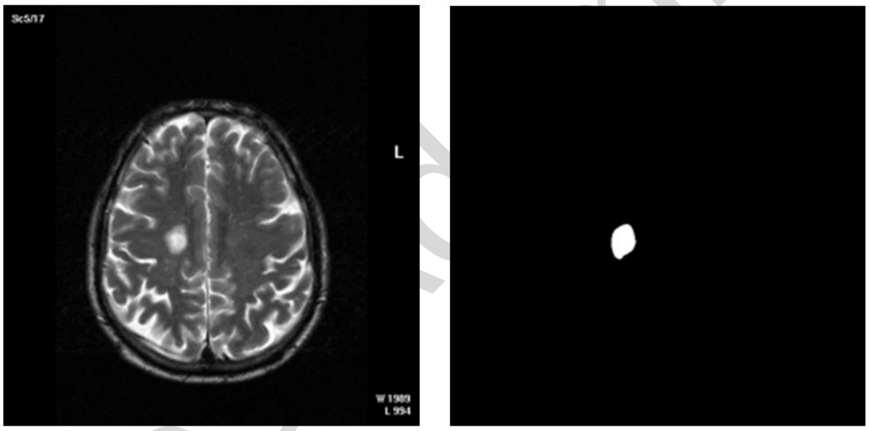


Figure 3.1 MRI image of the brain with the corresponding mask that indicates multiple sclerosis lesions

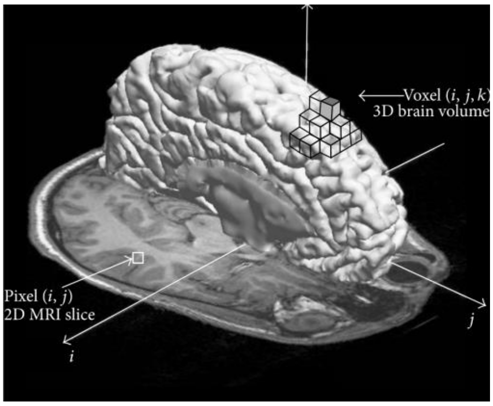


Figure 3.2 3D Representation of the brain using 2D slices
(Source: https://www.hindawi.com/journals/cmmm/2015/450341/, Last accessed 03/05/2024)

3.2 Initial Data format and Preprocessing

Magnetic resonance imaging data that were used are publicly available through the Electronic Health Laboratory of the University of Cyprus, which in collaboration with the Cyprus Institute of Neurology and Genetics managed to construct the mentioned dataset [9]. This specific dataset includes 38 patients, with each patient represented as a folder, and each of these 38 folders containing another two folders. The first of these two folders contains the MRI images from the patient's initial examination and the corresponding information about areas indicating Multiple Sclerosis. The second folder contains MRI images of the same patient, taken 6 to 12 months after the first examination, and also includes corresponding information about areas indicating Multiple Sclerosis [10] [11] [12].

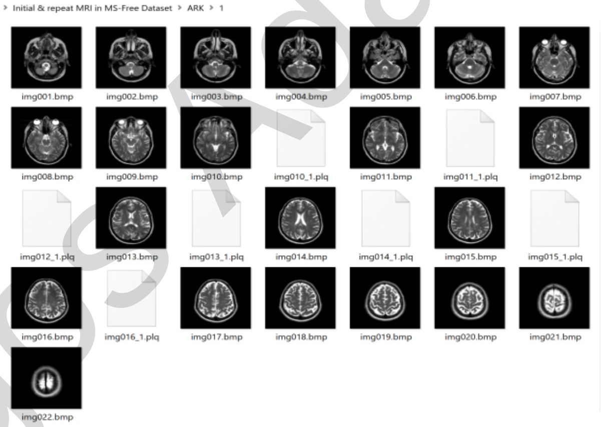


Figure 3.3 MRI brain images of the patient "ARK" during the first examination accompanied by the relevant lesion information.

The MRI images are distinguished into two categories: those in TIF format with dimensions of 512x512 and those in Bitmap format with dimensions of 378x378. There are 1027 images in TIF format and 811 images in Bitmap format, totaling 1838 images. Each image is accompanied by a MATLAB format file, which characterizes the areas where myelin damage is observed. If the corresponding file is missing, it means that the doctor did not identify any areas matching the conditions specified.

The MATLAB files represent the area marked as positive for Multiple Sclerosis by storing a number of outline points that match the content of the MRI image as follows:

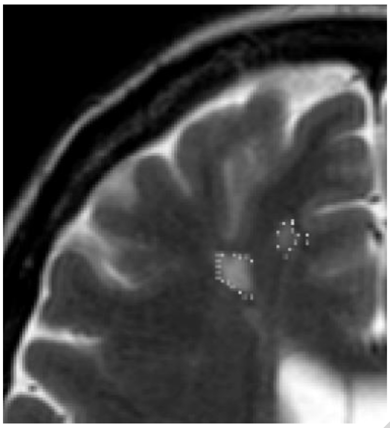


Figure 3.4 The points with white color represent the outline of the positive area.

The goal is to convert the points into a single outline, which will then be used to mark all the points of each area and ultimately produce the corresponding mask image for each MRI image.

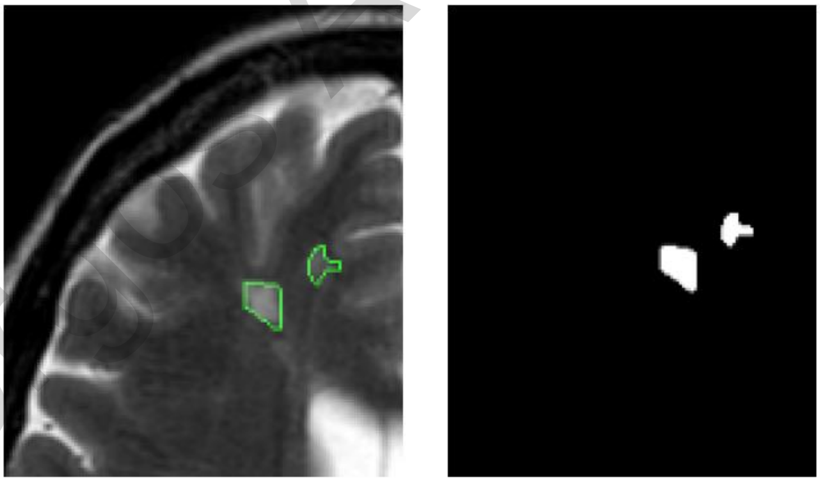


Figure 3.5 The outline of the areas in green color after merging the individual points and marking the positive areas on the mask image.

Moving forward, it is necessary to create a blank image of the same size as the MRI image and use the polygons derived from the outlines to mark the areas with a value of 1, so they will appear as a white color.

By repeating this process for all areas and all the available images, all the MRI images are now accompanied by another image, the mask, that indicates the lesions. For the purposes of data uniformity, and after the above actions, all images along with their masks were converted to TIF format and to size of 512x512.

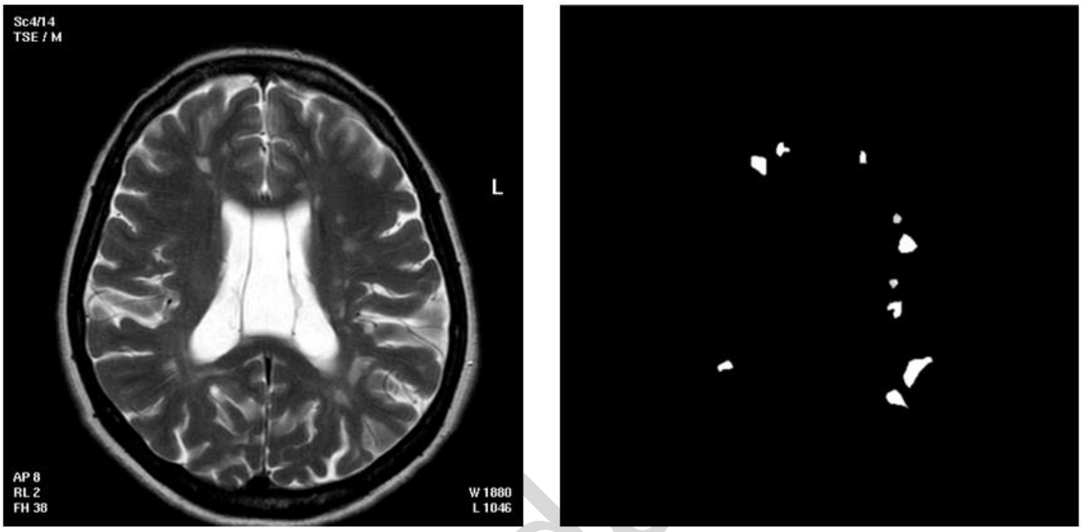


Figure 3.3.6 Final form of MRI image and corresponding mask image.

3.3 Data Processing and Experiments

After the preprocessing of the data was completed, there were 1838 MRI brain images and the corresponding 1838 mask images available to train the prediction model and to evaluate its effectiveness.

To achieve the two goals of learning and testing the prediction model, it was necessary to divide the newly created set again based on the patient. This separation is important because having identical or very similar images in the training and testing sets could bias the results and not accurately reflect the real capabilities of the prediction model. To ensure a balanced distribution of data, it was decided that 80% of the patients, or 30 patient folders, would be assigned for training the model, and 20% of the patients, or 8 patient folders, would be used for testing the model. These percentages and the direct reliance on a small number of available data represent a relatively sufficient volume of data for learning and a respectable number of tests to accurately assess the effectiveness of the efforts.

3.3.1 Initialization and Normalization

The first step in creating the prediction model was the normalization of the data. The purpose of data normalization is to adjust the data to a uniform format with the same minimum and maximum values, so that when they are fed as input to the model's architecture, no imbalances occur.

Focusing on the current problem and since the images are in grayscale, the normalization should produce images containing pure black, values equal to 0, and pure white, values equal to 255. Various approaches can achieve this specific result.

The chosen approach was Histogram Normalization. Initially, a histogram is a graphical representation of the distribution of the pixel values of each image. When an image does not show a uniform distribution of these values at the extremes of the graph, it is understood that there are no pixels with absolute black or white values or both.

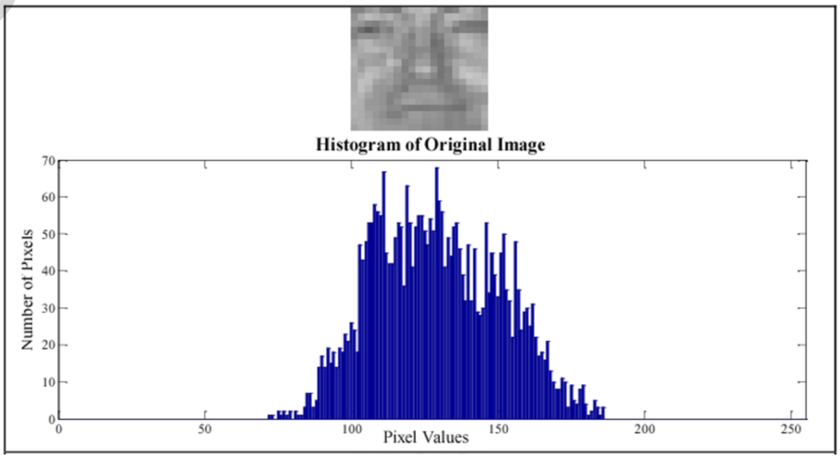


Figure 3.7 Example of a grayscale image and the histogram (Source: https://www.semanticscholar.org/paper/Face-detection-using-boosting-and-histogram-Derhalli-Nufal/5b48ed62242d6665a3d3685b4cfcbb63b1abbcd0, Last accessed 03/05/2024)

From the above image with its' histogram, it is easy to see that the pixel values range from 75 to 180. The purpose of histogram normalization is to modify these specific values to range from 0 to 255. To achieve this without losing the existing correlations of values, a uniform modification must be applied to all pixels in a single image.

For each image, the original value (g(x, y)) the minimum (g_{min}) and maximum (g_{max}) histogram values are identified, the upper limit (GWM) is set to 255, and the lower limit (BWM) to 0, and the new pixel value (f(x, y)) for the resulting image is calculated using the formula:

$$f(x,y)=\frac{GWM-BWM}{g_{\max}-g_{\min}}.(g(x,y)-g_{\min})+BWM$$

In this way, the data for all images now have a uniform structure and can be provided for training the prediction model, ensuring a smoother validation process.

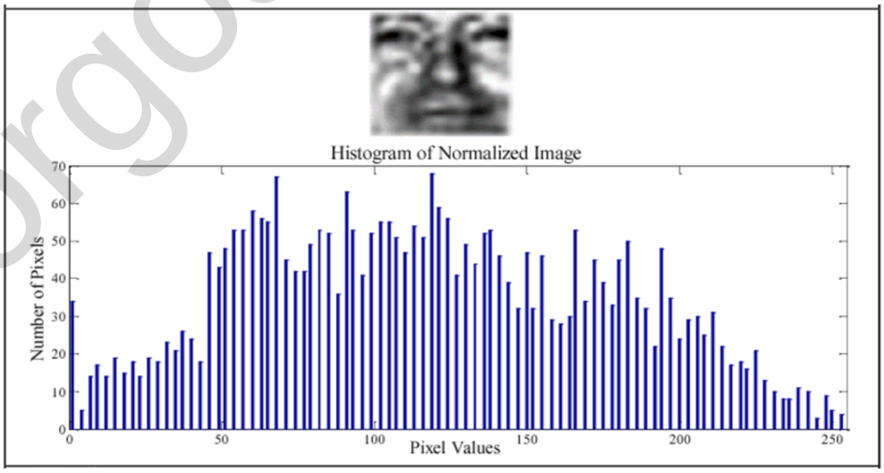


Figure 3.8 Image and histogram after normalization (Source: https://www.semanticscholar.org/paper/Face-detection-using-boosting-and-histogram-Derhalli-Nufal/5b48ed62242d6665a3d3685b4cfcbb63b1abbcd0, Last accessed 03/05/2024)

3.3.2 Batch Normalization

The results achieved using normalization alone were not satisfactory, necessitating the implementation of additional methods to optimize the operations and simultaneously increase the model's effectiveness.

One method was the addition of Batch Normalization layers [14] to the already defined levels of the U-Net architecture. To understand the idea behind this action, it is first necessary to comprehend the function of Batch Normalization layers.

Batches are subsets of the original data set, which are sequentially processed through the predefined layers of the architecture. By processing these specific batches, the activation vectors of the network are accordingly shaped. The Batch Normalization layers provide normalization of these activation vectors, hence they are applied immediately before or immediately after the activation method.

In the U-Net architecture, the first five layers represent the encoding of the data, and the remaining layers represent the decoding. Therefore, their batch normalization layer placement is recommended only in the first part of the architecture.

3.3.3 Data Augmentation

Applying various modifications to the original form of the data as well as to the training algorithm of the prediction model, notable progress was observed, though a relatively satisfactory level of effectiveness was not achieved. This prompted the question of what was lacking compared to other problems and their solutions, such as the problem of automatic detection of the subclavian artery.

The answer was clear and directly related to the available volume of data [17]. It is evident that an additional number of data helps in feature extraction, better parameter tuning during the training process, and in more accurate evaluation of results.

The process of acquiring data is a very time-consuming one, where the physician has access to a number of magnetic resonance images from various patients and needs to carefully study each one to locate the area of interest and mark the points that make up the outline of that specific area to subsequently create the corresponding ground truth mask. As it can be understood, taking into account these specific conditions, the solution to the problem should rely on the available data.

The solution of Data Augmentation [15], meaning the enhancement and increase of data by modifying existing ones, was now the only way forward. Specifically, data augmentation is the process where transformations, filters, and other modifications are applied to available images so that new images are generated and added to the training set. It is worth noting that any changes applied to an image must also be applied to the image's mask, so that a corresponding mask of the produced image is created. If every change is applied to all images, then the total number of images doubles each time. Thus, each change applied to an image exponentially increases the available number of data, which is expected to have a positive impact on the training and evaluation process. Therefore, it was decided to apply 3 modifications to the images, resulting in an eightfold increase in the final number of data [16].

The first change decided upon for the images, in order to generate new ones, was the randomized rotation of the image and consequently of the corresponding mask within the range of -20 to 20 degrees. The reason for this specific modification was observed because there were images in the original data that exhibited a slight inclination. Therefore, by adding slightly rotated images, the factor of image tilt as a characteristic for locating areas of interest could potentially be eliminated.

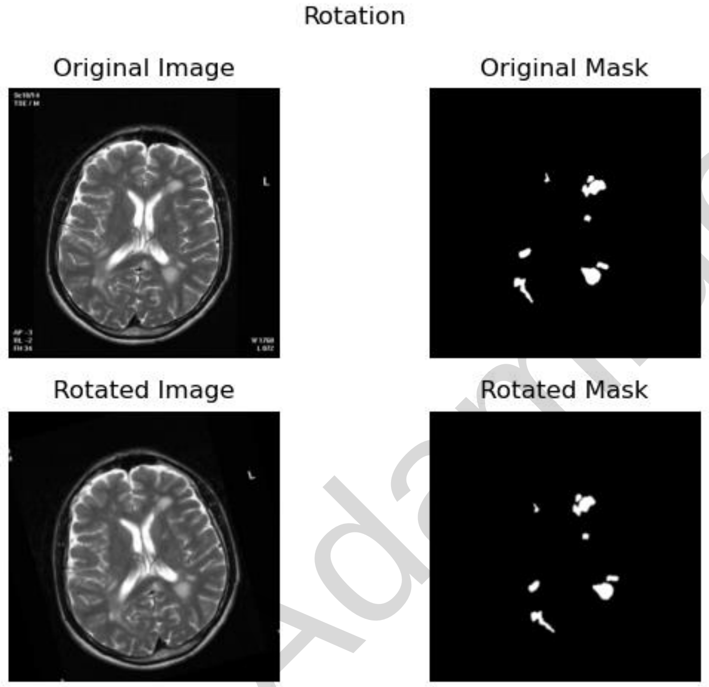


Figure 3.9 Original and Rotated image with their masks

The next change, which was applied to the new dataset now available after the randomized rotation of the images, was the horizontal flip of the pixels. That is, the elements on the left side are mirrored to the right and the other way around.

This swapping of image pixels can simulate one hemisphere of the human brain with the other, as in the new image the areas that belonged to one hemisphere appear on the opposite one. Thus, the algorithm is trained with the presence of areas of interest on both sides and does not interpret the side of their existence as a criterion for locating these areas. That is, in the case where a test image contains identical areas on the opposite side, it would be easier for the prediction model to identify them than previously.

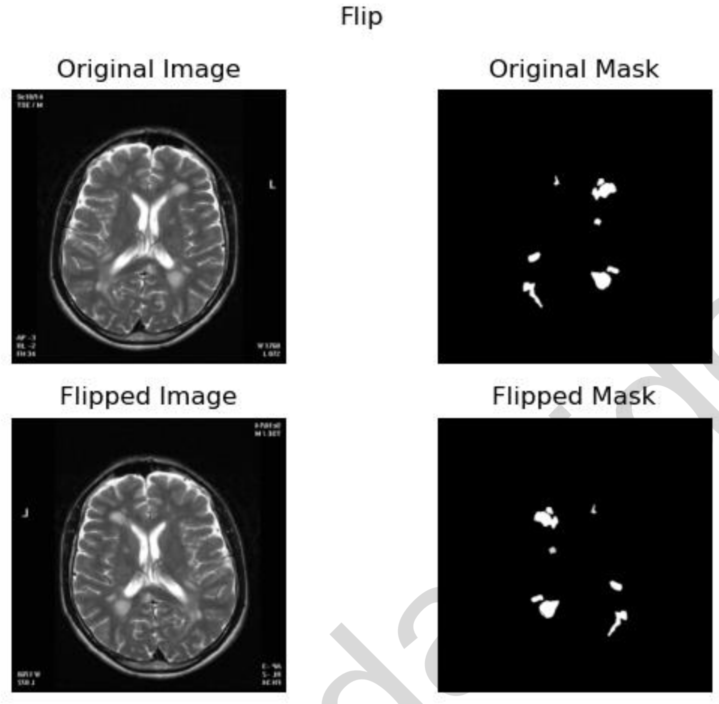


Figure 3.10 Original and Flipped image with their masks

The last modification applied to the image set, which was now quadruple the size of the original, was the randomized zoom focusing on the 80-90% of the image and then readjusting it to convert the dimensions to the same size as the original dimensions. The reason for this conversion was because several images in the original set were observed to have different focuses among them. In this way, the dependence on the focus presented by each image as a training parameter for locating areas indicating Multiple Sclerosis is eliminated.

The difference in image focus is due to the transformation of the three-dimensional shape of the brain into the two-dimensional form of the images. The images that map the patient's brain at its extremes, due to the smaller size of the brain at the edges, appear to have less focus than the images that map the same brain closer to its center.

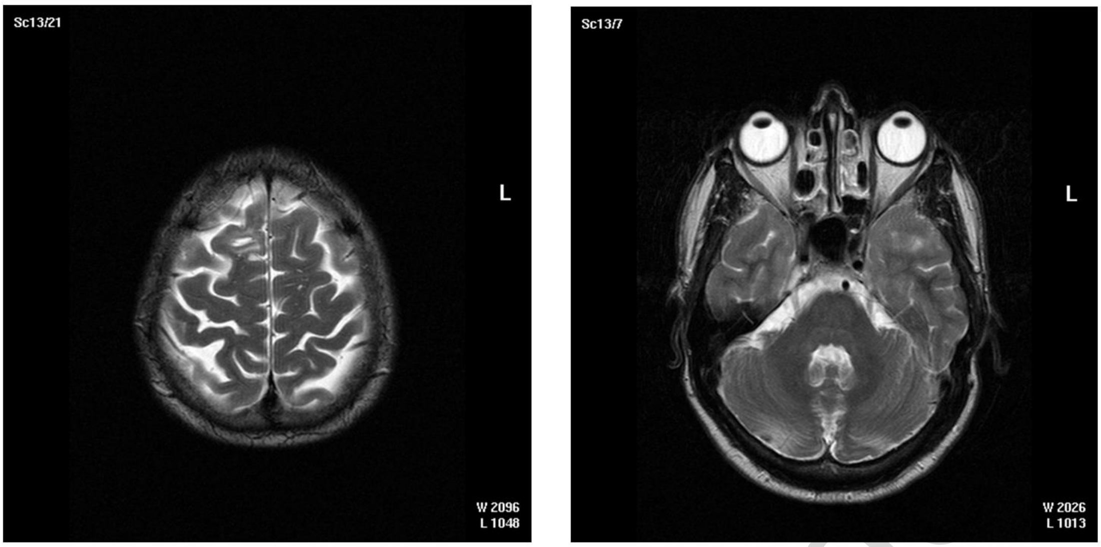


Figure 3.11 Focus difference between different brain slices

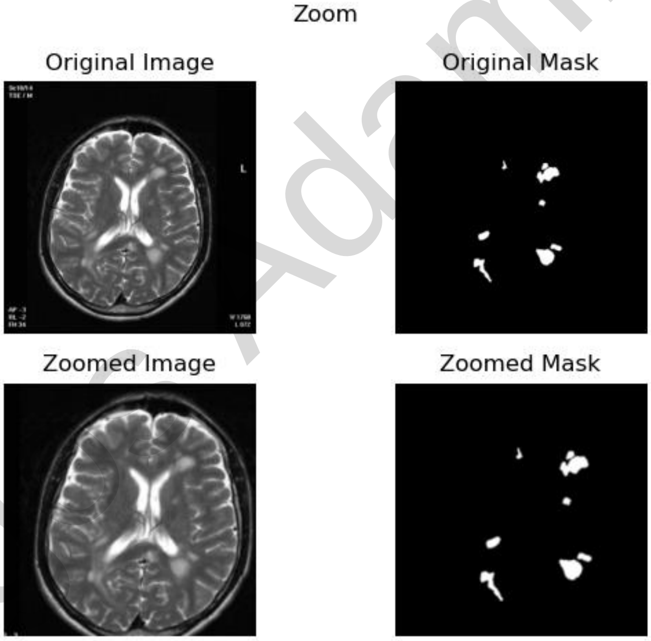


Figure 3.12 Original and Zoomed image with their masks

The final size of the training image set was eight times larger than the original. Thus, without any additional changes, the prediction model had to be created and then the effectiveness evaluated, to reveal to what extent and to what degree the Data Augmentation technique helped achieve an increase in effectiveness.

3.3.4 Learning Rate

Another parameter that required particular attention was the learning rate. The learning rate [18], determines the rate at which the weights in the training parameters are prone to change during the backpropagation stage of updating the weights. However, adjusting this specific parameter must be done with great care, as an excessive increase in this parameter can lead the prediction model to overfit the initial input images, resulting in an inability to readjust to the rest of the data. On the other hand, excessively reducing this variable can dramatically decrease training progress, thus significantly reducing progress per season and potentially failing to properly utilize and adapt all training parameters and getting stuck in a local optimum. The initial learning rate was set at 1e-6.

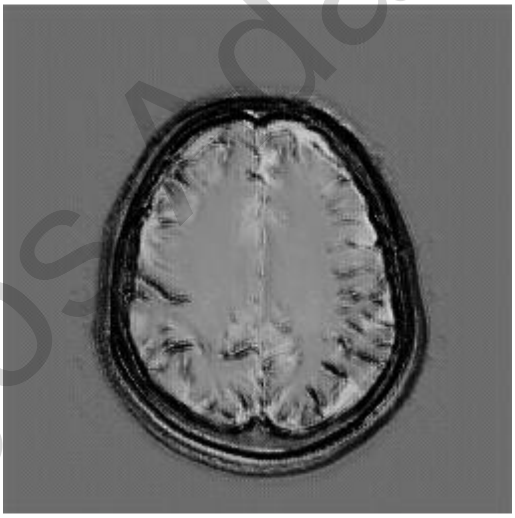


Figure 3.13 Example of a prediction with a high learning rate

During the first trial, the learning rate was adjusted from 1e-6 to 1e-4, which ultimately proved too high. As seen in the prediction example with such a learning rate value, the prediction model outlined the image contours in black and assigned grey values to the remaining pixels, so, intermediate values to balance the probabilities of black and white colours, resulting initially in what appeared to be a quite satisfactory effectiveness, which clearly did not correspond to reality.

In the second trial, it was decided to select an intermediate value, that of 1e-5, with the expectation that it would not present the overfitting problems exhibited by the higher value of 1e-4.

3.3.5 Resolution

As mentioned previously, the dimensions of the images were reduced to dimension of 256x256 from the original 512x512, impacting the resolution and the quality of the image. This change was made for performance reasons and to save computational memory. It is understandable that even with the reduced size of 256x256, the magnetic resonance imaging (MRI) images were still very demanding in terms of processing resources. However, this does not mean that no information was lost during the downsizing process [19], information that might have been particularly useful and could have aided further improvement in the learning process.

At this stage, the effort to increase the efficiency of the process of building the prediction model through the steps previously described becomes immediately apparent, as without these steps it would not have been possible to restore the original clarity of the images and to create the prediction model. Therefore, having this capability, the images were reverted back to their original size of 512x512, and the prediction model was trained again.

3.3.6 Transformation of Mask Areas into Rectangles

One of the attempted modifications in improving the brain MRI segmentation model was transforming the irregularly shaped areas within the masks into rectangles. This approach was based on the hypothesis that a standardized shape might simplify the learning process for the neural network. However, this method proved counterproductive. The U-Net architecture, designed to process images at a pixel-wise level, was less effective when dealing with these artificially standardized shapes. The complexity and variability of Multiple Sclerosis lesions, which are inherently irregular, meant that the rectangular masks failed to accurately represent the critical nuances in the images, leading to a degradation in model performance.

3.3.7 Increased Number of Filters

Increasing the number of convolutional filters within the U-Net layers was another strategy tested to enhance the model's ability to capture more detailed features from the MRI scans. Typically, a higher number of filters can allow for a richer representation of the input data. However, in this case, the addition did not lead to the anticipated improvements. Instead, the model became more prone to overfitting, learning noise and irrelevant details rather than generalizing from the training data. This adjustment did not contribute positively to the model's accuracy in segmenting MRI images.

3.3.8 Random Assignment of Filters in Data Augmentation

The implementation of data augmentation techniques aimed to increase the robustness of the model by introducing varied transformations to the training images. One specific experiment involved the random assignment of filters to different images, rather than applying a consistent set of transformations across all images. This approach intended to simulate a more diverse set of possible imaging scenarios.

3.3.9 Validation Readjustment

As of now, the validation set was selected randomly from the training data, which led to synthetic data from data augmentation being included in the validation set. The model's performance metrics could be affected by the overlap of training and validation data, which would give an overestimation on its generalization capabilities.

A new approach has been adopted to construct the validation set in order to address this issue and increase the model's ability to generalize on unseen data. Before the model training is being initialized, a validation set must be generated. In order to ensure that each of the training, validation and test set has a specific patients assigned to it, a grouping of patients to match the percentages that correspond to each set has to be performed.

This refinement ensures that each set has different patients, thereby eliminating any bias that might occur due to the leakage of data into the validation process. Strictly separating the data based on patients, sets the model to be trained and validated on distinct data samples, which improves its ability to generalize to new and unseen datasets. This approach enhances the robustness of the validation process and contributes to more reliable assessments of the model’s performance.

3.3.10 Skull Stripping

The skull stripping process, crucial for isolating brain tissue from other cranial structures, was another focal point for optimization. Given the original TIFF format of the MRI images, they were first converted to NIfTI format to utilize certain skull stripping algorithms designed specifically for this format. Despite this conversion, the process was largely unsuccessful. The conversion resulted in the loss of vital image details due to differences in how these formats handle data representation. The stripped images lacked critical information necessary for accurately identifying and segmenting the lesions, which impeded the effectiveness of subsequent segmentation processes.

These unsuccessful modifications highlight the challenges faced in optimizing deep learning models for medical image analysis. Each failed attempt provided valuable insights into the intricate balance required between data representation, model architecture, and learning process adjustments in developing effective diagnostic tools.

3.4 Transfer Learning

Transfer learning [26] is a widely used strategy in the context of deep learning, where a model trained for one particular task is reused to facilitate learning in a new but related task. This approach is useful in cases where a dataset for the new task can not cover the needs to support model training from zero, or if there isn't enough computing power.

Transfer learning relies on the ability to transfer the internal representations from a model trained on a complete and diverse dataset, such as ImageNet, to a new model that aims to serve a similar challenge. These internal parameters include the weights, the biases, and the architectural features that have been optimized to detect generic patterns and features across a broad range of inputs. The concept of transfer learning is based on the idea of utilizing already pretrained models as a foundational starting point for a task-specific model training.

These models have been trained on a huge amount of relevant information from large datasets and they are able to apply this knowledge to new and related tasks.

Researchers can rapidly accelerate the learning process by using a pretrained model, since it only needs to adapt its learnt pattern slightly in order to conform with the specific requirements of this new assignment.

Together with the other advantages, that transfer learning has to offer is the enhanced efficiency it brings to the training process. A model that relies on the transfer of knowledge typically requires a smaller number of epochs to achieve convergence, so it learns more quickly and does not use too much computing power. In areas where rapid deployment of accurate models is essential, such as medical diagnostics systems, this efficiency makes learning transfer a suitable option.

3.4.1 Transfer Learning for Multiple Sclerosis

In this case, transfer learning was implemented using pretrained models from the ImageNet dataset [27], which is mostly used for visual object recognition. The dataset consists of grayscale MRI images with a single colour channel, which does not respond to the input requirements of the commonly used pretrained networks that expect a three-channel RGB input.

To align the MRI data with these requirements, the grayscale images were converted into three-channel images by replicating the single grayscale channel across the RGB channels. This conversion made it possible to utilize the complex feature-detecting capabilities of models pretrained on ImageNet, which include a variety of architectures trained to recognize thousands of object categories in images.

3.5 Framework

A web application provides a simple and effective way for medical practitioners to improve diagnosis procedures, making it a necessary instrument in today's medical sector. In particular, the ease of use and speed of this procedure are the main benefits of an application that lets users provide MRI scans to receive predictions and the lesion outlines on the original images. Since this kind of platform is web-based and accessible through any typical web browser, it does not require specialist hardware or complicated software instalments. Regardless of geographical or technological constraints, the accessibility guarantees that medical practitioners may easily receive vital diagnostic information in a timely manner. Furthermore, the online application may be utilized easily on a variety of devices because to its' interoperability across multiple operating systems. The direct feedback and visualization provided by the application also assist in instant and accurate medical decisions, enhancing patient outcomes and streamlining the workflow within medical facilities.

3.6 Backend

The web application's development and operating tasks are significantly simplified by the use of a RESTful API [29] implemented using with the Flask framework, particularly when interacting with artificial intelligence models for medical image analysis. Flask is a lightweight, but powerful Python web framework that is appropriate for creating RESTful APIs since it is flexible and simplistic.

The Python-based Flask environment offers a uniform platform that makes it easier to integrate and manage the application's numerous components, especially the AI models and backend services. This is because the backend services and AI operations, such processing and analysing MRI images, are all done within the same Python environment, this homogeneity removes the hassle of managing several programming languages or frameworks. In this way, developers have access to a set of tools and libraries across the entire backend, ranging from web services to AI processing.

Using Flask makes the environment configuration easier since Python has efficient package management methods like pip and virtual environments. These tools guarantee that, across different development and production environments, all of the required libraries and dependencies are consistently updated and maintained. Libraries used for the Flask application, and libraries used for image processing or machine learning models, such as TensorFlow, can be handled under the same package ecosystem. Updates and cross checks for compatibility are simplified and potential dependency conflicts are minimized thanks to the uniformity in package management.

The RESTful API design makes it possible for client-server interactions to be carried out with greater efficiency, because it uses standard HTTP methods like GET, POST, PUT, and DELETE to make resource manipulation simple. The application's simplicity is especially helpful when handling complex data inputs and outputs, such as uploading MRI images and receiving the processed outputs plus the AI-generated predictions. The application is able to scale to meet the demands of a growing user base because of Flask's ability to handle several requests at once. This is relevant in a medical context where rapid processing of data is essential.

The backend is able to distinguish between normal image inputs and the DICOM format, which needs a different processing to align with the specifications of the other image type files. This is automatically done at the controller level on the backend, so the process for the all the other components of the system is seamless.

3.7 Frontend

The web application's frontend was developed using Angular, which is a modern framework that is well-known for its strong capabilities of creating dynamic and interactive user interfaces. Angular suits applications requiring complicated user interactions and real-time data updates, which are typical requirements in medical imaging applications.

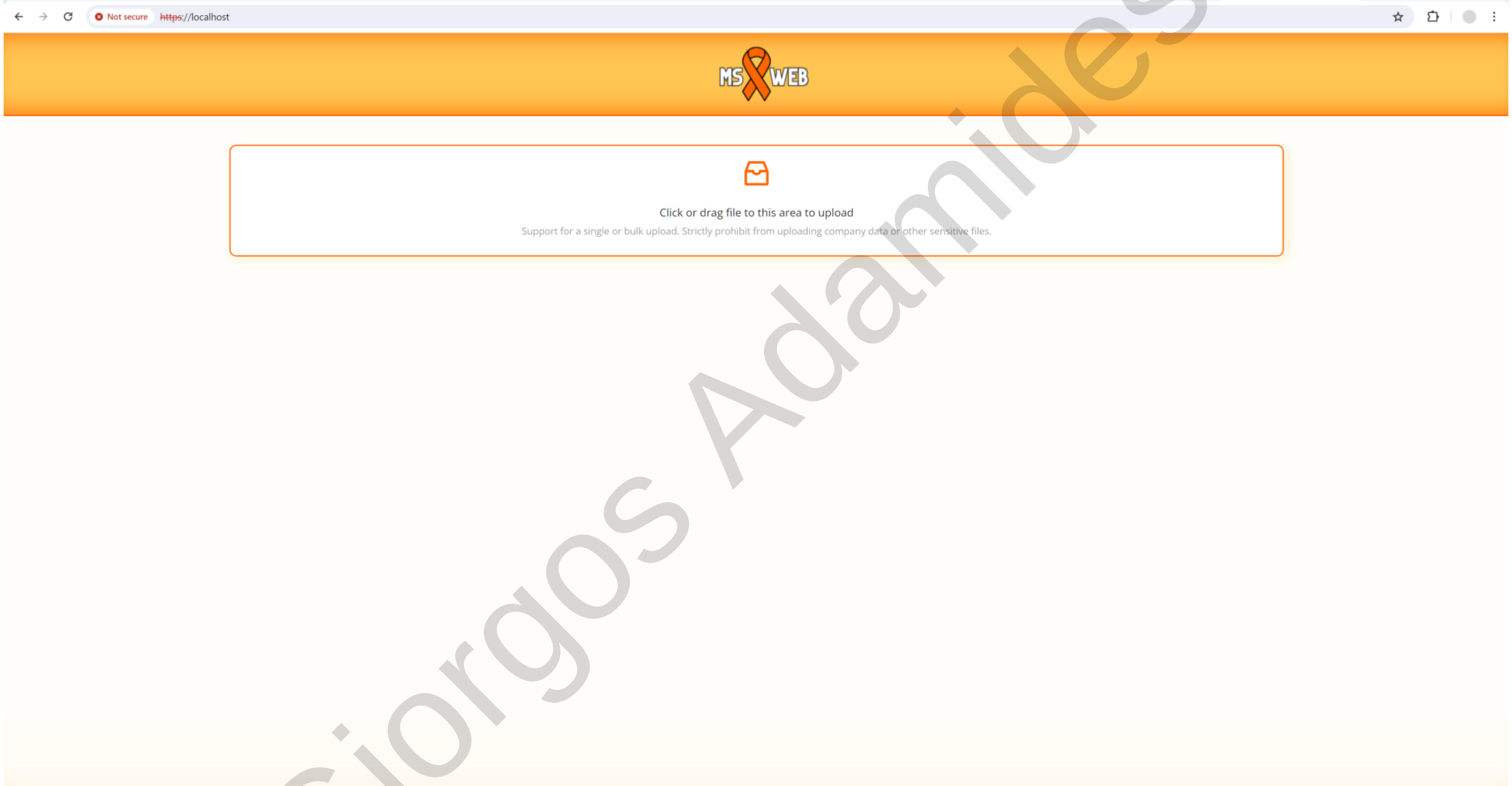
Given its component-based architecture, developers can create UI components that are highly reusable and maintainable. This is essential for medical web applications that require a variety of views and functionalities, such as demonstrating the image analysis outcomes. By making it possible to create an interface that is both responsive and easy to use, Angular improves the overall user experience. Its' data-binding capability makes sure updated data are consistently added to the user interface components without necessitating a complete page reload, which is essential for applications where medical staff need to see real-time data. The framework's state management feature makes it easier for users to navigate between to different application sections without losing context, which contributes to smooth user experience.

Angular integrates smoothly with RESTful APIs, facilitating effective communication between the frontend and the backend. This integration is essential for uploading MRI images and receiving the analysed data back from the server. Angular’s HttpClient module provides a powerful set of HTTP client tools that make it easy to perform HTTP requests and process responses. This is particularly useful for handling the upload of large image files and retrieving detailed results from the backend.

The application is designed to work seamlessly across different platforms and devices, ensuring that medical professionals can access the application from anywhere, whether in a hospital setting or remotely. The consistent behaviour across different browsers and devices enhances the usability and ensures that all users have the same high-quality experience regardless of their access point.

3.8 Interoperability and Portability

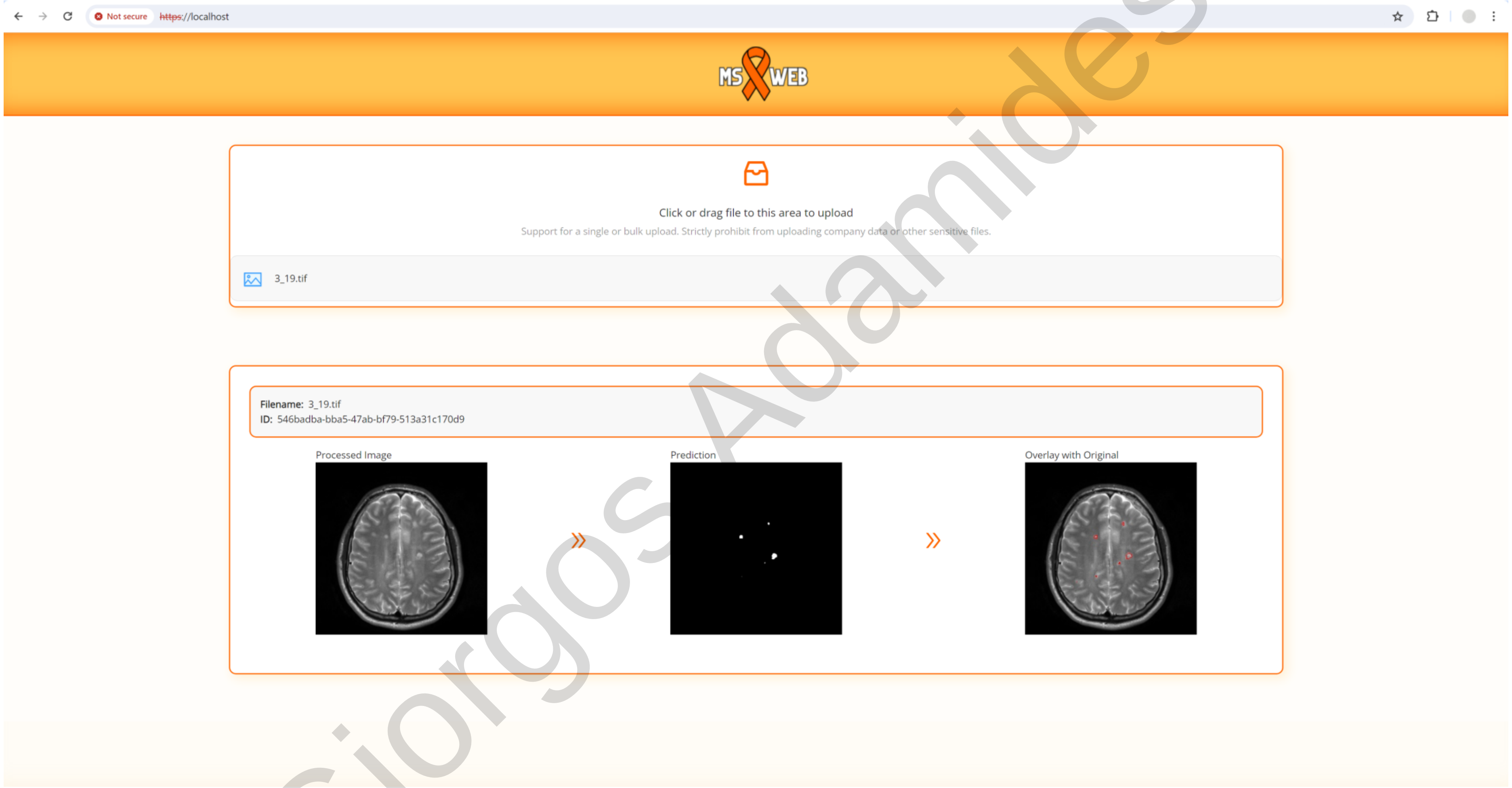
The interoperability and portability are achieved using Docker's [28] containerization technology, which ensures that an application will function consistently across every machine. Because of this uniformity, there is no longer the common problem where programs will function differently in development and production because of differences in operating systems, dependencies, or configurations. The frontend and backend subsystems are containerized using Docker, verifying that the system works as planned throughout the development, testing, and production phases.



Click or drag file to this area to upload

Support for a single or bulk upload. Strictly prohibit from uploading company data or other sensitive files.

Figure 3.14 Framework’s home screen



Not secure https://localhost

Click or drag file to this area to upload

Support for a single or bulk upload. Strictly prohibit from uploading company data or other sensitive files.

3_19.tif

Filename: 3_19.tif
ID: 546badba-bba5-47ab-bf79-513a31c170d9

Figure 3.15 Displaying a prediction using the framework

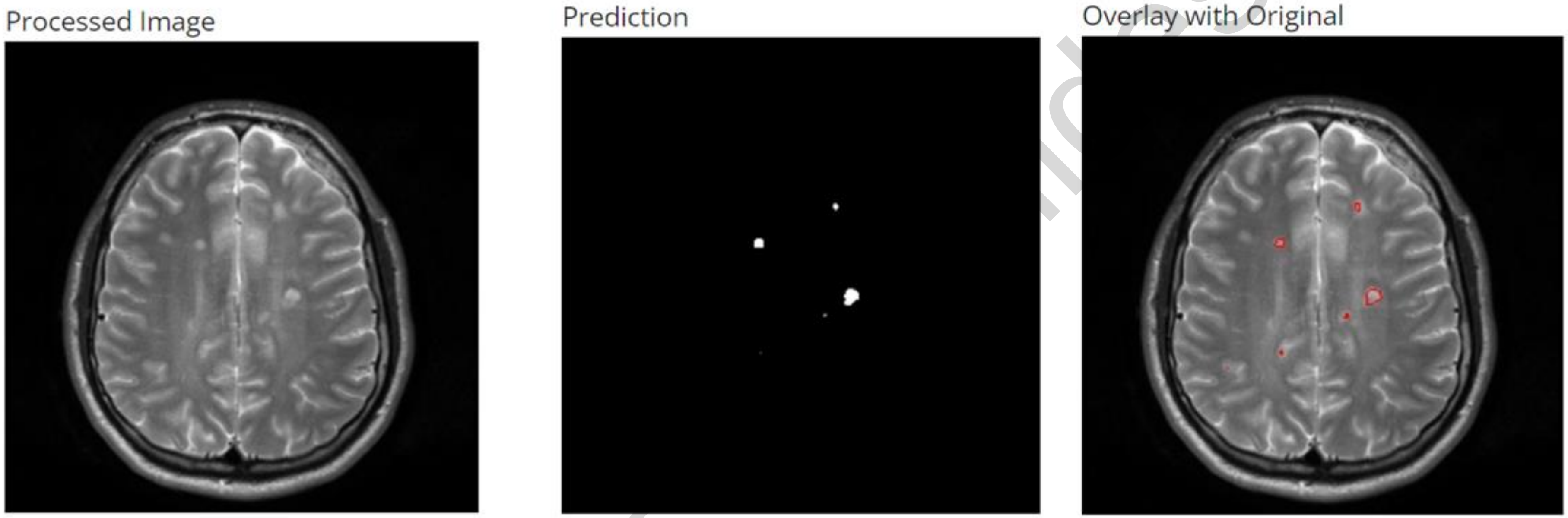


Figure 3.16 Output is the processed image, the mask prediction and last the overlay between the original MRI image and the prediction in red color

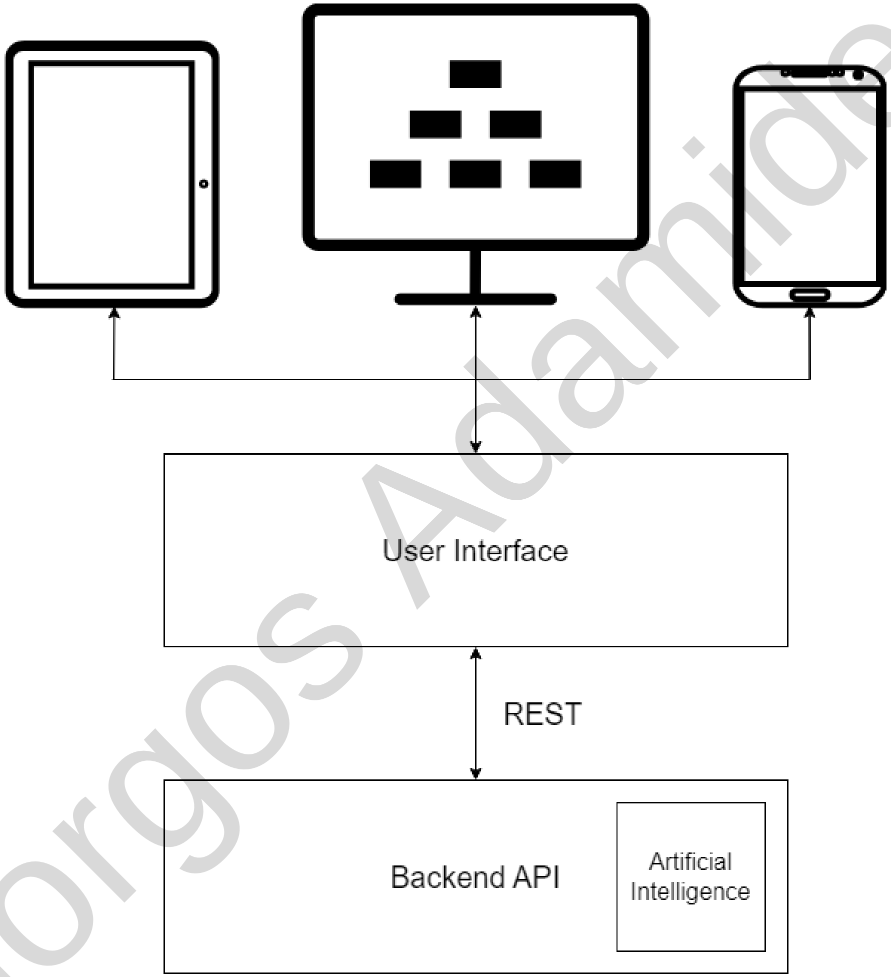


Figure 3.17 Multiple Sclerosis Segmentation Framework architecture diagram

Chapter 4

Results

4.1 Data Normalization

Having now the data available in a normalized form, enables them to be used for training, as it was done. The number of epochs which was set to 100, because that was the point when the stabilization of progress for the validation set was observed.

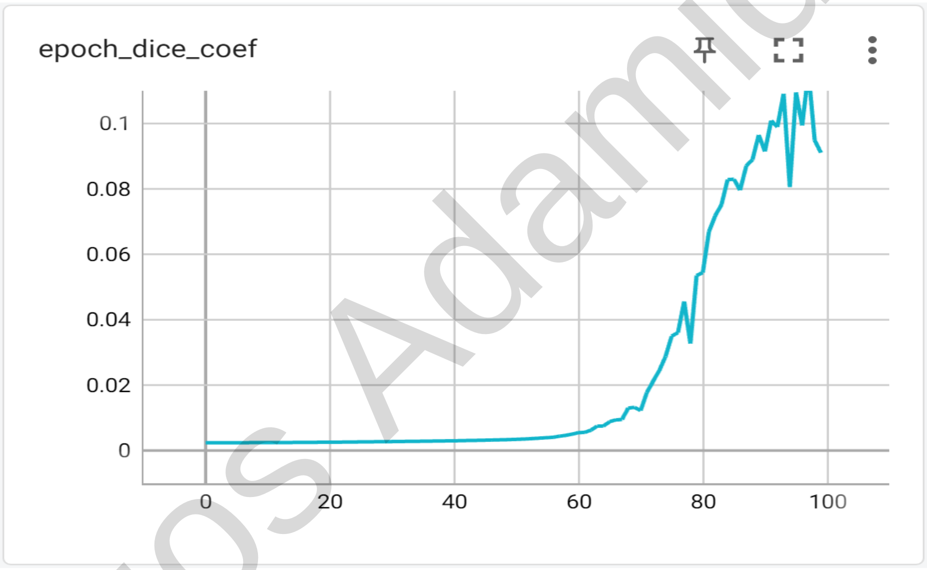


Figure 4.1 Dice Similarity Coefficient Metric Over Seasons with Data Normalization

It is observed that significant progress in prediction occurred after 60 epochs, and began to stabilize at 90 epochs. The highest value achieved during the training period, in the validation set, was a Dice Similarity Coefficient of only 0.11.

4.2 Batch Normalization

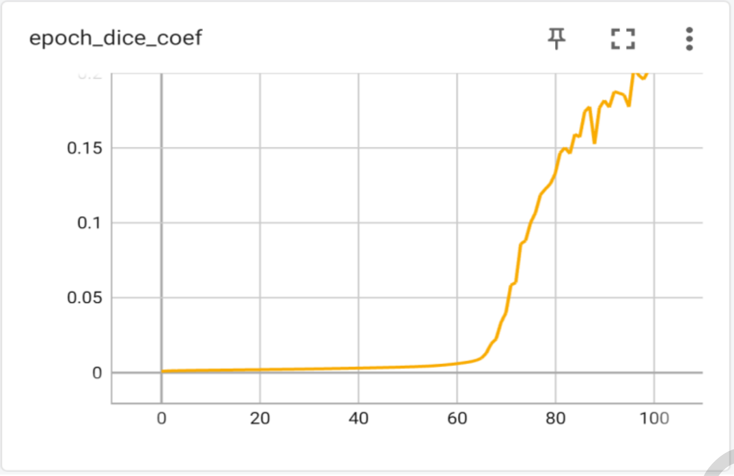


Figure 4.2 Dice Similarity Coefficient Metric over epochs after adding Batch Normalization Layers

Again, it is observed that significant progress in prediction occurred after 60 epochs and began to stabilize at 90 seasons, similar to the previous training effort. However, the key difference is that in this particular attempt, there was a significant improvement in prediction effectiveness, especially considering that the data and parameters remained unchanged and only the addition of these layers led to achieving a Dice Similarity Coefficient of 0.20, almost a double of increase in comparison to the previous attempt.

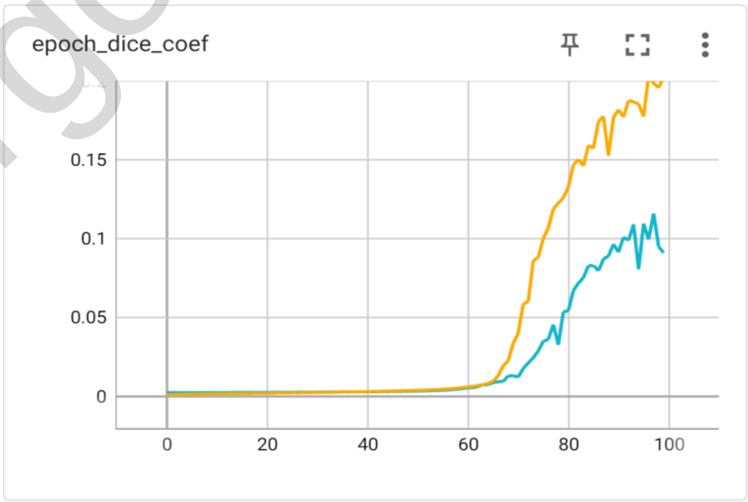


Figure 4.3 Comparison of Dice Similarity Coefficient metric after the addition of Batch Normalization Layers

4.3 Data Augmentation

As seen from the graphical representation, the use of the Data Augmentation technique helped the prediction model in adjusting the training parameters, so as to achieve a Dice Similarity Coefficient of 0.35 for the validation set of results, a result clearly greater than the 0.20 which was achieved without the use of the Data Augmentation technique.

It is also worth noting that the prediction model began to show significant progress as early as 7 seasons, as opposed to the 65 seasons previously needed to begin progress.

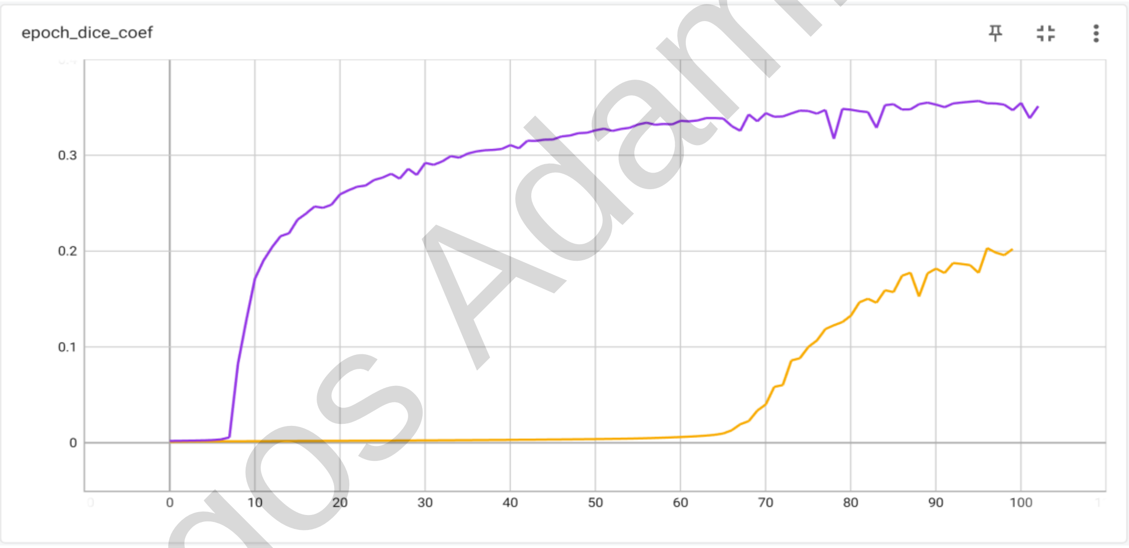


Figure 4.4 Dice Similarity Coefficient in comparison after adding Data Augmentation

4.4 Learning Rate

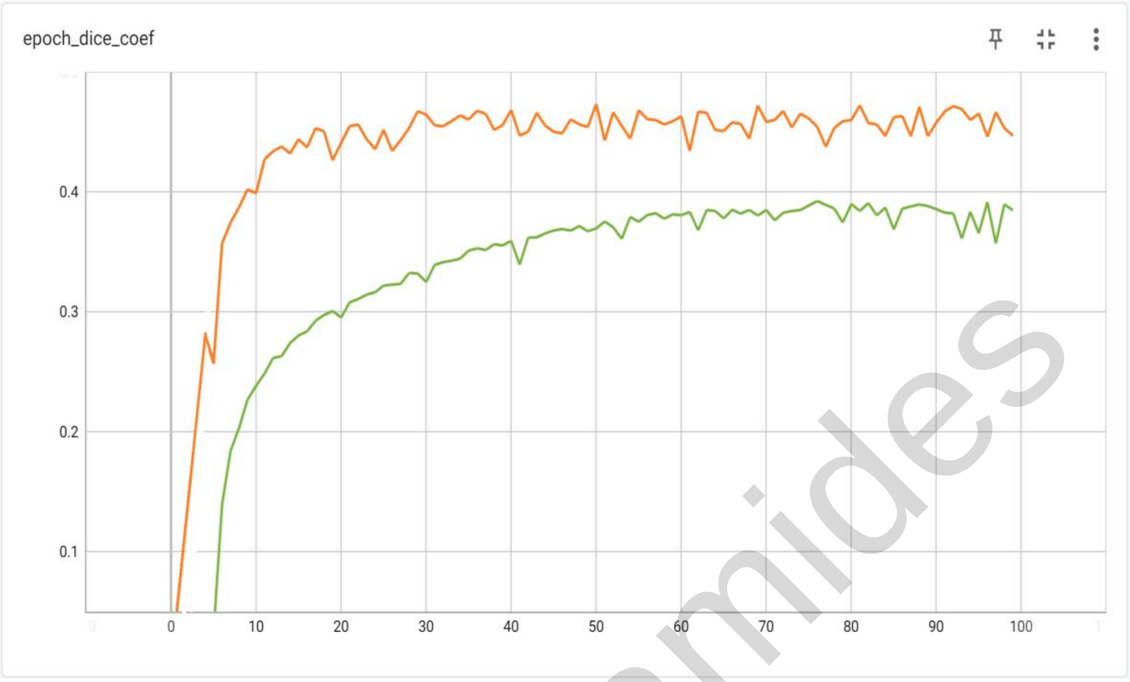


Figure 4.5 Dice Similarity Coefficient metric per Epoch with Learning Rates 1e-6 and 1e-5.

The prediction images from the model with a learning rate of 1e-5 were again as expected, including only black and white pixels and no intermediate value. From the training results, it is concluded that increasing the learning rate helped in detecting new features important for the automatic detection of areas of interest, raising the Dice Similarity Coefficient from 0.39 to 0.47. Furthermore, it is observed that the progress in the model's effectiveness is evident from the first season and stabilizes by the 20th season. Therefore, there was an opportunity for subsequent executions to run for only 20 seasons, thus having a significant impact on the time it took to create the prediction model and, consequently, on the number of tests that could be fulfilled within a specific timeframe.

4.5 Resolution

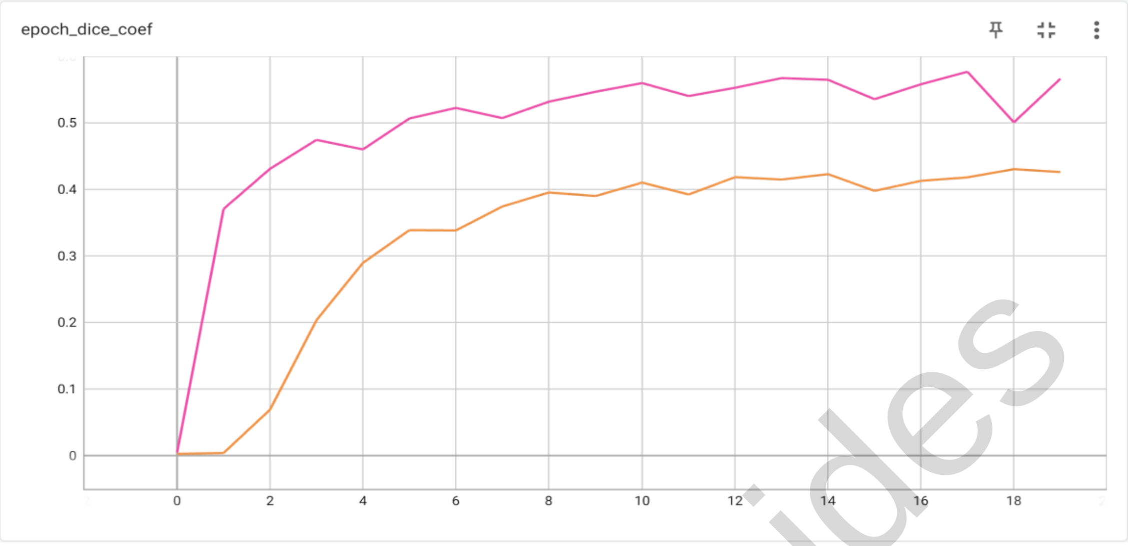


Figure 4.6 Dice Similarity Coefficient Metric per Season Retaining Original Clarity

The final value of the Dice Similarity Coefficient, using the original resolution of the images, for the validation set skyrocketed to 0.58, a fact that was also verified through the test set with a Dice Similarity Coefficient of 0.60.

4.6 Augmentation randomization and Validation readjustment

While some of the previously mentioned experiments did not enhance the model performance on their own or at all, the cumulative application of some of them proved beneficial to a noticeable improvement in the overall performance. Specifically, the stacking of some of these effective strategies, and more specifically the data augmentation randomization and the validation set readjustment, resulted in an increase in the Dice Similarity Coefficient, demonstrating the value of integrating multiple successful techniques in the model's development.

Models	Dice Similarity Coefficient
U-Net	0.69
TransUNet	0.61
Attention U-Net	0.65
ResUNet-a	0.62
U-Net++	0.65

Table 3 Results comparison between the different architectures

4.7 Transfer Learning

Adopting the pretrained weights, the model did not have to learn all features from scratch. Instead, it inherited the patterns these networks had already learned, such as edges, textures, and gradients to the specific task of identifying lesions in brain MRI scans. The primary advantage that it was observed, it was not the improvement in final accuracy, compared to models initialized with random weights, but a significant acceleration in the convergence rate of the training process.

Models	**Random Initialization**		**Transfer Learning**	
	DSC	Epochs	DSC	Epochs
U-Net	0.69	50	0.68	10
TransUNet	0.61	50	0.59	10
Attention U-Net	0.65	50	0.62	10
ResUNet-a	0.62	50	0.56	10
U-Net++	0.65	50	0.61	10

Table 4 Comparison between transfer learning and random initialization

Chapter 5

Discussion

5.1 Discussion

After analysing the steps taken to solve the problem and explaining their effectiveness, the results are now ready to be directly compared with other results from similar research within the same thematic unit. One such study is that of Andreas Georgiou [21], which also aimed at identifying areas indicating the presence of Multiple Sclerosis. The similarities between the methodology of this work and the aforementioned study focus on the architecture of the Convolutional Neural Network, with both approaches using the U-Net architecture and employing the Dice Similarity Coefficient as the evaluation metric.

Despite these similarities, the two approaches show significant differences. The first and major difference is the data on which they relied. The current research had data available from the Electronic Health Laboratory of the University of Cyprus, consisting of 38 patients, each at two time points, with T2 type magnetic resonance images, totalling 1838 available images with their corresponding masks. The comparative approach used part of the ISBI Dataset, utilizing only FLAIR type magnetic resonance images, 5 patients, at 4 different time points, having a total of 3822 available images with their corresponding masks, more than double the data.

Another significant differentiation between the two approaches was the preprocessing applied to the data. Initially, in the current approach, the images were resized to 512x512, while in the comparative approach, they were resized to 256x256. Before being input into the algorithm, as previously described, they were only normalized. In the comparative approach, they underwent significant preprocessing, including image smoothing, skull stripping, and meninges removal.

With this information in mind, the current approach, with just half the number of available data, managed to achieve a DSC of 0.69, while the researcher of the comparative work managed to achieve exact the same DSC of 0.69. Moreover, in the comparative work, a system for manual correction of predictions was implemented, allowing the prediction model's function to be adjusted according to the accuracy of its predictions. With this method, the DSC showed a significant difference, reaching 0.82, with the author noting, "This large difference was expected, as the system we propose has the advantage of knowing some aspects of the morphological arrangement of the brain in patients, where the user of the system applied manual error corrections during previous visits. Thus, it has an awareness of the morphology of the cerebral structure of patients who have been previously evaluated by the system."

This comparison highlights not only the effectiveness of the methods used but also the potential benefits of incorporating manual adjustments to enhance model accuracy, especially in complex medical imaging tasks where understanding the specific patient's morphology can significantly impact the accuracy of predictions.

Based on the literature, variations of 2D U-Net have been used by multiple researchers for Multiple Sclerosis segmentation. Sarica et al. [35] achieved a DSC of 0.67 with a 2D dense residual U-Net on 3D PDw, T1w, T2w, and FLAIR images, which is slightly lower than the DSC in the frame of this thesis, however their method demonstrated better performance compared to other state-of-the-art methods. The dense residual connections likely helped mitigate the vanishing/exploding gradient problem [34], which happens when the data propagated through the network pass through several layers causing the gradients to become extremely small or extremely big and thus the model becomes non-functional, enhancing feature propagation and learning.

On the other hand, Alijamaat et al. [36] used a 2D U-Net based on wavelet pooling on 3D T1w, FLAIR, and 2D PDw images and achieved a DSC of 0.82, a significant increase in performance compared to the previous mentioned studies, emphasizing the effectiveness of wavelet pooling, which assists by providing better multi-resolution analysis and feature extraction. Additionally, a study performed by Rondinella et al. [37] reported a DSC of 0.84 using a 2D U-Net with a C-LSTM unit and attention mechanism on 3D FLAIR images. The methodology followed, includes the use of C-LSTM units and attention mechanisms, which likely contributed to more effective feature learning and integration, particularly for temporal or sequential data as the dataset used, consists of 4 to 5 timepoints for each of the 5 patients. The higher DSC can be attributed either to the lower variability due to the low number of patients or to the addition of C-LSTM units and attention mechanisms, which increase the capabilities of the model and may be beneficial for lesion segmentation.

Apart from the 2D U-Net variations, various studies followed the 3D U-Net implementation, which accepts the brain of the patients in a 3D format and process it as a whole, instead of slicing the brain as the 2D approach. Gamal et al. [38] employed a 3D U-Net with a GAU unit on 2D T1w, T2w, and 3D FLAIR images, achieving a DSC of 0.72 equal to the results of Krishnan et al. [39] who used a 3D Multi-arm U-Net on 3D T1w and 2D T2w, FLAIR images. The inclusion of the GAU unit and the multi-arm architecture, provided a more sophisticated feature extraction and simultaneous multi-scale processing capabilities resulting in slightly higher DSC, indicating the benefits of integrating such advanced approaches into the U-Net architecture. Using two 3D U-Net for brain extraction and lesion segmentation, De Oliveira et al. [40] achieved a DSC of 0.89, which has a significantly higher model performance compared to all the previous methods mentioned, on 3D T1w and FLAIR images. This represents an indication of the advantages of the dual-network approach, allowing for more precise segmentation through sequential processing stages.

U-Net variants include models such as the triplanar 2D U-Net and nnU-Net. Hitziger et al. [41] used a triplanar 2D U-Net, and Barasan et al. [42] utilized the nnU-Net both on 3D FLAIR images, reporting a DSC of 0.44 and 0.51 respectively. The triplanar approach might not be able to fully utilize the 3D context as well as a unified 3D network. The low DSCs might indicate that the 2D U-Net approach produced better segmentation results in this context, perhaps because of its more simplistic architecture, which was better suited for the particular dataset and task at hand.

Chapter 6

Conclusion

5.1 Conclusion

As previously mentioned, the highest result for the Dice Similarity Coefficient metric, after the application of all the mentioned techniques, was 0.60. On a general level, this appears to be a mediocre rate of effectiveness.

However, considering that the masks for the MRI images for detecting Multiple Sclerosis consist of over 99% black pixels, which indicate that these specific areas are not of interest, and taking into account that the Dice Similarity Coefficient does not account for the areas that both the physician and the prediction model identified as negative (True Negative), the importance of this specific metric becomes clear.

In other words, it is evident that there needs to be great coherence between the real areas of interest and those identified by the prediction model to even display a relatively satisfactory overlap rate, since these specific areas constitute less than 1% of the entire image [19].

It is necessary to emphasize that the truth masks, which were created by the physician, contain points from which the outline of the areas of interest is formed. Therefore, these masks should not be considered flawless and may show a slight deviation from reality, so a small margin of error must always be considered.

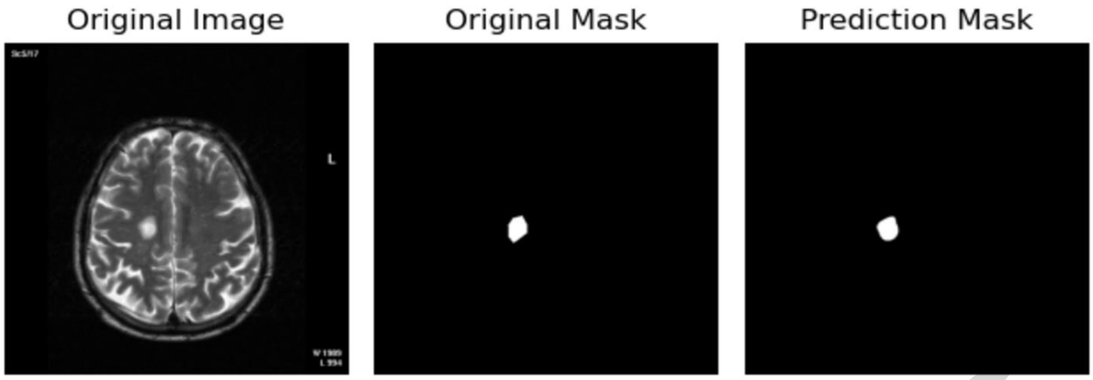


Figure 6.6.1 Example of Comparison Between Physician's Mask and Prediction Model

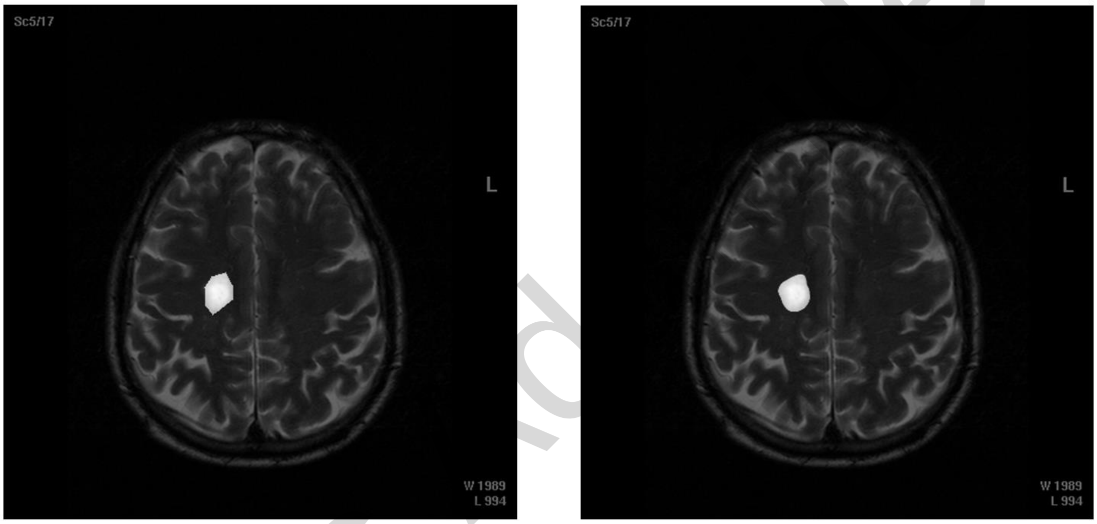


Figure 6.6.2 Example of Comparison Between Physician's Mask and Prediction Model with Overlay

In this particular case, it is observed that the prediction model is more effective than the truth mask, since to create this, the points were joined with straight lines. Therefore, the final shape of the area is a polygon and not perfectly identical to the area it represents.

The metrics arise after comparing the mask created by the physician with that created by the trained prediction model. A simplistic method of comparison is to compare the individual pixels and distribute them accordingly.

	Pixel Count	Percentage
True Positive	575	0.22%
True Negative	261155	99.62%
False Positive	46	0.02%
False Negative	368	0.14%
Dice Similarity Coefficient	0.74	

Table 5 Ground Truth and Prediction metrics

At this point, it is appropriate to also present a table that displays the progress for the training set, the validation set, and the test set to reveal the contribution of each technique.

	Training Set	Validation Set	Test Set
Normalization	0.15	0.11	0.07
Batch Normalization	0.31	0.20	0.17
Data Augmentation	0.50	0.35	0.33
Learning Rate	0.57	0.42	0.43
Resolution	0.71	0.59	0.60
Different Experiments	0.79	0.70	0.69

Table 6 Dice Similarity Coefficient metric per set and technique

5.2 Future Work

In a potential future study, it could be investigated how further modifications, both to the data and the algorithm, might contribute to improving the results and reducing the computational resources required for such a process. Also, as it regards the framework, it would be very beneficial both for the system and the medical professionals using it, to be able to manually correct the predictions through the User Interface and apply continuous training to the model, so it demonstrates progressive improvement with each interaction.

Bibliography

[1] Keiron O’Shea and Ryan Nash, “An Introduction to Convolutional Neural Networks”, arXiv preprint arXiv:1511.08458, 2015

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham, 2015

[3] Chenyi Zeng , Lin Gu, Zhenzhong Liu and Shen Zhao, “Review of Deep Learning Approaches for the Segmentation of Multiple Sclerosis Lesions on Brain MRI”, Frontiers in Neuroinformatics, 14, 610967, 2020

[4] Saad Albawi , Tareq Abed Mohammed and Saad Al-Zawi, “Understanding of a Convolutional Neural Network”, International conference on engineering and technology (ICET), pp. 1-6, , 2017

[5] Ahmet Çinar and Muhammed Yildirim, “Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture”, Med. Hypotheses, 139, 109684, 2020

[6] Daniel Goldberg-Zimring, Ion-Florin Talos, Jui G. Bhagwat, Steven J. Haker, Peter M. Black and Kelly H. Zou, “Statistical Validation of Brain Tumor Shape Approximation via Spherical Harmonics for Image-Guided Neurosurgery”, Academic Radiology, vol. 12, pp. 459-466, 2005

[7] Juul van Boxtel, Vincent Vousten, Josien Pluim and Nastaran Mohammadian Rad, “Hybrid Deep Neural Network for Brachial Plexus Nerve Segmentation in Ultrasound Images”, 29th European Signal Processing Conference (EUSIPCO), pp. 1246-1250, 2021

[8] Ultrasound Nerve Segmentation Competition : https://www.kaggle.com/competitions/ultrasound-nerve-segmentation/data

[9] MRI Lesion Segmentation in Multiple Sclerosis Database: http://www.ehealthlab.cs.ucy.ac.cy/index.php/facilities/32-software/218-datasets

[10] C.P. Loizou, V. Murray, M.S. Pattichis, I. Seimenis, M. Pantziaris and C.S. Pattichis,
"Multi-scale amplitude modulation-frequency modulation (AM-FM) texture analysis of multiple sclerosis in brain MRI images," IEEE Trans. Inform. Tech. Biomed., vol. 15, no. 1, pp. 119-129, 2011.

[11] C.P. Loizou, E.C. Kyriacou, I. Seimenis, M. Pantziaris, S. Petroudi, M. Karaolis and C.S. Pattichis,
"Brain white matter lesion classification in multiple sclerosis subjects for the prognosis of future disability,"
Intelligent Decision Technologies Journal (IDT), vol. 7, pp. 3-10, 2013.

[12] C.P. Loizou, S. Petroudi, I. Seimenis, M. Pantziaris and C.S. Pattichis,
"Quantitative texture analysis of brain white matter lesions derived from T2-weighted MR images in MS patients with clinically isolated syndrome",
J. Neuroradiol., acepted.

[13] C.P. Loizou, M. Pantziaris, C.S. Pattichis and I. Seimenis,
"Brain MRI Image normalization in texture analysis of multiple sclerosis",
J. Biomed. Graph. & Comput., vol. 3, no.1, pp. 20-34, 2013.

[14] Sergey Ioffe and Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", PMLR , vol. 37, pp. 448-456, 2015

[15] Connor Shorten and Taghi M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning", J. Big Data, vol. 6, pp. 1-48, 2019

[16] Luke Taylor and Geoff Nitschke, “Improving Deep Learning with Generic Data Augmentation”, IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1542-1547, 2018

[17] Junghwan Cho, Kyewook Lee, Ellie Shin, Garry Choy, and Synho Do, “HOW MUCH DATA IS NEEDED TO TRAIN A MEDICAL IMAGE DEEP LEARNING SYSTEM TO ACHIEVE NECES - SARY HIGH ACCURACY ?”, arXiv preprint arXiv:1511.06348, 2015

[18] Fabian Isensee, Jens Petersen, Andre Klein, David Zimmerer, Paul F. Jaeger, Simon Kohl, Jakob Wasserthal, Gregor K¨ohler, Tobias Norajitra, Sebastian Wirkert, and Klaus H. Maier-Hein, “nnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation”, arXiv preprint arXiv:1809.10486, 2018

[19] Carl F. Sabottke and Bradley M. Spieler, “The Effect of Image Resolution on Deep Learning in Radiography”, Radiology: Artificial Intelligence, vol. 2, e190015., 2020

[20] Hang Zhang, Jinwei Zhang, Chao Li, Elizabeth M. Sweeney, Pascal Spincemaille, Thanh D. Nguyen, Susan A. Gauthier and Yi Wang, “ALL-Net: Anatomical information lesion-wise loss function integrated into neural network for multiple sclerosis lesion segmentation”, NeuroImage: Clinical, vol.32, 102854, 2021

[21] Andreas Georgiou, “An Adaptive Semi-Automated Integrated System for Multiple Sclerosis Lesions Segmentation in Longitudinal MRI Scans Based on a Convolutional Neural Network”,In: 19th International Conference of Computation Analysis of Images Patterns CAIP, pp.256-265, 2021

[22] Bousias Alexakis, E., & Armenakis, C. (2020). Evaluation of UNet and UNet++ architectures in high resolution image change detection applications. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 43, 1507-1514.

[23] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., ... & Rueckert, D. (2018). Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999.

[24] Diakogiannis, F. I., Waldner, F., Caccetta, P., & Wu, C. (2020). ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. ISPRS Journal of Photogrammetry and Remote Sensing, 162, 94-114.

[25] Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., ... & Zhou, Y. (2021). Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306.

[26] Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. Journal of Big data, 3, 1-40.

[27] Huh, M., Agrawal, P., & Efros, A. A. (2016). What makes ImageNet good for transfer learning?. arXiv preprint arXiv:1608.08614.

[28] Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. Linux j, 239(2), 2.

[29] Masse, M. (2011). REST API design rulebook: designing consistent RESTful web service interfaces. " O'Reilly Media, Inc.".

[30] Panayides, A. S., Amini, A., Filipovic, N. D., Sharma, A., Tsaftaris, S. A., Young, A., ... & Pattichis, C. S. (2020). AI in medical imaging informatics: current challenges and future directions. IEEE journal of biomedical and health informatics, 24(7), 1837-1857.

[31] Holzinger, A., Biemann, C., Pattichis, C. S., & Kell, D. B. (2017). What do we need to build explainable AI systems for the medical domain?. arXiv preprint arXiv:1712.09923.

[32] Nicolaou, A., Kakas, A., Pattichis, C. S., Pattichis, M. S., Fotso, K., Loizou, C. P., & Pantzaris, M. (2023, June). An explainable artificial intelligence model in the assessment of brain MRI lesions in multiple sclerosis using amplitude modulation–frequency modulation multi-scale feature sets. In 2023 24th International Conference on Digital Signal Processing (DSP) (pp. 1-4). IEEE.

[33] Nicolaou, A., Loizou, C. P., Pantzaris, M., Kakas, A., & Pattichis, C. S. (2021, September). Rule extraction in the assessment of brain MRI lesions in multiple sclerosis: Preliminary findings. In International Conference on Computer Analysis of Images and Patterns (pp. 277-286). Cham: Springer International Publishing.

[34] Hanin, B. (2018). Which neural net architectures give rise to exploding and vanishing gradients?. Advances in neural information processing systems, 31.

[35] Sarica, B., Seker, D. Z., & Bayram, B. (2023). A dense residual U-net for multiple sclerosis lesions segmentation from multi-sequence 3D MR images. International Journal of Medical Informatics, 170, 104965.

[36] Alijamaat, A., NikravanShalmani, A., & Bayat, P. (2021). Multiple sclerosis lesion segmentation from brain MRI using U-Net based on wavelet pooling. International journal of computer assisted radiology and surgery, 16(9), 1459-1467.

[37] Rondinella, A., Crispino, E., Guarnera, F., Giudice, O., Ortis, A., Russo, G., ... & Battiato, S. (2023). Boosting multiple sclerosis lesion segmentation through attention mechanism. Computers in Biology and Medicine, 161, 107021.

[38] Gamal, R., Barka, H., & Hadhoud, M. (2023). GAU U-Net for multiple sclerosis segmentation. Alexandria Engineering Journal, 73, 625-634.

[39] Krishnan, A. P., Song, Z., Clayton, D., Jia, X., de Crespigny, A., & Carano, R. A. (2023). Multi-arm u-net with dense input and skip connectivity for t2 lesion segmentation in clinical trials of multiple sclerosis. Scientific Reports, 13(1), 4102.

[40] de Oliveira, M., Piacenti-Silva, M., da Rocha, F. C. G., Santos, J. M., Cardoso, J. D. S., & Lisboa-Filho, P. N. (2022). Lesion volume quantification using two convolutional neural networks in MRIs of multiple sclerosis patients. Diagnostics, 12(2), 230.

[41] Hitziger, S., Ling, W. X., Fritz, T., D'Albis, T., Lemke, A., & Grilo, J. (2022). Triplanar U-Net with lesion-wise voting for the segmentation of new lesions on longitudinal MRI studies. Frontiers in Neuroscience, 16, 964250.

[42] Basaran, B. D., Matthews, P. M., & Bai, W. (2022). New lesion segmentation for multiple sclerosis brain images with imaging and lesion-aware augmentation. Frontiers in neuroscience, 16, 1007453.