Automatic Segmentation and Learning of the Distribution of White Matter Hyperintensities from 3D Neuro MRI

A Project / Dissertation as a Course requirement for
Master of Science
(Data Science and Computing)

Mrinal Kanti Saha
21234

SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING
(Deemed to be University)

Department of Mathematics and Computer Science
Muddenahalli Campus

April 2023

SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING
(Deemed to be University)

Dept. of **Mathematics & Computer Science**
Muddenahalli Campus

CERTIFICATE

This is to certify that this Project / Dissertation titled **Automatic Segmentation and Learning of the Distribution of White Matter Hyperintensities from 3D Neuro MRI** submitted by **Mrinal Kanti Saha**, Regd No. **21234**, Department of Mathematics and Computer Science, Muddenahalli Campus is a bonafide record of the original work done under our supervision as a Course requirement for the Degree of Master of Science Data Science and Computing.

Dr. N Uday Kiran	Dr. Sampath Lonka
Co-Superviser	Supervisor

Place: Muddenahalli
Date: April 26, 2023

Countersigned by
Dr. (Ms.) Y Lakshmi Naidu
Head of the Department

DECLARATION

The Project / Dissertation titled **Automatic Segmentation and Learning of the Distribution of White Matter Hyperintensities from 3D Neuro MRI** was carried out by me under the supervision of Dr. Sampath Lonka, Department of Mathematics and Computer Science, Muddenahalli Campus and Dr. N Uday Kiran, Department of Mathematics and Computer Science, Prashanti Nilayam Campus as a Course requirement for the Degree of Masters of Science in Data Science and Computing and has not formed the basis for the award of any degree, diploma or any other such title by this or any other University.

Place: Muddenahalli
Date: April 26, 2023

_____
Mrinal Kanti Saha, 21234
M.Sc. (Data Science and Computing)

Acknowledgements

First and foremost, I would like to express my gratitude to our Swami, who is the reason I am here, studying in his reverential institute. It is He who gave me the opportunity to work on this project, and guided me along every step.

I would like to thank Sri. PVSS Prakash for the workshop on Deep Learning that gave us a basic understanding of CNNs and acted as a headstart on the project.

I would also like to thank Sri. Hirak Doshi, Research Scholar, SSSIHL, Prashanti Nilayam campus for his help with regard to 3D medical image data.

I would like to extend my gratitude to my classmates and my teachers who were instrumental in providing feedback and giving me encouragement in tough times.

Finally, I would express my gratitude to my supervisors, Dr. Sampath Lonka and Dr. N. Uday Kiran, for their guidance, patience, and time invested in my project. They have inspired me to think critically and push myself to achieve my best work.

Abstract

White Matter Hyperintensities (WMH) are areas of increased signal intensities that appear bright white patches on brain imaging techniques, particularly, the Fluid Attenuated Inversion Recovery (FLAIR) Magnetic Resonance Images (MRI). Previously, WMH were perceived as a normal consequence of aging. Studies, however, reveal that WMH are associated with a wide range of neurological and cognitive disorders, the most common being dementia, cognitive impairment, and strokes.

Automatic segmentation and quantification of WMH is an important diagnostic step in this case. Deep learning has had a significant impact on the medical image community, revolutionizing the field and enabling novel approaches to diagnosis, prognosis, and research. The key benefit of deep learning is that, given large amounts of medical image data, it often outperforms traditional machine learning-based methods and human experts in image classification and segmentation tasks.

In this work, we focus on the segmentation of WMH on data obtained from Sri Sathya Sai Institute of Higher Medical Sciences, taking the load off the shoulders of radiologists, speeding up the process, and standardizing it as well. Also, it sheds light on the problem of inter-observer variability in the process of manual segmentation. In addition, this project employs generative modeling to learn the distribution of WMH from the FLAIR MRI images

Table of Contents

Chapter 1: White Matter Hyperintensities - Diagnosis

1.1. Introduction

White Matter Hyperintensities (WMH) are abnormalities in the white matter of the brain detected using brain imaging techniques, such as Magnetic Resonance Imaging (MRI). WMH are usually seen on T2-FLAIR images as hyperintense blobs. Early detection of WMH is critical as it can lead to a prognosis of potential future cognitive decline.

1.2. The Human Brain

The human brain is composed of two main types of tissue: gray matter and white matter.
Gray matter, actually pinkish-gray in color, is made up of neuronal cell bodies, dendrites, and axon terminals. This tissue is abundant in the cerebrum, cerebellum, and brainstem. The white matter is composed of bundles of axons. These axons are coated with myelin, a mixture of protein and lipids, that helps conduct nerve signals and protects the axons. It is due to the myelin that the white matter gets its 'white' appearance.

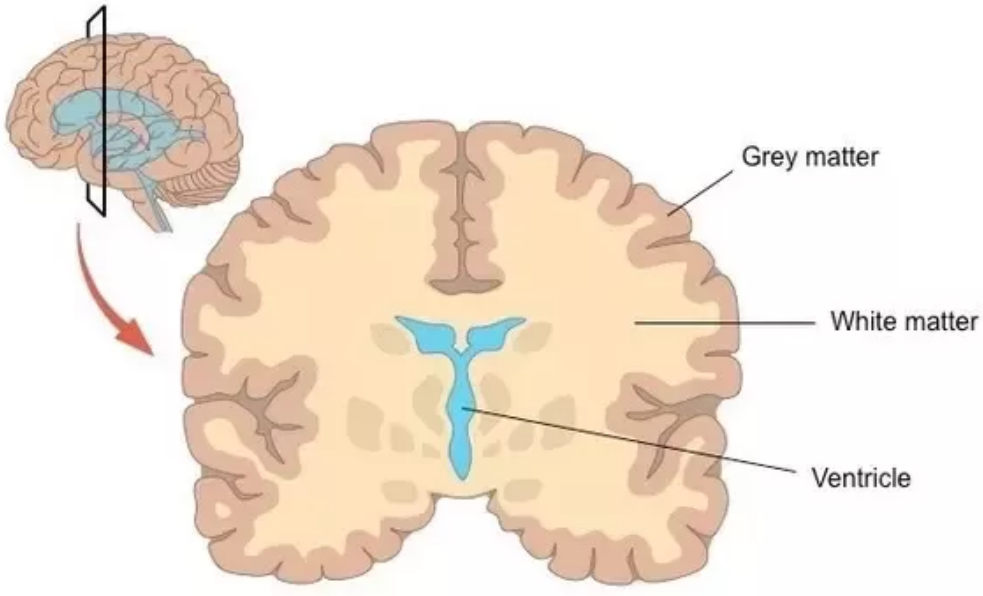

Figure 1.1 - The human brain in terms of grey and white matter (source)

1.4. Clinical Significance

White matter hyperintensities (WMH) are extensively studied neuroimaging biomarkers of cerebral small vessel diseases (CSVD) and are correlated to strokes, dementia, and cognitive decline. Some studies confirm the relationship between cerebrovascular diseases and WMH. Also, motor and cognitive deficits are associated with the volume and location of the WMH. Periventricular and subcortical WMH are associated with cognitive and motor impairment respectively. WMH proliferates with an increase in age.

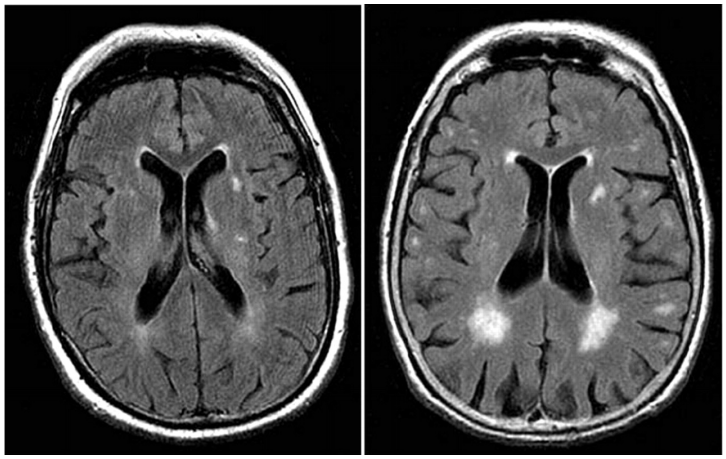

Figure 1.2 - The White Matter Hyperintensities in T2-FLAIR (source)

The automatic segmentation of WMH could be introduced in clinical use to support diagnosis, prognosis, and treatment monitoring of dementia, CSVD, frontotemporal dementia, etc. Patients with extensive WMH are at increased risk of strokes. Therefore, WMH is identified as an important part of investigative diagnosis and their discovery should lead to further screening of other factors of dementia and strokes, making WMH an intermediate biomarker.
As the WMH are heterogeneous in shape, size, location, and appearance, manual marking of WMH by radiologists suffers from inter- and intra-observer variability, apart from being a time-consuming process. Also, the marking of WMH and other brain lesions is a tough process that requires radiologists to strictly abide by the STRIVE (STandards for ReportIng Vascular changes on nEuroimaging). Automatic segmentation methods would help standardize the process of segmentation while achieving comparable results. Learning of the distribution would open up new directions to the research of WMH and its effects.
“Examining this relationship is of importance as WMH markers may aid in future patient selection for preventive treatment to ameliorate the risk of CSVD-related death and ischemic stroke.” - Rashid Ghaznawi [1].

Chapter 2: Neuro Imaging - MRI

2.1. Introduction

Magnetic Resonance Imaging (MRI) is a non-invasive medical imaging technique that employs strong magnetic fields and radio waves to visualize the internal organs of the body, such as the brain, muscles, etc. In the context of neuroimaging, MRI provides highly detailed images of the brain's internal structure. It is generally used to visualize the different brain tissues, such as white matter, gray matter, and cerebrospinal fluid (CSF). Hence, it is one of the preferred modalities for the detection and diagnosis of abnormalities such as lesions, tumors, strokes, and neurodegenerative diseases such as Alzheimer's and Parkinson's.

2.2. The MRI Process

Unlike X-rays and Computed Tomography (CT), MRI is free from the harmful effects of ionizing radiation, as it uses powerful magnets instead. The patient is made to lie down inside the MRI machine, with large tube-shaped magnets.

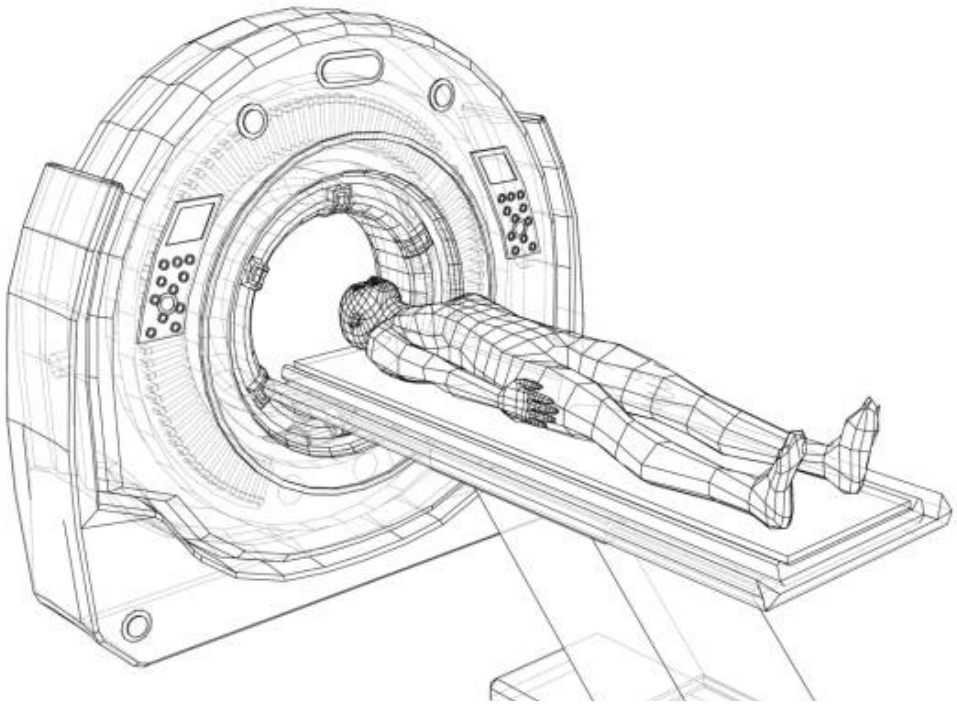

Figure 2.1 - The MRI Machine (source)

The MRI machine generates a strong magnetic field (usually of field strengths 3T or 1.5T), that aligns the protons in our body. The machine emits radio waves that are absorbed by these protons, causing the excitation of the protons. As the excited protons return to their alignment, they emit the stored energy which is detected as signals by the MRI machine. These signals are in turn converted into an image using a computer and specialized imaging protocols.
The strength of the magnetic field determines the quality of the image. Higher field strengths lead to better quality imaging, but may also lead to an increased risk of artifacts and can be harmful to the patients. In addition, the strength and the frequency of the radio waves can be controlled to ensure that they are absorbed by certain tissues of concern, allowing for selective imaging of tissues.

2.3. Types of Neuro MRI

There are many different types of MRI, but in our context, we will only be considering structural and functional MRI.

2.3.1. Structural MRI (sMRI)

This is the most common type of MRI, used to visualize the structure of the brain and detect anomalies. Structural MRI can be acquired using different image capture techniques. Most frequent and popular would be the T1-weighted, T2-weighted, and T2-FLAIR scans.

1. T1-weighted images provide high-contrast visualization of anatomical structures, such as white matter and fat. T1 Gd (Gadolinium) is another type of T1-weighted scan that involves the injection of a contrast agent called Gadolinium. This enhances the visibility of inflammations and tumors on the T1-weighted images.
2. T2-weighted images provide information about the water content of the tissues. Tissues with high water content, like CSF and edema, appear bright while tissues with low water content appear dark, like white matter and bone. They are mainly useful to detect inflammation and water-filled regions, such as the ventricles of the brain or fluid-filled cysts.
3. T2-FLAIR (Fluid Attenuated Inversion Recovery) images are actually a combination of T2-weighted and inversion recovery techniques that selectively suppress the signals coming from fluids in the brain while keeping the abnormalities bright. Hence, CSF appears dark in FLAIR. It is very sensitive to pathologies and is helpful for the detection

of small lesions that usually do not stand out on the other types of imaging. Also, FLAIR images are the preferred modality in the case of WMH.

All the modalities can be used together for the detection of certain conditions. In our case, we will be using T1-weighted and T2-FLAIR for the segmentation of WMH.

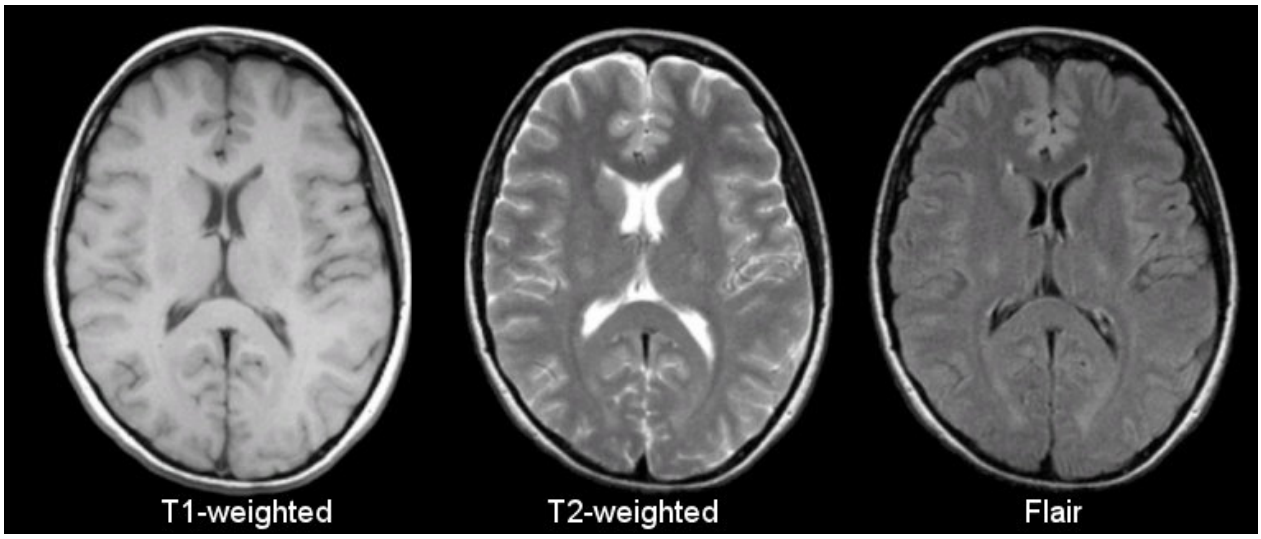

Figure 2.2 - Comparison of T1-weighted, T2-weighted, and T2-FLAIR images (source)

Table 2.1 - Comparison of shades of different brain tissues viewed on T1-weighted, T2-weighted, and T2-FLAIR images (source)

Tissue	**T1-weighted**	**T2-weighted**	**T2-FLAIR**
White Matter	Light	Dark Gray	Dark Gray
Gray Matter (cortex)	Gray	Light Gray	Light Gray
CSF	Dark	Bright	Dark
Inflammation	Dark	Bright	Bright

2.3.2 Functional MRI (fMRI)

This type of MRI is used to measure the brain's functioning, by analyzing the change in blood oxygenation levels in different parts of the brain, using BOLD (Blood Oxygen Level Dependent) signals. Under fMRI, there are two main categories.

First, resting state fMRI (rs-fMRI) wherein the patient is asked to not engage in any cognitive activity while undergoing the scan. This aims to create a functional connectivity map of the brain, without any task-related influences.
Second, task-based fMRI requires the patient under scan to perform some behavioral activity as per stimuli or instructions. This, again, aims to create a functional connectivity map of the brain that corresponds to a particular cognitive process.

2.5. MRI Images

The MRI machine produces a 3D image from the signals, using a computer and specialized imaging techniques. A 3D image should not be mistaken to be a video where the height, width, and time make up the three dimensions. In the medical context, the three dimensions stand for height, width, and depth. Instead of having only a flat 2D image, the 3D image consists of a stack of such 2D images, also called slices, which are parallel and equally spaced along the depth axis.

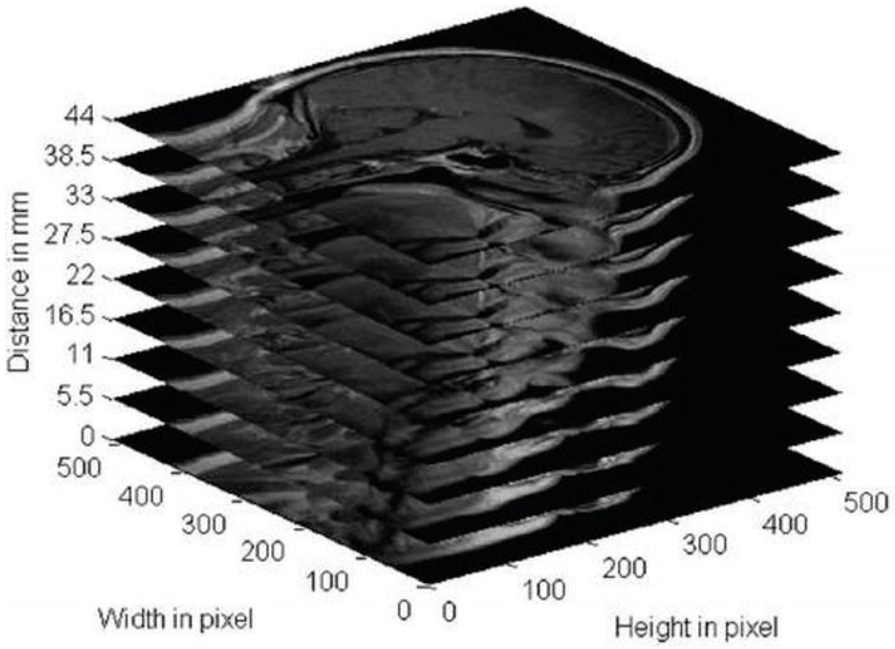

Figure 2.3 - 3D MR image showing the sagittal view of the brain (source)

This kind of image is also obtained from CT and Ultrasound. As they cover all three spatial dimensions, they are extensively used to study and analyze the internal structures of the body. For example, a 3D brain scan can be utilized to locate and determine the size and extent of a tumor.

The slices can be of varying thickness in the sense that it averages signals coming from a range of depths into a flat 2D image. This however leads to poor representation of anatomical features. MRI scans are typically classified into thin-slices and thick-slice scans if the slice thickness is more than 3 mm and less than 3 mm respectively. Thin-slice scans are preferred as they contain more precise details lost in thick-slice due to averaging.
Slice thickness should not be confused with slice interval or interslice gap, which represents the distance between two consecutive slices across the depth dimension (in mm).
The concepts of slice thickness and slice interval can be well summarized in the figure below.

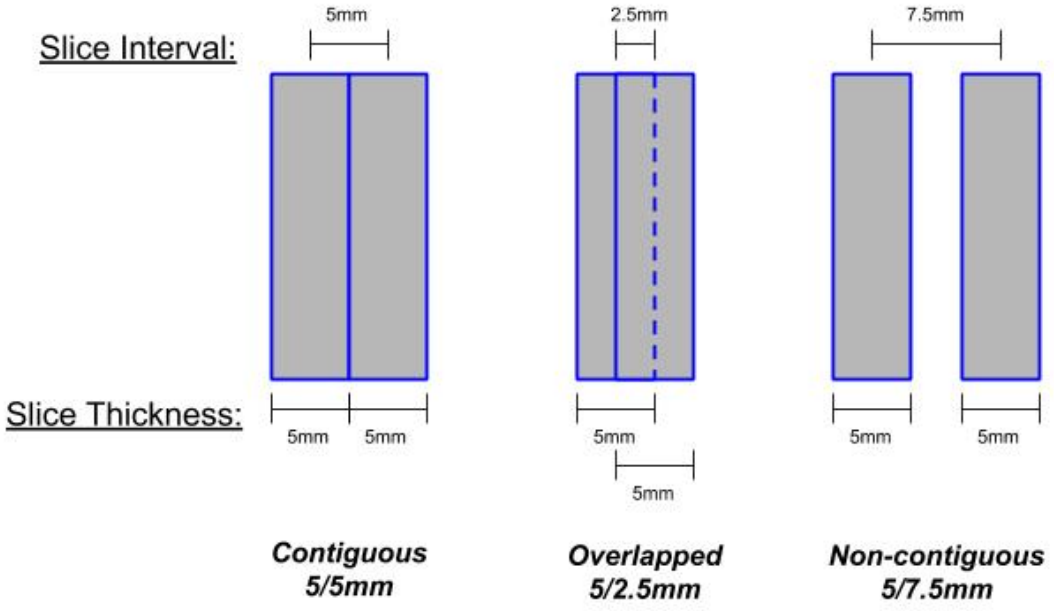

Figure 2.4 - Understanding slice thickness and slice interval (source)

2.5. Popular Data Formats

Specific data formats are required to store a 3D medical image, to account for the 3D image factor, and also the metadata that is usually associated with medical images such as patient information, scanning parameters, and image metadata. The data formats for medical image images need to be able to handle such information, at the same time, be compatible with the scanning hardware, storage devices, and various software programs. The two most popular data formats are DICOM (Digital Imaging and Communications in Medicine) and NIfTI (Neuroimaging Informatics Technology Initiative).

2.5.1 DICOM

DICOM is the de facto standard for storing and transferring medical images, like MRI, CT, and X-rays. Additionally, it is not limited to 3D images, but also provisions for 2D and 4D images. It has been implemented in almost every radiological device and PACS (Picture Archiving and Communication Systems). Most vendors pre-bake their devices with the DICOM standards, to ensure interoperability with devices and software. DICOM data objects contain a number of attributes, one among them being the image pixel array. Other attributes can be patient-related, study-related, or scanner-related attributes. DICOM converts the 3D image into a series of 2D slices and stores it as a separate file with all the metadata for every single file, also including the position and orientation of the slice in the 3D volume. DICOM files (.dcm), therefore are stored as multiple files in a directory for a single 3D image.

```
Dataset.file_meta -------------------------------
(0002, 0000) File Meta Information Group Length  UL: 196
(0002, 0001) File Meta Information Version       OB: b'\x00\x01'
(0002, 0002) Media Storage SOP Class UID         UI: CT Image Storage
(0002, 0003) Media Storage SOP Instance UID      UI: 1.3.6.1.4.1.14519.5.2.1.7085.2626.214140401149739
(0002, 0010) Transfer Syntax UID                 UI: Explicit VR Little Endian
(0002, 0012) Implementation Class UID            UI: 1.2.40.0.13.1.1.1
(0002, 0013) Implementation Version Name         SH: 'dcm4che-1.4.35'
-------------------------------------------------
(0008, 0005) Specific Character Set              CS: 'ISO_IR 100'
(0008, 0008) Image Type                          CS: ['ORIGINAL', 'PRIMARY', 'AXIAL', 'CT_SOM5 SPI']
(0008, 0016) SOP Class UID                       UI: CT Image Storage
(0008, 0018) SOP Instance UID                    UI: 1.3.6.1.4.1.14519.5.2.1.7085.2626.214140401149739
(0008, 0020) Study Date                          DA: '20100227'
(0008, 0021) Series Date                         DA: '20100227'
(0008, 0022) Acquisition Date                    DA: '20100227'
(0008, 0023) Content Date                        DA: '20100227'
(0008, 0030) Study Time                          TM: '161937.171'
(0008, 0031) Series Time                         TM: '162536.14'
(0008, 0032) Acquisition Time                    TM: '162203.028699'
(0008, 0033) Content Time                        TM: '162203.028699'
(0008, 0050) Accession Number                    SH: '1598252606449858'
(0008, 0060) Modality                            CS: 'CT'
(0008, 0070) Manufacturer                        LO: 'SIEMENS'
(0008, 0090) Referring Physician's Name          PN: ''
(0008, 1030) Study Description                   LO: 'CT CHEST W IV CONTRAST'
(0008, 103e) Series Description                  LO: 'LUNG 3.0  B70f'
(0008, 1090) Manufacturer's Model Name           LO: 'Sensation 16'
```

Figure 2.5 - List of a few attributes of a DICOM file as viewed using pydicom (source)

2.5.2. NIfTI

NIfTI is a popular format specifically meant for storing neuroimaging data. Like DICOM, this too can handle 2D and 4D images. NIfTI files consist of a header file and the actual image array which can be combined into a single file (.nii). The header contains information similar to those in DICOM format. NIfTI stores the 3D images together as a single file, and not as separate slices. Hence, prevent the replication of metadata across slices.

Chapter 3: Literature Review

3.1. Introduction

The chapter summarizes the research that was conducted in the project, including the focus on segmentation, generative modeling, and deep learning, as well as the potential applications of deep learning methods to these problems. It provides a clear overview of the topics covered as part of the project.

3.2. Segmentation Models

Segmentation is often viewed as a simple classification problem, where every pixel in the image is assigned a class label, also known as dense prediction.
Ciresan et al. [25] developed a deep convolutional neural network along the same lines that won them the EM segmentation challenge at ISBI 2012. It used a square area (patch) of width w centered on a pixel p to predict its label, in the form of probabilities of whether the pixel belongs to the membrane class or not, w being an odd number. Data augmentation techniques such as mirroring and random 90° rotations were used to increase the size of the dataset. Also, there were multiple architectures in use, the final result was the average of the probabilities of the outputs.

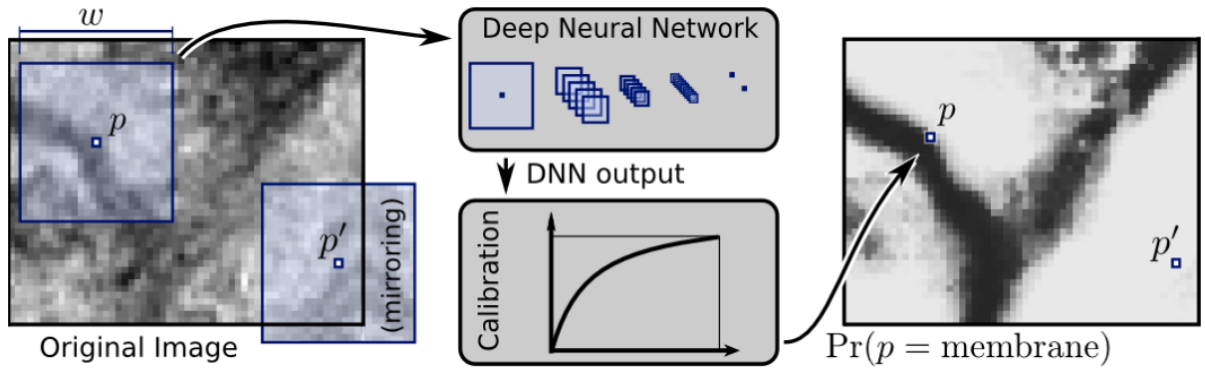

Figure 3.1 - Ciresan et al. approach to pixel classifier [25]

But the CNN-based classifiers failed in the case of arbitrary-sized inputs, due to the presence of dense or fully connected layers in them. Long et al. [2] built the Fully Convolutional Network (FCN) which allowed deep neural networks to take in inputs of arbitrary sizes. Also, they took ILSVRC classification networks like AlexNet, LeNet and turned them into FCN by

replacing the dense layers with convolutions that have a kernel size equal to the input size at that dense layer. Also, the authors exploited the learned networks for the task of semantic segmentation, by upsampling the final layer (of the converted FCN) output using deconvolution or transposed convolution and a pixel-wise loss.

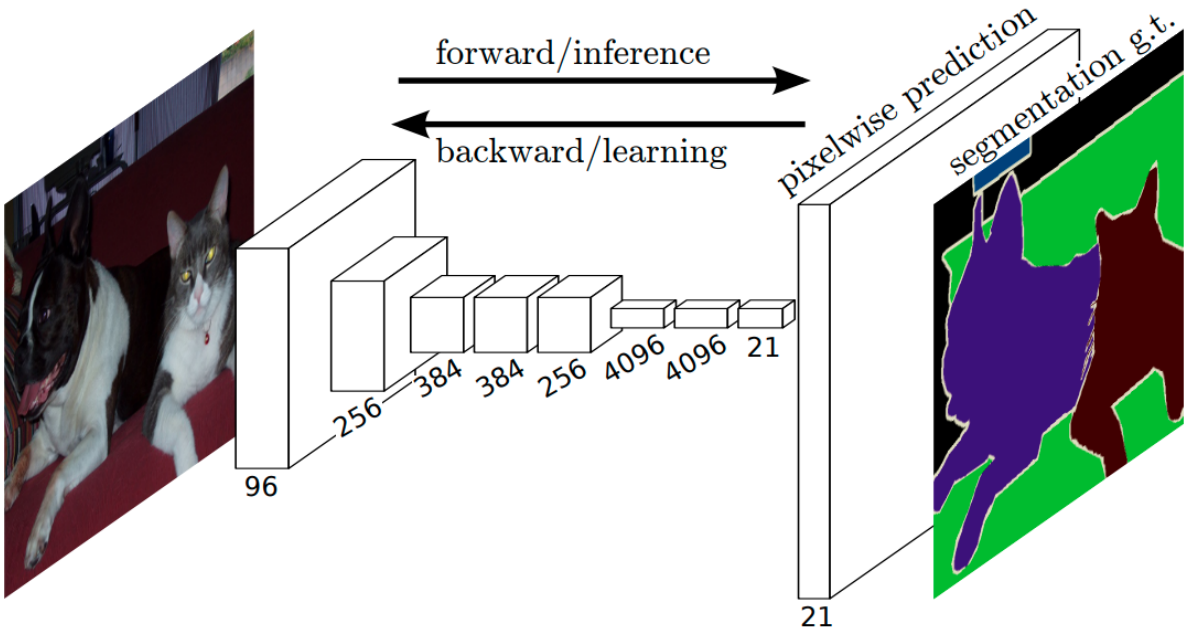

Figure 3.2 - Fully Convolutional Networks architecture [2]

Ronneberger et al. [3], citing the redundancy of using overlapping patches in the approach of Ciresan et al. developed the U-Net architecture, an FCN, which has become the staple segmentation architecture. The U-Net architecture is a symmetric encoder-decoder architecture, shaped as a 'U'. The encoder or contracting path learns high-level features, doubling the number of feature maps at each level but are low-resolution feature maps. The decoder or expansive path upsamples the feature map, in the process halves the number of feature maps, and concatenates it with the corresponding high-resolution spatial features from the skip connection. The double convolutional layers after the concatenation ensure an optimal assembly of the upsampled feature maps and high-resolution features from the encoder. The 1x1 convolution at the end of the network is used to reduce the number of channels from 64 to 2 (generally, the number of classes). It uses only valid convolutions, thereby losing out on border pixels, making it necessary to crop the high-resolution feature maps before concatenation. Hence, the output of the network comes out to be spatially smaller than the input. The authors also employ various data augmentation techniques, such as shifts, rotations, grey-value variations, and elastic deformations. The loss function used is a weighted binary cross-entropy loss, which tackles the imbalance of classes.

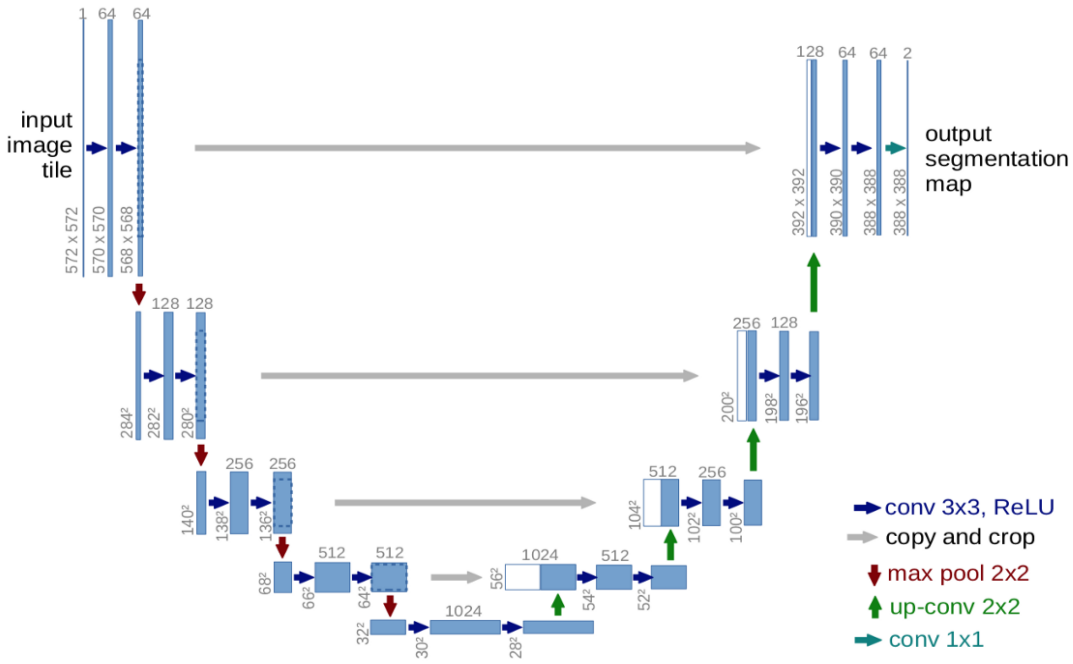

Figure 3.3 - The U-Net architecture [3]

Taking the U-Net from 2D to 3D, Cicek et al. [4], take the original U-Net and modify it to adapt to the volumetric segmentation of 3D medical data. The 2D convolutions, 2D max pooling, and 2D up-convolutions are replaced with their 3D counterparts. Also, batch normalization is used after the convolutional layers, before the ReLU activation function.

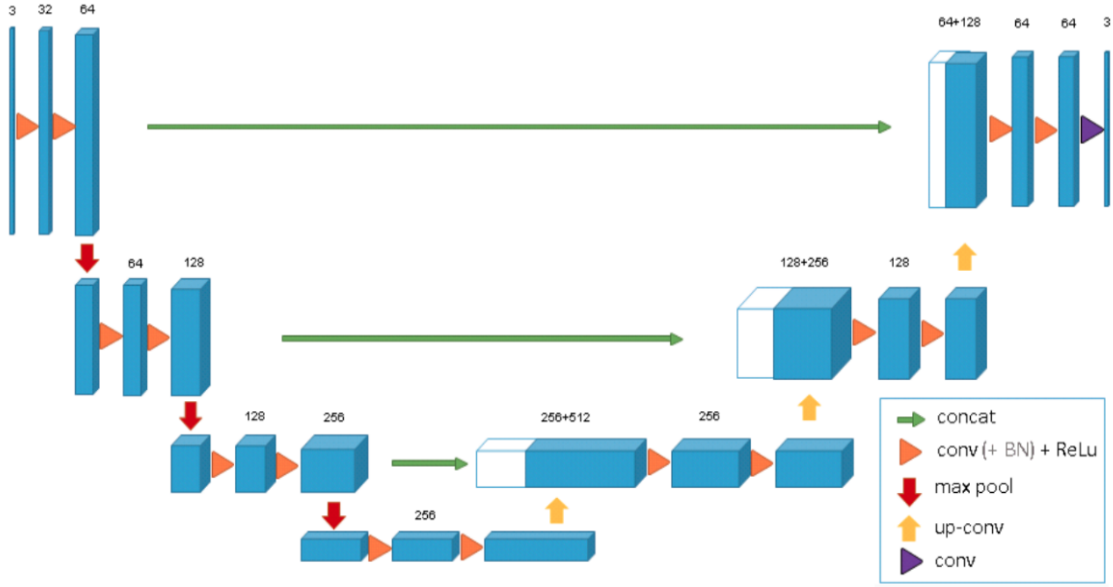

Figure 3.4 - The 3D U-Net architecture [4]

Milletari et al. [5] extend the concept of 3D U-Net further into V-Net. The V-Net uses convolutions to extract features, as well as to downsample, by using strided convolutions in place of max pooling. The convolutions are padded to maintain image resolution. At each level, there are residual connections in each block of the encoder and decoder. This helps the network to converge much faster than such a network without residual connections.

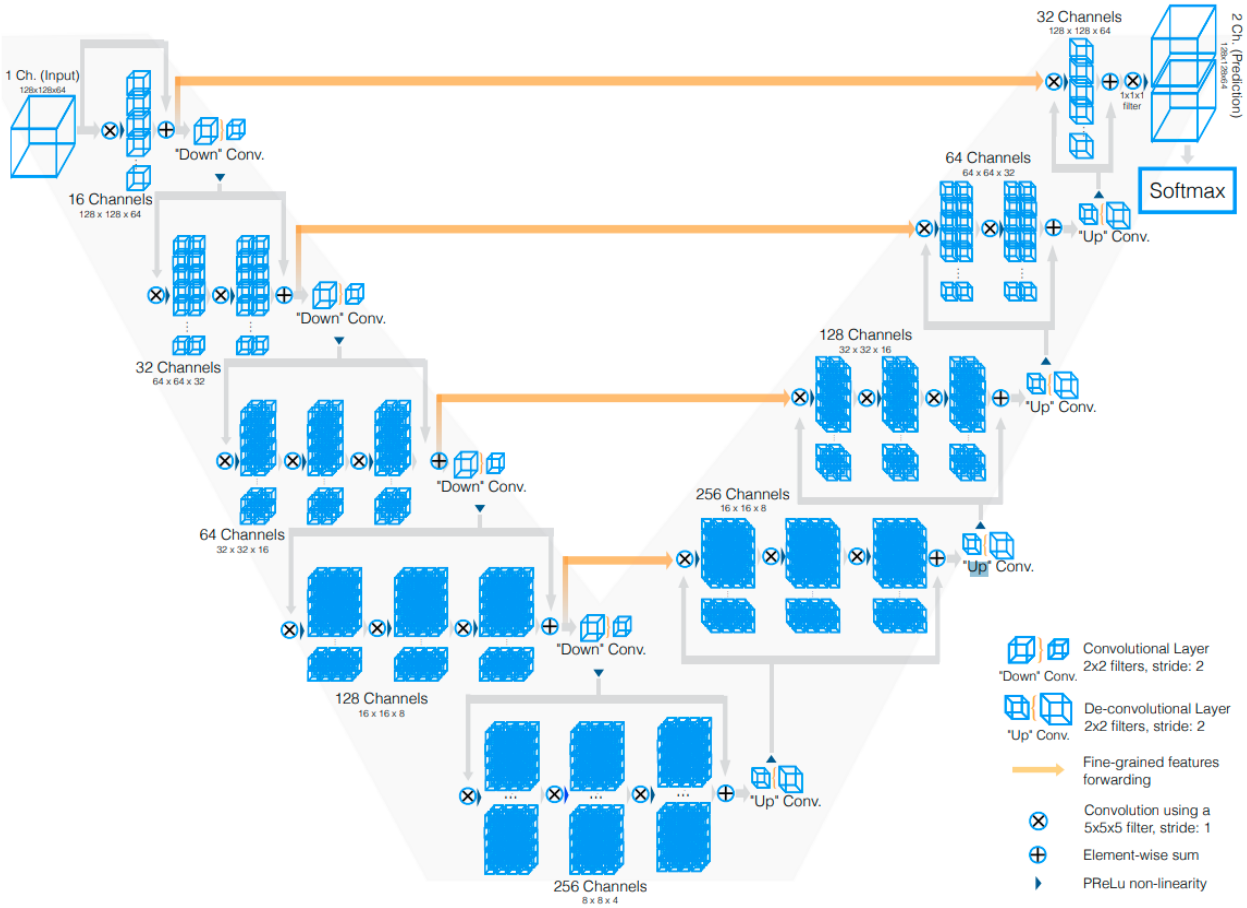

Figure 3.5 - The V-Net architecture [5]

In medical images, the region of interest usually covers a very small area (or volume) in the mask and hence the network trained on such data tends to get biased towards the background pixel. Dice loss, introduced by the authors, tackles this problem of severe class imbalance and is based on the dice coefficient. The dice coefficient D is calculated as:-

$$D = \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}$$

where p_i and g_i represents a voxel in the predicted segmentation volume P and in the ground truth segmentation volume G respectively, where i runs over N voxels. The network uses PReLU instead of ReLU.

With attention mechanisms on the rise, Oktay et al. [6] came up with Attention U-Net. The high-resolution feature maps from the skip connection bring in lots of irrelevant low-level features. Attention helps to suppress such features and focus on the relevant (salient) features. In their work, they introduced a self-attention mechanism called Attention Gate (AG). Attention coefficients, $\alpha \in [0, 1]$ can be thought of as assigning higher probabilities to the parts of the image that is learned (through backpropagation) as relevant and lower probabilities to the irrelevant parts.

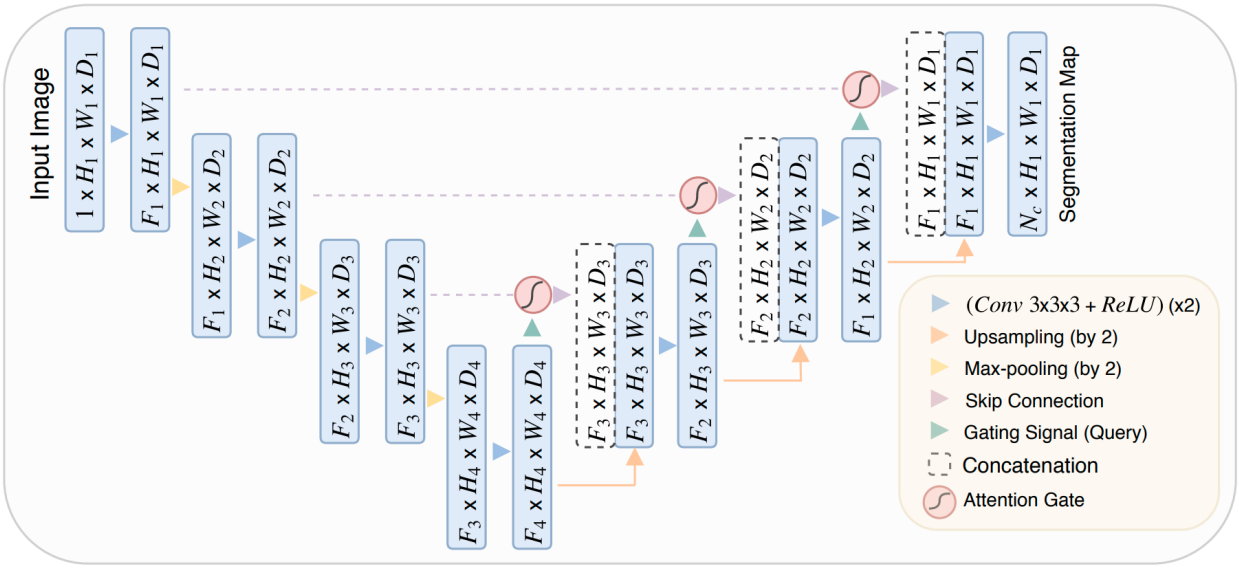

Figure 3.6 - The Attention U-Net architecture [6]

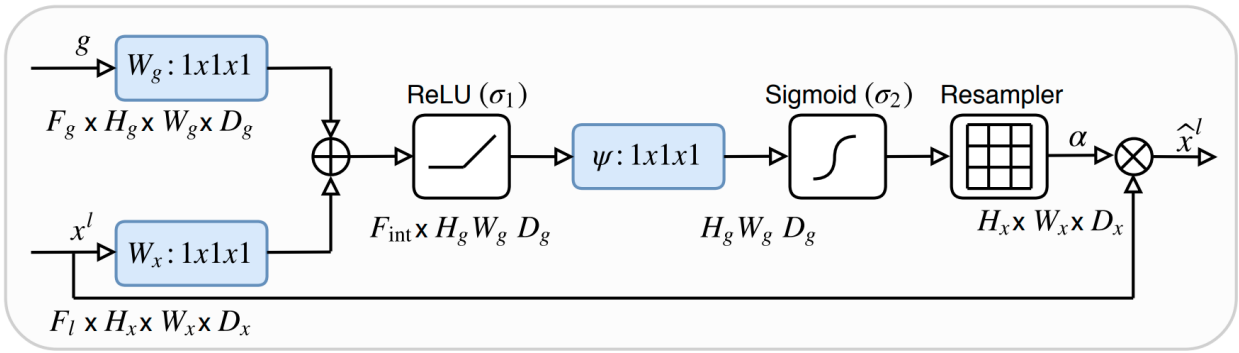

Figure 3.7 - Attention Gate [6]

The output of the block below (before upsampling) is convolved with a 1x1x1 filter and used as the gating signal. The gating signal contains contextual information that suppresses the low-level features. The Attention Gate module takes in this gating signal g and the high-res feature maps x^l coming from the encoder. The gating signal gets convolved with a 1x1x1 filter

to F_{int} feature maps. The skip connection input is convoluted to F_{int} feature maps and dimensions as of the gating signal. They are added and passed through a ReLU (σ_1), convolved with 1x1x1 (ψ) to get a single feature map. It is converted into the attention mask by applying the sigmoid activation (σ_2). The attention mask (α) is then resampler using trilinear interpolation to match the size of x^l. The element-wise product of the attention map and x^l is then sent normally as input from the skip connection.

Zhou et al. [7], inspired by DenseNets [8], connected the encoder and decoder blocks of the U-Net with a series of dense convolutional blocks. The intuition behind their architecture is that the optimization would be easier if the feature maps of the encoder were semantically similar to the decoder's, which is done by adding these dense blocks.

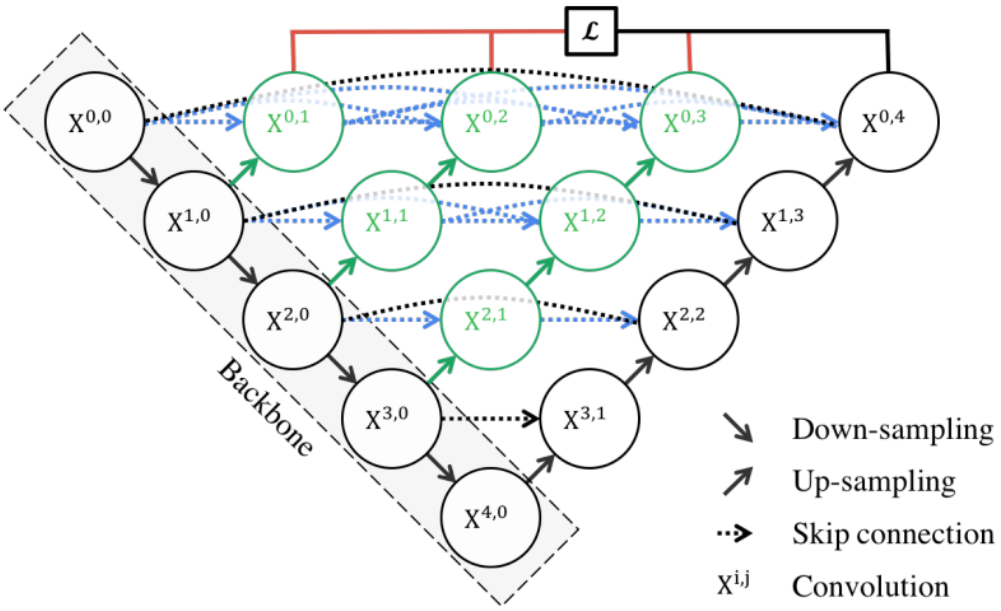

Figure 3.8 - The U-Net++ Architecture [7]

The ability to use deep supervision in the U-Net++ makes it special and allows for the use of the model in two modes, the accurate mode and the fast mode. The accurate mode uses all the feature maps from the top layer. The fast mode uses only the final feature map.

3.3. Generative Models

Goodfellow et al. [9] introduced the Generative Adversarial Networks (GAN) which consists of two Multi-Layer Perceptrons as the generator and the discriminator. The generator takes in a

random noise vector z and generates an image $G(z)$ with it. Theoretically, the random vector should be a vector representing the latent features of the image to be generated.

The discriminator is a binary classifier that gives the probability $D(x)$ or $D(G(z))$ whether a given image is real, given whether the image is actually real or generated respectively. The training of the GAN is done in an adversarial fashion, with the generator trying to produce fake images that can fool the discriminator as real. The discriminator, however, learns to classify the fakes better, and the generator is forced to up its game so as to be able to deceive the discriminator, and vice-versa. The generator and the discriminator play a minimax game to optimize the loss function,

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

The p_{data} and p_z are the probability distribution of the real data and the random noise vector. In the early stages, the discriminator usually wins and forces the gradient for the generator to vanish, and make learning very slow and unstable.

Deep Convolutional GAN (DCGAN) [10] is a GAN with convolutional layers instead of MLP, with pooling layers replaced by strided convolutions in the discriminator and transposed convolutions in the generator. Batch normalization is used in both. ReLU is replaced by LeakyReLU.

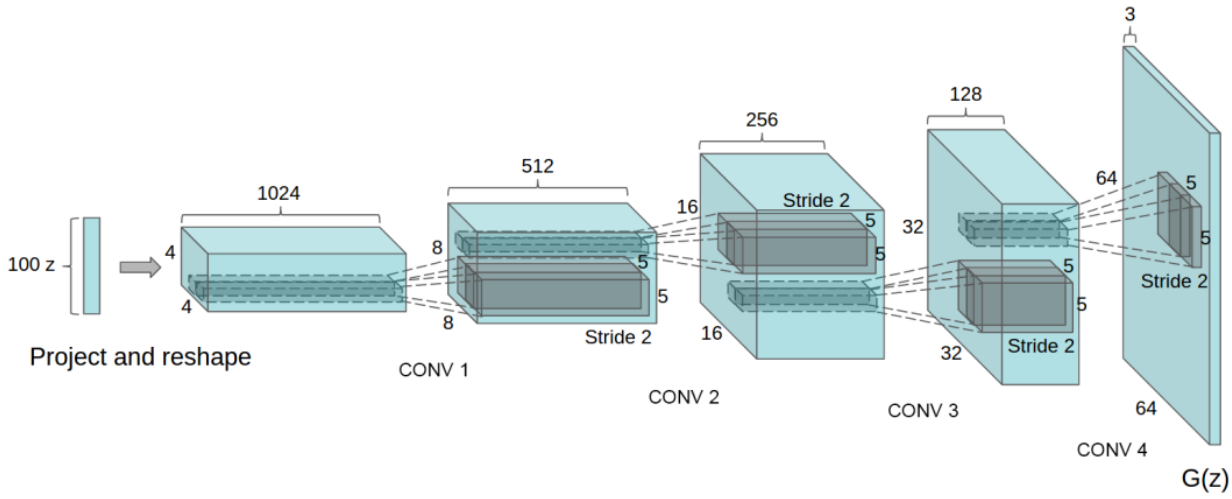

Figure 3.9 - The DCGAN generator architecture [10]

Arjovsky et al. [11] introduced the Wasserstein GAN (WGAN), which is an upgrade over the DCGAN or GAN in general. It addresses the problem of mode collapse. As the training of a GAN is in process, the generator learns to produce passable images for a particular mode, it tries to

neglect other modes and picks up that mode, and keeps generating similar images, in order to justify its goal of fooling the discriminator, i.e. generator is struck to a single mode of the distribution.
Instead of using the binary cross-entropy loss, where the discriminator is forced to give out values between 0 and 1, they introduced Wasserstein's loss (W-loss). The Wasserstein's loss gives approximate Earth Mover's distance between the real and generated distributions and its value is not limited between 0 and 1. Because there is no bound to the values of the discriminator, it improves without degrading its feedback to the generator.
This change in loss function greatly stabilizes the training of GANs. Now, with the new loss, the discriminator wants to maximize the distance and push the two distributions apart from each other and the generator wants to minimize the distance and bring the distribution closer. The discriminator is now called the Critic, as it is no longer discriminating, but instead criticizing.
To use the W-loss, the Critic needs to be 1-Lipschitz continuous, ie, the norm of the gradient at every point should be at most 1. The original WGAN introduces the techniques of weight clipping to enforce 1-Lipschitz continuity. After gradient descent, the weights are clipped to take values within a fixed interval. This limits the Critic's ability to learn.

Gulrajani et al. [12] introduced the technique of gradient penalty (WGAN-GP) to enforce 1-Lipschitz continuity. The gradient penalty is a regularization technique for the Critic's gradients, such that it penalizes the Critic when its gradient norm is greater than 1. However, this method doesn't strictly enforce 1-Lipschitz continuity but empirically works better than weight clipping.

Chapter 4: Dataset

4.1. Introduction

In order to achieve good results on our private dataset from the Sri Sathya Sai Institute of Higher Medical Sciences (SSSIHMS), which had very little to no WMH, we had to use two other publicly available datasets as part of our study. The dataset taken from various sources does increase the size of the data but also brings in the challenge of intra- and inter-observer variability, which is the major problem with WMH and other pathologies, where no real ground truth exists.

4.2. WMH Segmentation Challenge Dataset

This dataset [13] consists of 20 cases each from three different institutes: University Medical Center (UMC) in Utrecht, VU University Medical Centre (VU) in Amsterdam, and the National University Health System (NUHS) in Singapore. The scanners used were also from different vendors. 3T Philips Achieva, 3T Siemens TrioTim, and 3T GE Signa HDxt were used to scan the patients in UMC Utrecht, NUHS Singapore, and VU Amsterdam respectively.

Each case consists of a 3D T1-weighted and a 2D multi-slice FLAIR image. As the manual reference standard is defined for 2D multi-slice FLAIR images, the 2D multi-slice T1-weighted sequence is resampled from the 3D T1 to match the FLAIR one. Also, the 3D FLAIR images available in the case of VU Amsterdam were resampled to slices of 3 mm along the transverse axis. All the data was available in NIfTI format. Other details about the dataset have been provided in Table 4.1.

The images were inhomogeneity corrected using SPM12, and the 3D T1-weighted were registered to the FLAIR sequences using the elastic toolbox. The data can be accessed from the challenge website.

WMH in the FLAIR was manually segmented following the STRIVE criteria. One observer with extensive experience in segmenting WMH marked the WMH regions, which was peer-reviewed by a second observer with 11+ years of experience in the field of neuroimaging and clinical neurology. The sixty training images were segmented with two more observers to mitigate inter-observer variability, one trained for WMH but lacked proper experience, and one trained for WMH and had a good experience.

The masks contain three kinds of labels, 0 for the background, 1 for the WMH, and 2 for other brain lesions.

4.3. Tongji Hospital Dataset

This dataset [14] contains images from all three prime modalities, T1-weighted, T2-weighted, and T2-FLAIR of 94 patients with the WMH labels (in the NIfTI format). The data was collected from subjects of the inpatient and outpatient departments of Tongji Hospital, Wuhan, China. The images of subjects below 18 years of age, with intracranial surgery or poor quality, were discarded.

The images were preprocessed in the following order:-

1. Inhomogeneity correction using the N4 algorithm applied to T1-weighted images.
2. Registration of the T1-weighted and T2-weighted images to the FLAIR images using Advanced Normalization Tools (ANTs) via rigid and affine transformations.
3. Skull stripping the T1-weighted image using FSL BET (FMRIB Software Library Brain Extraction Tool) and using the mask to strip non-brain tissues from corresponding T2-weighted and T2-FLAIR images.
4. Quantile Normalization was applied to all images to bring the intensities to a range of [0, 1] by the formula,

$$I' = \begin{cases} 0 & \text{if } I < P_{0.001}; \\ 1 & \text{if } I > P_{0.999}; \\ \frac{I - P_{0.001}}{P_{0.999} - P_{0.001}} & \text{otherwise.} \end{cases}$$

Hence, only the skull-stripped (preprocessed) images are available as a part of the dataset, not the original images.

Again, the background receives label 0, the WMH label 1, and other lesions label 2. WMH delineation was done independently by two skillful observers under the guidance of an experienced neuroradiologist. If for any two images, the dice score between the two sets of segmentation maps is below 0.5, the two observers came together to discuss and repeated the independent delineation until the dice score crossed 0.5. The regions of intersection between the two observers were marked as certain WMH labels and the regions where only one marking was found, were labeled as suspected WMH.

4.4. SSSIHMS Dataset

The last dataset, also the dataset of concern, is the MRI data that was obtained from Sri Sathya Sai Institute of Higher Medical Sciences. This dataset consists of MRI data of 74 patients. Unlike the other two datasets, these scans are not filtered to retain only the scans

that have a good volume of WMH in them. There is no filtering as per age bracket. The data was collected as a part of regular diagnosis and is now being used to study WMH. The data from SSSIHMS is raw and has not been preprocessed. The images were available as DICOM files.

For every case, there are T1-weighted and FLAIR images with their corresponding mask annotated by a neuroradiologist at SSSIHMS. The mask consists of 0 and 1, for the background and WMH respectively.

Table 4.1 - Summary of the dimension, slice thickness, and scanner hardware used at various institutes that make up the datasets

Institute	Scanner	Slice Interval	Dimension of scans
VU Amsterdam	GE Signa HDxt	3 mm	132 x 256 x 83
NUHS Singapore	Siemens TrioTim	3 mm	252 x 232 x 48
UMC Utrecht	Philips Achieva	3 mm	240 x 240 x 48
Tongji, Wuhan	UIH uMR 780	5.5 mm	456 x 396 x 18
Tongji, Wuhan	GE Signa HDxt	6 mm	512 x 512 x 16
Tongji, Wuhan	GE Discovery MR750	6 mm	512 x 512 x 17
SSSIHMS	Unknown	6.5 mm	270 x 320 x 26

Chapter 5: Methodologies

5.1. Introduction

With all the data at our disposal, it was necessary to combine them in the right manner so that the data doesn't get biased towards a particular set of images. Also, the data were in various formats, sizes, and levels of preprocessing. Hence, it is important to account for these and bring all the data to the same level to be able to train the neural networks successfully. Data preprocessing and preparation form a huge chunk of the process, prescribed in [13][14]. Data augmentation is the next logical step that helps feed the data-hungry networks with more samples and also adds certain invariances to the network. Under "2D segmentation", all the model architectures used for segmentation as a part of the project are described with the implementation details. Under "Learning the distribution", the generative models used are discussed along with the metric used to measure how much of the distribution has been captured by the generative model.

5.2. Data Preprocessing and Preparation

As the data from Tongji Hospital is available as skull-stripped images, the data from other datasets cannot be combined as it is, to form a larger dataset. We'll need to skull-strip the images from WMH Challenge and SSSIHMS datasets. In addition, the T1 images from SSSIHMS are raw and they have to be inhomogeneity corrected and co-registered to the FLAIR sequences.
Therefore, we form one large dataset that contains all the images with the skull on, combining WMH Challenge and SSSIHMS data.

The steps in the preprocessing of the SSSIHMS dataset are given as follows:-

1. As the data in the case of SSSIHMS is in DICOM format, we convert the DICOM images to NIfTI using the dcm2niix.
2. We applied the N4 algorithm for inhomogeneity correction or bias-field correction using Advanced Normalization Tools (ANTs). N4 is a type of bias field correction algorithm that follows after N3. It removes the bias field, which is a low frequency signal that distorts the MRI image leading to non-uniform brightness across the image. This happens mostly at higher field strengths, the images acquire an intensity gradient, which hurts the segmentation algorithms.

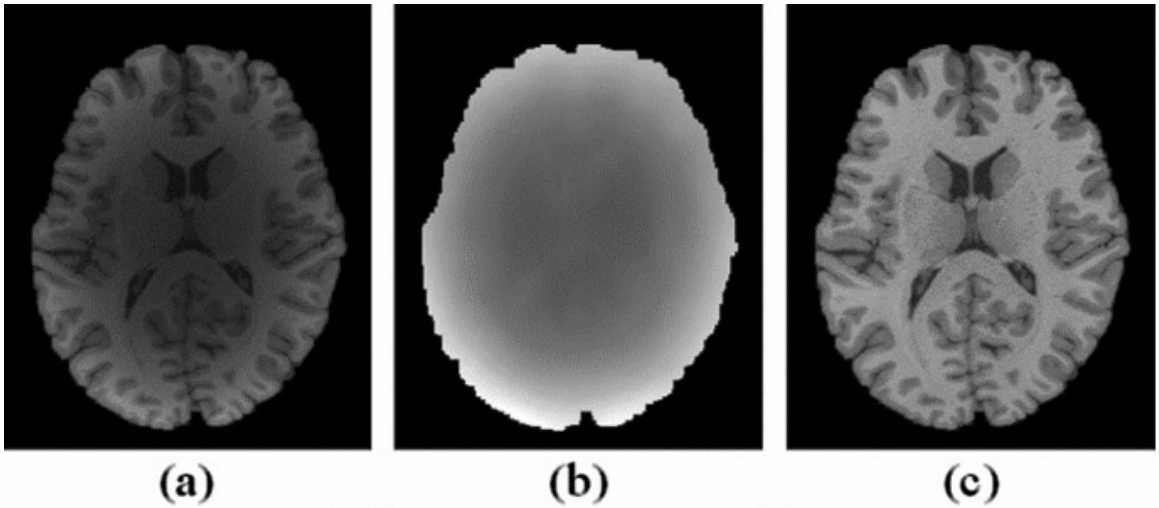

Figure 5.1 - (a) MRI image with bias field, (b) The intensity gradient mask for the MRI image, (c) The bias-field corrected image (source)

3. Now that pathology is easily perceived on the FLAIR sequences, we register our T1-weighted images to the FLAIR sequences using ANTs. Registration is important as it brings the T1-weighted image to the same space as the FLAIR by rotating, scaling, and translating in the form of rigid and affine transformations.

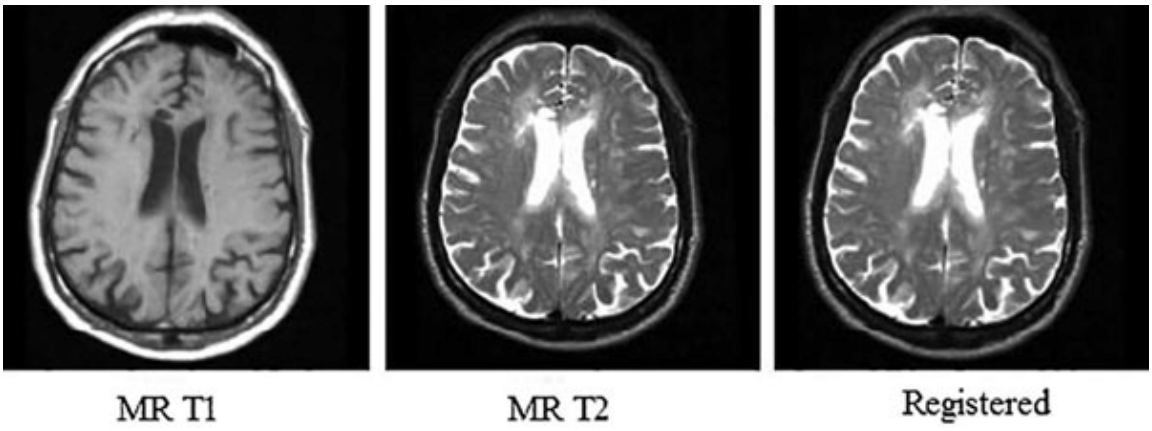

Figure 5.2 - The T2 image (middle) is registered to the T1 image (left) gives the registered T2 image (right) (source)

4. Finally, FSL BET is used to strip away the non-brain tissues from the brain tissues from the FLAIR images and then the holes inside the binary brain mask are filled using morphological operations. The binary mask, thus obtained, is multiplied with the FLAIR and T1-weighted 2D sequences to get the final brain-extracted sequence.

Although brain extraction helps the process of segmentation, in our case, this is done to be able to use the Tongji Dataset.

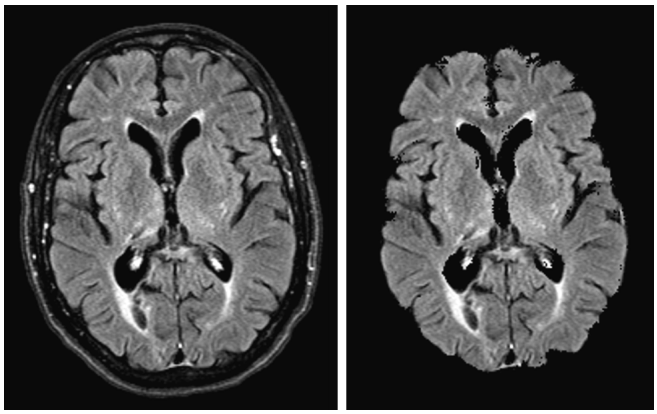

Figure 5.3 - The original image and its brain-extracted version (source)

The 3D images are then cropped or padded to 200 x 200. Also, some slices were removed from the final datasets as they do not contain brain tissues (or white matter) as such. These are mostly the starting and ending slices of every scan. The details of the exclusion of such slices as presented in Table 5.1. The 3D images are then converted into 2D slices.

Table 5.1 - Slices excluded from the 3D image.

Institute	VU, Amsterdam	NUHS, Singapore	UMC, Utrecht	Tongji, Wuhan	SSSIHMS
Slices (removed from start, removed from end)	(30, 20)	(6, 6)	(6, 6)	(2, 2)	(5, 5)

We create a dataset by combining the already preprocessed WMH Challenge data and the SSSIHMS data obtained after the third step of preprocessing and call this Dataset-1.
Dataset-2 is just a copy of Dataset-1. The WMH Challenge data is further preprocessed to remove the non-brain tissues using FSL BET. With this, we form another dataset which consists of the WMH challenge data, the Tongji data, and the SSSIHMS after all the preprocessing steps, and call this Dataset-3.
Further, Gaussian normalization is applied to the Dataset-1 3D images, and Quantile Normalization as prescribed in [14], was applied to the Dataset-3 3D images. Also, Gaussian normalization is applied to Dataset-2 by creating a brain mask using thresholding for T1 and

FLAIR and applying Gaussian normalization on both T1 and FLAIR using the masked brain region, as done in [15]. The thresholds for the T1 and FLAIR were determined empirically.
The train-validation split is an 80-20 split in all the cases (UMC, VU, NUHS, SSSIHMS, and Tongji). Also, care is taken to ensure a similar number of slices from each of these places, so that our dataset is not biased to one kind of data.
The usual preprocessing of brain MRI data consists of brain extraction and then registration, but we chose this way of preprocessing given in [14].

Also, for preparing the data for the generative modeling, we modify Dataset-3 to contain images with WMH and use the dataset for generative modeling.

5.3. Data Augmentation

Deep Neural Networks are known to be data-hungry. Data augmentation not only increases the size of the dataset but also can add some necessary invariances that we expect our network to learn, due to the position and orientation of the head.
Dataset-1, Dataset-2, and Dataset-3 were all augmented using similar augmentation tactics. The augmentation tactics are taken from the winners of the WMH Segmentation Challenge, team sysu media [15], and tweaked to our use case.
Rotation, Scaling, Shearing, and Horizontal Flips were applied to every slice to obtain a dataset 5 times larger than the original. The parameters of the augmentation are given below in Table 5.2.

Also, for the generative modeling, we apply Rotation, Scaling, and Horizontal Flips. Horizontal Flips are used as there is no study that suggests the correlation of the scale or severity of WMH to the hemispheres. We avoid Shearing as it distorts the images, which would lead to the generation of distorted images by the generator. The parameters of the augmentation are given below in Table 5.3.

Table 5.2 - Parameters for data augmentation for segmentation purposes.

Transformation	Rotation	Shearing	Scaling	Horizontal Flip
Parameters	[-10°, 10°]	[-10°, 10°]	[0.9, 1.1]	-

Table 5.3 - Parameters for data augmentation for generation purpose.

Transformation	Rotation	Scaling	Horizontal Flip
Parameters	[-5°, 5°]	[0.9, 1.1]	-

5.4. Automatic Segmentation

Segmentation, apart from deep learning based methods, can be done using traditional image processing and machine learning methods.
In traditional image processing, thresholding is a way to perform segmentation that actually should gel well with our use case of segmenting WMH. But, the thresholding ended up giving huge false positives as the intracranial lesions other than WMH could not be told apart by just thresholding. Apart from that, the thresholded image contained a lot of noise, sometimes sharing intensities with other white matter tissues. Although morphological operations such as erosion, and dilation can be applied to get rid of the noise, the WMH are sometimes so tiny that we end up pruning them too. Also, these techniques do not generalize well to the various scans.
Other techniques, such as region growing too, falls short in our case, as there is no fixed seed point, which would not have been the case in, say, left-ventricle segmentation.
Machine Learning techniques are avoided as it involves the task of manual feature extraction to be fed to the model which is impossible as there are tens of thousands to choose from and choosing a subset might result in the loss of relevant information.
Therefore, we turn to deep learning based methods, the Convolutional Neural Networks (or Fully Convolutional Networks), as they have been known to work well with image or grid data.

5.4.1. Network Architectures

This section lists all the network architectures tried as part of the project.

1. Vanilla U-Net:-
   As described in Chapter 3, the U-Net architecture is *the* architecture for medical image segmentation. In the original architecture, due to the use of valid convolutions, there was a loss of border pixels in the final image. The modified U-Net architectures employ same or padded convolutions at all the places to maintain the spatial dimension of the final output segmentation map. The double convolutional layers have been replaced

by two Convolution Blocks, which consist of a convolutional layer, followed by a Batch Normalization layer and ReLU activation. Further, the upsampling used here is a bilinear interpolation, followed by a Convolution Block. The batch normalization is applied to tackle the problem of internal covariance shift, leading to faster convergence.

2. Attention U-Net:-
   The Attention U-Net architecture, which originally was 3D was adapted to 2D for our use. In addition, dropout layers were used to achieve a regularizing effect, to prevent overfitting. Our implementation included 4 levels instead of the 3 levels in the official implementation.

3. U-Net with Inception Blocks:-
   With Inception block debuting in the GoogLeNet [16], the network had filters of different sizes at the same level, with the idea of merging features information from various scales at every level. This made the network wider than deeper. We modified the vanilla U-Net and replaced the Conv Block with the Inception Block.

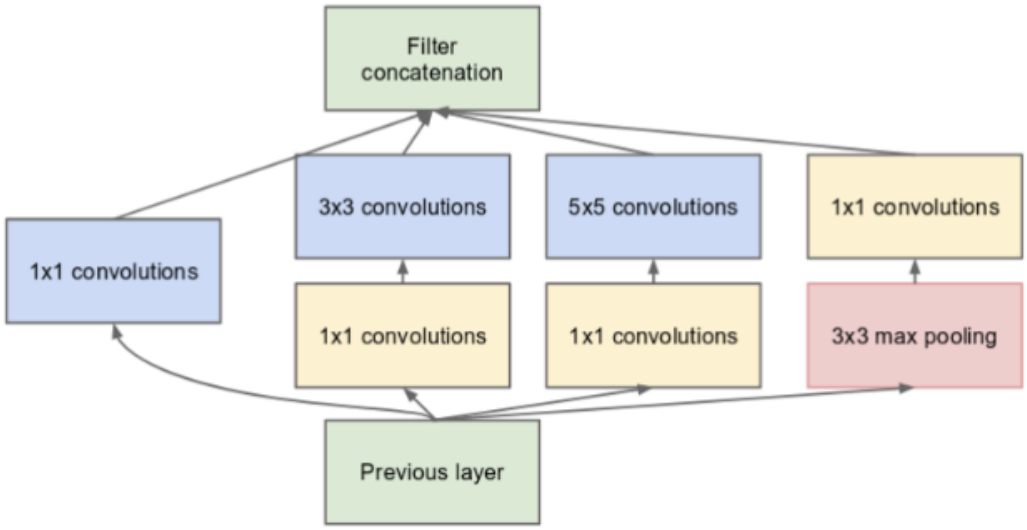

Figure 5.4 - The Inception Block [16]

4. U-Net++:-
   The U-Net++ implementation also closely resembles the official implementation, with 3 layers and with the option of using deep supervision.

5.4.2. Loss Functions

There are a bunch of loss functions that are in use for semantic segmentation. Here are a few that we have experimented with and used in our project:-

1. The Dice loss is the go-to loss function for image segmentation tasks, as it handles class imbalance well. It is derived from the Dice coefficient, which measures the overlap between images. It weighs and penalizes false positives and false negatives equally.
2. The binary cross-entropy loss is also a commonly used loss function for binary image segmentation. It is easy to compute and works well when there is very less class imbalance between foreground and background. Therefore, it suffers in cases where there is a severe class imbalance, rendering it almost obsolete for medical image segmentation.
3. The DiceBCE loss is just a linear combination of the two losses, Dice and binary cross-entropy loss. The most common implementation is averaging both losses.
4. The Tversky loss is a special kind of loss function that introduces hyperparameters α and β that lets us control whether we want to penalize the false positives or false negatives more, respectively.
5. The Focal Tversky loss combines Tversky loss with the Focal loss term, which gives more weightage to hard-to-classify examples rather than easy examples. This introduces another hyperparameter γ that determines the degree of weightage given to hard examples. Higher values give more weightage to hard examples, while lower value gives more weightage to easy examples, enabling faster convergence.

5.4.3. Metrics

There are a plethora of metrics used for validating and evaluating segmentation models. We would only look at a few of these.

1. Pixel Accuracy - This is the most naive metric for segmentation. This metric compares each pixel in the predicted mask with the ground truth mask. This works well enough when there is very less class imbalance, but in the case of class imbalance, it cannot be relied upon as a metric. In case, the ground truth mask contains very little amount of segmented area, then a black image will yield a pixel accuracy close to 90 percent, which is misleading.
2. Dice Score - Earlier used as a loss, this metric measures the overlap between the predicted and the ground truth mask. It is defined as twice the intersection of masks

divided by the sum of the number of pixels in the two masks. Unlike pixel accuracy, the dice score would not fail to penalize the black predicted masks.

3. IoU Score (Jaccard Index) - It also measures the overlap between the predicted segmentation mask and the ground truth mask. It is defined as the ratio of the intersection of the two masks to the union of the two masks.
4. Precision - It measures the percentage of pixels correctly classified as positive out of all the pixels classified as positive (in the predicted mask).
5. Recall - It measures the percentage of pixels correctly classified as positive out of all the pixels that are actually positive (in the ground truth).

5.5. Learning of the distribution

First, we applied two generative models, namely DCGAN and WGAN. The generator and the discriminator architectures used are the same for both. The generator is trained to produce synthetic brain MRI images with WMH, while the discriminator (or critic) is trained to classify (or criticize) the real and synthetic scans. After training, the generator is now able to generate scans with WMH that are similar to those in the training set. The generator can be viewed as a function that maps a random noise vector to a synthetic brain MRI with WMH. The learned distribution of brain MRI scans with WMH has been captured by the probability distribution function of the generator network. By adjusting the components of the noise vector, we can adjust the shape, size, and other attributes of WMH. This is called controllable generation [17]. There are several metrics to quantify the learned distribution, including Inception Score, Precision & Recall, and Frechet Inception Distance (FID) [18]. Here, we will be using FID to quantify the learned distribution.

First, to evaluate the GANs, we look at the feature distance between the real and the synthetic samples. Frechet Distance is a distance metric used to measure the distance between curves, hence, can also be used for normal distributions. For the task of feature extraction, FID uses the Inception v3 network trained on ImageNet, which yields a 2048-dimensional feature vector. As in [19], we apply FID to our use case, we'll have to fit a multivariate normal distribution to the feature vectors of the real and the generated images, and then calculate Frechet Distance between the two distributions.

Chapter 6: Experiments and Results

6.1. Introduction

In this section, see the various experiments done with the loss function and other hyperparameters, metrics chosen to evaluate, data preprocessing and augmentation techniques, and finally, network architecture chosen for our tasks at hand. This section is divided into two parts, one that summarizes the experiments and results of the segmentation problems, and the other that summarizes the same for the distribution learning part of the project.

6.2. Automatic Segmentation

The first experiment was to apply the U-Net architecture to the SSSIHMS Dataset directly. This led to very poor results, especially with the BCE loss. Using dice loss and choosing the slices with some region of interest on it leads to a validation dice score of 0.6251. But this statistic cannot be relied upon as the dataset was biased. As our dataset is not sufficient for the task at hand, we add more data in the form of the WMH Challenge dataset and the Tongji Hospital dataset and create Dataset-1, Dataset-2, and Dataset-3.
Starting with the base model of U-Net, we tried several loss functions to see what worked the best, on Dataset-1 without augmentation.
The loss functions at scrutiny are Dice loss, DiceBCE loss, Tversky loss, and Focal Tversky loss.
We consider recall to be more important than precision, as in medical cases, it is much more problematic to miss out and underestimate the criteria at hand than to overestimate it. We are allowing for more false positives but reducing false negatives in the process.
In addition, the Tversky and Focal Tversky loss introduces hyperparameter tuning. We have tried combinations α and β values, such as 0.2 and 0.8, 0.3 and 0.7, 0.5 and 0.5 (which is the dice loss) for Tversky loss, penalizing the false negatives harder. For Focal Tversky loss, we try γ values of 1.33 and 2, along with the combinations of α and β values.

Network Hyperparameters:
We experiment with different values of batch sizes from 16, 32, 48, and 64, and found that larger batch sizes stabilized training. The experiments could be done till 64 due to the GPU memory constraints for most of the networks and a batch size of 32 for U-Net++. The learning rate was chosen to be 0.0001 for most of the models.

Table 6.1 - Validation metrics of U-Net trained with various loss functions on Dataset-1 without augmentations.

Loss Functions	**Dice**	**IoU**	**Precision**	**Recall**
DiceBCE	0.7675	0.7001	0.8438	0.8143
Dice	0.7700	0.7193	0.8434	0.8118
Tversky (α=0.3, β=0.7)	**0.7790**	**0.7092**	0.7868	0.8757
Tversky (α=0.2, β=0.8)	0.7646	0.6931	0.7600	**0.8903**
FocalTversky (α=0.3, β=0.7, γ=1.33)	0.7511	0.6818	0.7662	0.8558
FocalTversky (α=0.2, β=0.8, γ=1.33)	0.7558	0.6858	0.7465	0.8898
FocalTversky (α=0.2, β=0.8, γ=2.00)	0.7183	0.6457	0.6938	**0.9033**

Also, as we are primarily concerned with the SSSIHMS Dataset, Table 6.2 consists of the same networks, but performance only of the SSSIHMS Dataset is recorded.

Table 6.2 - Validation metrics of U-Net trained with various loss functions on the SSSIHMS part of Dataset-1 without augmentations.

Loss Functions	**Dice**	**IoU**	**Precision**	**Recall**
DiceBCE	0.6345	0.5576	0.8321	0.6450
Dice	0.5836	0.5079	0.8034	0.6283
Tversky (α=0.3, β=0.7)	0.6461	0.5622	0.7258	0.7349
Tversky (α=0.2, β=0.8)	**0.6614**	**0.5782**	0.7140	**0.7747**
FocalTversky (α=0.3, β=0.7, γ=1.33)	0.6100	0.5294	0.7096	0.7013
FocalTversky (α=0.2, β=0.8, γ=1.33)	0.6284	0.5468	0.6607	0.7724
FocalTversky (α=0.2, β=0.8, γ=2.00)	0.4960	0.4143	0.5101	**0.7860**

Results on Dataset-1 without augmentations:

1. Tversky loss, with α and β values of 0.3 and 0.7, performed the best with respect to the dice score and the IoU score.
2. Focal Tversky loss, with α, β and γ values of 0.2, 0.8, and 2, excelled at recall, but falls short of the dice score and IoU.
3. The next best recall comes from Tversky loss, with α and β values of 0.2 and 0.8.

Results on the SSSIHMS part of Dataset-1 without augmentations:

1. Tversky loss, with α and β values of 0.2 and 0.8, performed the best with respect to the dice score and the IoU score.
2. Focal Tversky loss, with α, β and γ values of 0.2, 0.8, and 2, excelled at recall, but again falls short of the dice score and IoU.

So, we choose Tversky loss (α=0.3, β=0.7) and Tversky loss (α=0.2, β=0.8) for further experiments.
Moving on, we now test our preprocessing techniques to see which of them is more effective.

Table 6.3 - Validation metrics on various datasets for vanilla U-Net trained with Tversky loss (α=0.3, β=0.7).

Dataset	**Dice**	**IoU**	**Precision**	**Recall**
Dataset-1	**0.7790**	**0.7092**	0.7868	**0.8757**
Dataset-2	0.7746	0.7060	0.7953	0.8683
Dataset-3	0.7771	0.6993	0.8063	0.8548

Results:

1. We find that in the case of preprocessing, N4 bias correction and Co-Registration of the T1 to the FLAIR image, followed by Gaussian normalization of the images works the best.
2. Even though we have access to more data in the case of Dataset-3, there is no considerable improvement in the validation statistics. We do use this dataset further down for our transfer learning methods.

With respect to augmentations, there is not a considerable change but the dice score does improve a little in every case with augmentation. We continue with augmentations as it does involve variance to the dataset.
Now that we have experimented with the loss functions and the data preprocessing techniques, we now choose the best network architecture to go forward with.

Table 6.4 - Validation metrics for different architectures on Dataset-1.

Architecture	**Dice**	**IoU**	**Precision**	**Recall**
U-Net	**0.7790**	0.7092	0.7868	**0.8757**
U-Net with Inception Blocks	0.7707	0.7088	0.8352	0.8254
Attention U-Net	**0.7811**	0.7100	0.7998	**0.8748**
U-Net++ (w/o deep supervision)	0.7693	0.6993	0.7898	0.8727
U-Net++ (with deep supervision)	0.7559	0.6867	0.7706	0.8750

The best networks in our case, comes out to be U-Net, and Attention U-Net. Though U-Net++ is a superior model on paper this does not reflect in our experiments with Dataset-1. Also, deep supervision doesn't improve U-Net++.

Transfer learning is a type of machine learning technique where a model trained on one task is reused for another task. This involves taking a pre-trained model and freezing its encoder parameters, as it has learned to derive feature representations from the original training data, and only the decoder parameters are learned via training on the new dataset.
We train our models on data from VU Amsterdam, NUHS Singapore, and Tongji Wuhan, and transfer their learning to our SSSIHMS dataset. The choice of the data for transfer learning is a consequence of the observation that the data from VU Amsterdam, NUHS Singapore, and Tongji Wuhan when put together and trained gave similar individual testing results, while the data from UMC Utrecht and SSSIHMS usually performed low in their comparison. The loss function was empirically chosen to be Tversky (α=0.2, β=0.8) based on previous results.

Finally, we settle with the vanilla U-Net model with transfer learning for the final model, as it performs the best among all the experimented combinations of architectures, datasets, and loss functions.

Table 6.5 - Validation metrics for transfer learning on SSSIHMS dataset using various U-Net-like architectures and Tversky loss (α=0.2, β=0.8).

Architecture	**Dice**	**IoU**	**Precision**	**Recall**
U-Net	**0.7110**	**0.6301**	**0.7590**	0.7720
Attention U-Net	0.6743	0.5934	0.7142	**0.7863**

Final results and findings:

1. Finally, to get the best results on the SSSIHMS, we had to turn to transfer learning with vanilla U-Net. To make sense of the results, we compare them with the state-of-the-art methods for WMH segmentation. The top spot is still claimed by sysu_media [15] in terms of Dice score and Recall. Our Attention U-Net does perform better in both cases but is trained on a smaller WMH Challenge dataset, which lacks the data from 2 other scanners and also we prune out more slices from the start and the end. This might explain the higher performance of our model.
2. Though we have tried to focus on false negatives more, we do have some of them in the prediction masks in Figure 6.2, given a recall value of 0.77. Also looking carefully at the images in Figure 6.1 and Figure 6.2, we can notice that there are more false positives around the regions predicted as WMH.
3. In all the experiments done on various datasets, our models do not perform so well on our SSSIHMS dataset when compared to other datasets. When using Dataset-3 which contains all the datasets combined, we notice that SSSIHMS data always performs poorly with respect to Dice, IoU, and Recall. Despite the presence of Tongji data, the network actually manages to obtain a decent score for the WMH Challenge data and the Tongji data. This discrepancy in all the experimental scenarios in the SSSIHMS case calls for the guidance of domain experts, i.e. radiologists in our case.

Table 6.6 - Validation metrics of several methods for WMH segmentation [14].

Teams	**Dataset**	**Method**	**Dice**	**Recall**
sysu media	WMH Challenge	CNN	**0.80**	**0.84**
Tongji	WMH Challenge	2D VB-Net	0.79	0.79
Ours	*WMH Challenge (Modified)*	*Attention U-Net*	**0.82**	**0.90**
Ours	*SSSIHMS*	*U-Net (Transfer Learning)*	**0.71**	**0.77**

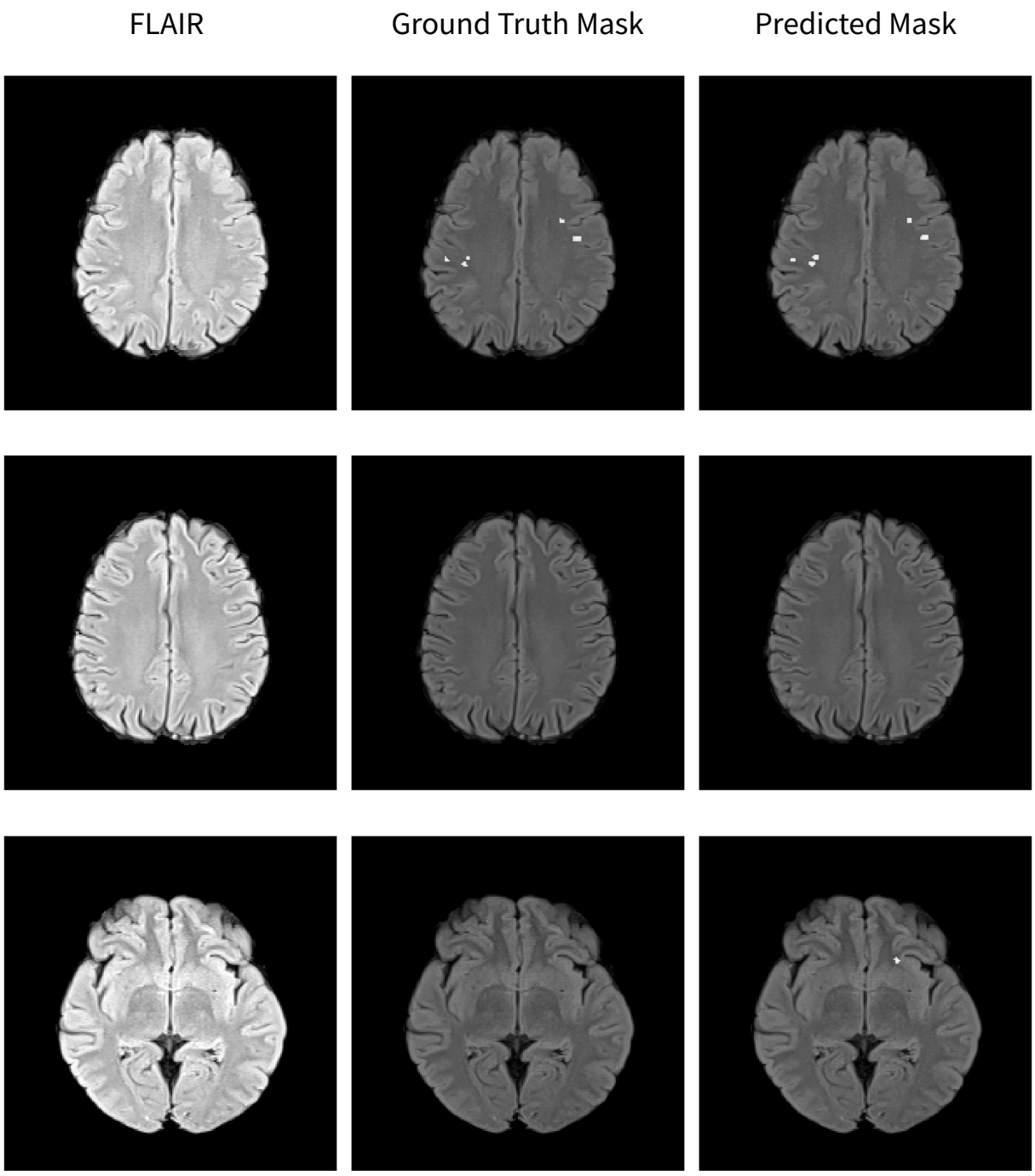

Figure 6.1 - The first and second set of images shows a near-perfect prediction, and the third set shows the prediction of false positives.

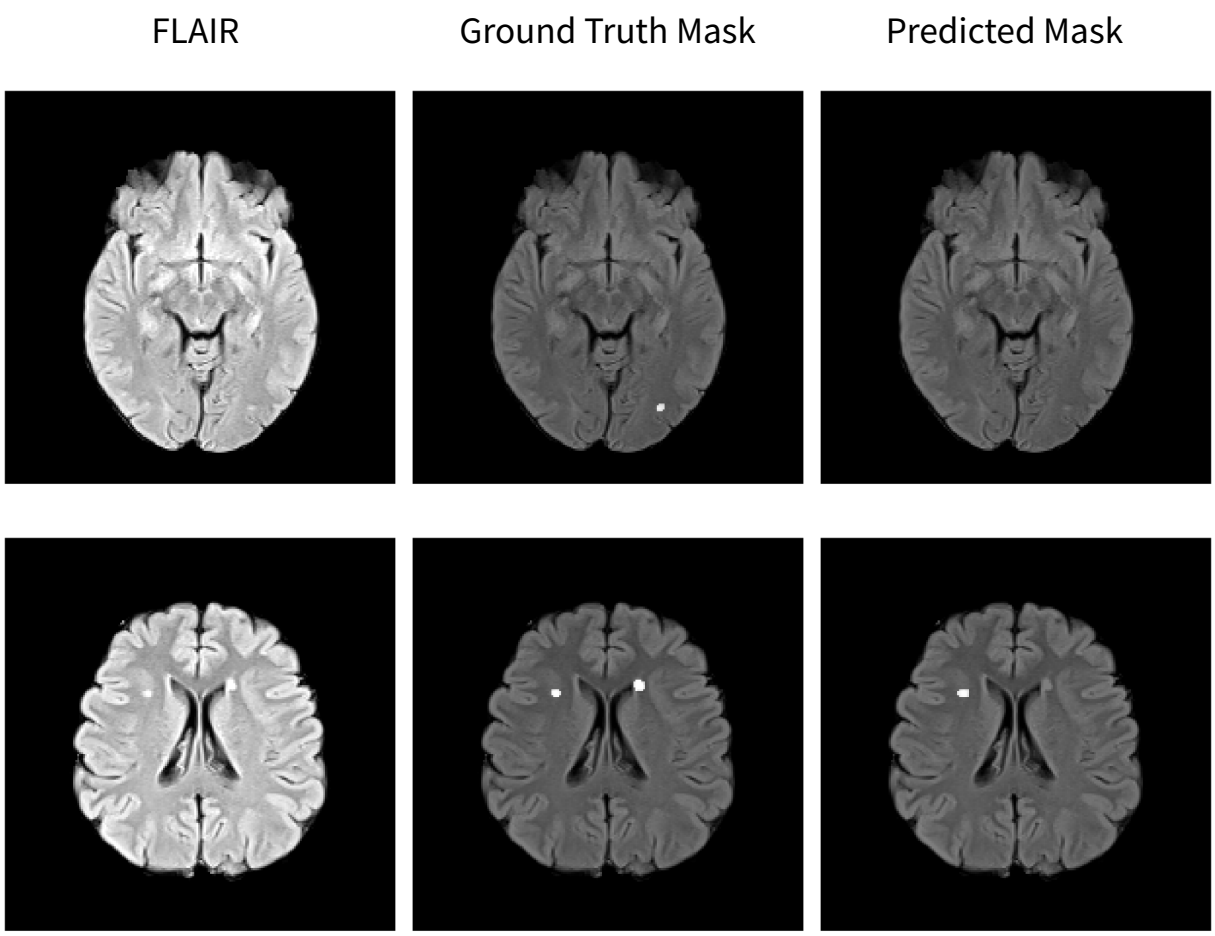

Figure 6.2 - The first and second set of images here shows the presence of false negatives in the predictions.

6.3. Learning of the Distribution

We tried the DCGAN architecture first but after a while it stopped learning as the discriminator overpowered the generator into becoming the perfect discriminator, denying gradients for the generator to learn. Therefore, we shifted to WGAN, which was quite stable and easy to train.

Table 6.7 - Summary of hyperparameters for the WGAN.

Hyperparameters	Batch size	Learning rate (generator)	Learning rate (discriminator)
Values	16	0.0002	0.0001

The learning of the distribution is quantified with the metric Frechet Inception Distance, which uses Inception v3 to extract features from the real and synthetic images, forms a multivariate normal distribution with the means and covariances of the respective images feature vectors then calculates Frechet Distance between the two distributions. FID closer to zero indicates that the generator has captured a good amount of the distribution of the real lesion-marked images.
The WGAN in our implementation has been trained for 900 epochs, taking almost 40 hours to train. The FID in our case comes out to be 134.4184. To give a sense of comparison, we'll take a glance at the results from [19].

Table 6.8 - Comparison of various GAN architectures from [19].

Dataset	**GAN**	**FID Score**
ACDC (Heart)	DCGAN	60.12
ACDC (Heart)	WGAN	**74.30**
ACDC (Heart)	StyleGAN	24.74
Our Dataset (Brain)	*WGAN*	**134.42**

Also, Figure 6.3 captures the correlation heatmaps of the features of the real and synthetic samples. Observing carefully, the correlation heatmaps look similar. Given the frame of reference and the heatmap, our WGAN generator is able to do a decent job at estimating the real WMH lesion image distribution.

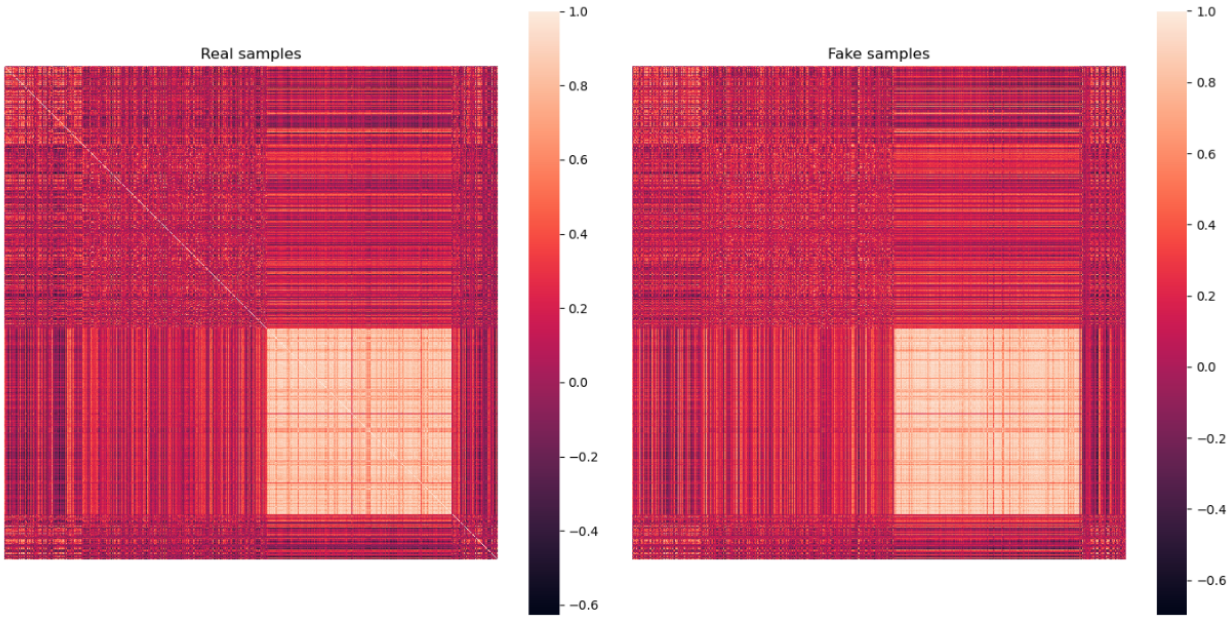

Figure 6.3 - Correlation Heatmaps of the features of real and fake samples.

Let's now see the images generated by the WGAN.

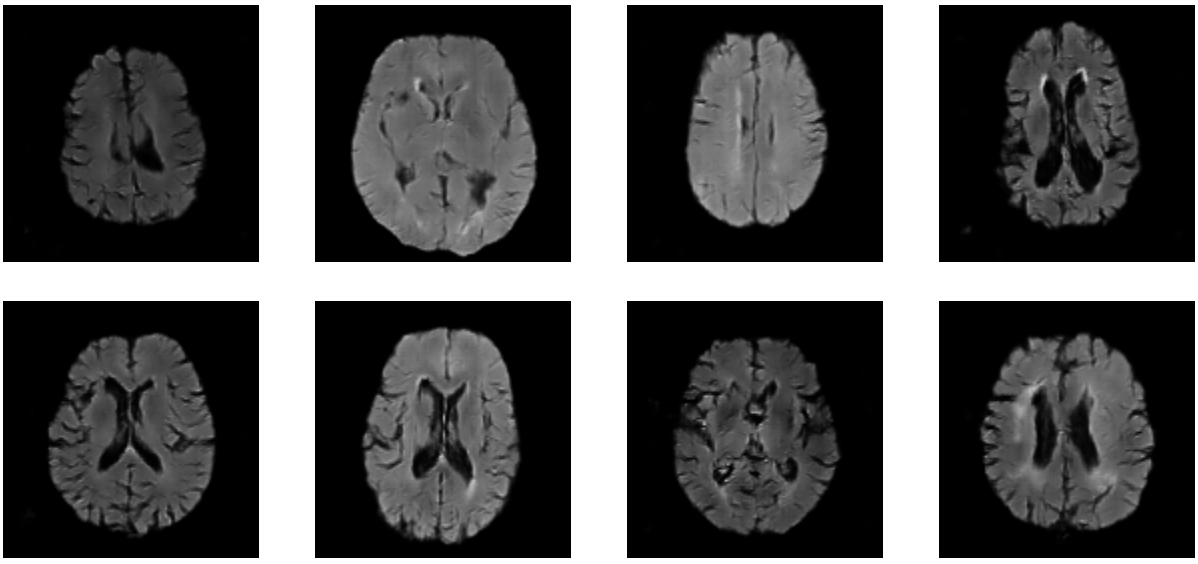

Figure 6.4 - Generated Scans after 900 epochs.

Chapter 7: Conclusion and Future Scope

7.1. Conclusion

Medical image segmentation is a very critical task and needs to be precise. The deep learning based methods for segmentation are not very well received and are still in development for medical use cases. Usually, the networks are trained with retrospective or historical data and therefore, they fail to generalize well when applied to prospective or real-world data. No model is the best for all kinds of data. This leads to a myriad number of models coming up drawing inspiration from each other and reusing concepts from other networks. Often, we tend to go towards state-of-the-art or heavier models to get our job done. But, this isn't always the case. Sometimes, lightweight models do the job well enough.
Learning the distribution of WMH lesions in brain MRI scans needs a lot of computing resources and also training data. The newer and heavier architectures that do perform better than WGAN are very tough to train as they are computationally expensive to train. In addition, FID as a metric has its own drawbacks and sometimes needs qualitative human evaluation, possibly by domain experts in this case to evaluate the clinical relevance of the results.

7.2. Future Scope

While this project tries and tests various segmentation networks with different preprocessing techniques and tries to get good scores on the private in-house dataset, there is much to explore here, as well as in the distribution estimation of the WMH lesions. In particular, future works can focus on these areas:-

1. Transformers [20] have changed the world of deep learning in so many ways. Though initially, transformers existed only for the Natural Language Processing (NLP) domain, it has adapted to the Computer Vision world in the form of Vision Transformers (ViT) [21]. Transformer-based segmentation architectures like UNETR, Swin-Unet, etc. could be taken up to get better segmentation results.
2. Pytorch-based libraries like segmentation_models_pytorch, monai, and MedicalZoo could be tried. These libraries come prebuilt with the models, loss functions, and metrics necessary for the task at hand.

3. The network's performance gets bogged down due to no or wrong preprocessing. Identifying correct preprocessing techniques or coming up with novel preprocessing techniques for the task at hand, could be the way forward.
4. GANs (Pix2Pix [22] or CycleGAN [23]) could be used to generate T2-weighted images for the SSSIHMS and WMH Challenge Dataset, which are already available for the Tongji Dataset. The added modality can help the network learn the difference between WMH lesions and other intracranial lesions.
5. Improved GAN architectures, such as StyleGAN2 could be explored to be able to generate realistic (high fidelity) MRI scans and also would be able to learn the distribution better.
6. Explainable AI (XAI) is very important when it comes to clinical usage. The model's decision process should match that of a trained clinician. XAI tools like Neuroscope [24] could be explored to get a deep-dive understanding of the models' decision-making process.

References

1. R. Ghaznawi, M. I. Geerlings, M. Jaarsma-Coes, J. Hendrikse, and J. de Bresser, “Association of White Matter Hyperintensity Markers on MRI and Long-term Risk of Mortality and Ischemic Stroke,” *Neurology*, vol. 96, no. 17, pp. e2172–e2183, Mar. 2021, doi:10.1212/wnl.0000000000011827.
2. J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” *arXiv.org*, Nov. 14, 2014. https://arxiv.org/abs/1411.4038
3. O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *arXiv.org*, May 18, 2015. https://arxiv.org/abs/1505.04597
4. Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation,” *arXiv.org*, Jun. 21, 2016. https://arxiv.org/abs/1606.06650
5. F. Milletari, N. Navab, and S.-A. Ahmadi, “V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation,” *arXiv.org*, Jun. 15, 2016. https://arxiv.org/abs/1606.04797
6. O. Oktay *et al.*, “Attention U-Net: Learning Where to Look for the Pancreas,” *arXiv.org*, Apr. 11, 2018. https://arxiv.org/abs/1804.03999
7. Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++: A Nested U-Net Architecture for Medical Image Segmentation,” *arXiv.org*, Jul. 18, 2018. https://arxiv.org/abs/1807.10165
8. G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” *arXiv.org*, Aug. 25, 2016. https://arxiv.org/abs/1608.06993
9. I. J. Goodfellow *et al.*, “Generative Adversarial Networks,” *arXiv.org*, Jun. 10, 2014. https://arxiv.org/abs/1406.2661
10. A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” *arXiv.org*, Nov. 19, 2015. https://arxiv.org/abs/1511.06434
11. M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” *arXiv.org*, Jan. 26, 2017. https://arxiv.org/abs/1701.07875
12. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved Training of Wasserstein GANs,” *arXiv.org*, Mar. 31, 2017. https://arxiv.org/abs/1704.00028
13. H. J. Kuijf *et al.*, “Standardized Assessment of Automatic Segmentation of White Matter Hyperintensities and Results of the WMH Segmentation Challenge,” *arXiv.org*, Apr. 01, 2019. https://arxiv.org/abs/1904.00682
14. W. Zhu *et al.*, “Automatic segmentation of white matter hyperintensities in routine

clinical brain MRI by 2D VB-Net: A large-scale study," *Frontiers in Aging Neuroscience*, vol. 14, Jul. 2022, doi: 10.3389/fnagi.2022.915009.
15. H. Li *et al.*, "Fully Convolutional Network Ensembles for White Matter Hyperintensities Segmentation in MR Images," *arXiv.org*, Feb. 14, 2018. https://arxiv.org/abs/1802.05203
16. C. Szegedy *et al.*, "Going Deeper with Convolutions," *arXiv.org*, Sep. 17, 2014. https://arxiv.org/abs/1409.4842
17. M. Lee and J. Seok, "Controllable Generative Adversarial Network," *arXiv.org*, Aug. 02, 2017. https://arxiv.org/abs/1708.00598
18. M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," *arXiv.org*, Jun. 26, 2017. https://arxiv.org/abs/1706.08500
19. Y. Skandarani, P.-M. Jodoin, and A. Lalande, "GANs for Medical Image Synthesis: An Empirical Study," *arXiv.org*, May 11, 2021. https://arxiv.org/abs/2105.05318
20. A. Vaswani *et al.*, "Attention Is All You Need," *arXiv.org*, Jun. 12, 2017. https://arxiv.org/abs/1706.03762
21. A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv.org*, Oct. 22, 2020. https://arxiv.org/abs/2010.11929
22. P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," *arXiv.org*, Nov. 21, 2016. https://arxiv.org/abs/1611.07004
23. J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," *arXiv.org*, Mar. 30, 2017. https://arxiv.org/abs/1703.10593
24. C. Schorr, P. Goodarzi, F. Chen, and T. Dahmen, "Neuroscope: An Explainable AI Toolbox for Semantic Segmentation and Image Classification of Convolutional Neural Nets," *Applied Sciences*, vol. 11, no. 5, p. 2199, Mar. 2021, doi: 10.3390/app11052199.
25. Ciresan, Dan C., Alessandro Giusti, Luca Maria Gambardella and Jürgen Schmidhuber. "Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images." *NIPS* (2012).