

MULTIPLE SCLEROSIS DETECTION USING DEEP LEARNING TECHNIQUES

By

Samikshya Adhikari

A Thesis Submitted to the Faculty
of Southeastern Louisiana University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Integrated Science and Technology

Southeastern Louisiana University
Hammond, Louisiana

May 2025

Copyright by
Samikshya Adhikari
2025

MULTIPLE SCLEROSIS DETECTION USING DEEP LEARNING TECHNIQUES

By

Samikshya Adhikari

Approved:



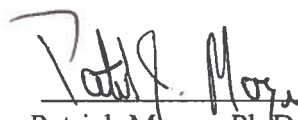
Kazim Sekeroglu, Ph.D.
Associate Professor of Computer Science
(Director of Thesis)



Paulo Alexandre Regis, Ph.D.
Assistant Professor of Computer Science
(Committee Member)



Ghassan Alkadi, Ph.D.
Professor of Computer Science
(Committee Member)



Patrick Moyer, Ph.D.
Interim Dean
College of Science and Technology

Name: Samikshya Adhikari

Previous Degree: B.E., Tribhuvan University, 2023 (Computer Engineering)

Date of Current Degree: May 17, 2025

Institution: Southeastern Louisiana University

Major Field: Integrated Science and Technology (Data Science)

Major Professor: Dr. Kazim Sekeroglu

Title of Study: MULTIPLE SCLEROSIS DETECTION USING DEEP LEARNING
TECHNIQUES

Pages in Study: 67

Candidate for Degree of Master of Science

In health care, deep learning algorithms have the potential to transform patient outcomes, improve treatment quality, enhance the delivery of health care and most of all make the healthcare sector more accessible and cost effective. Applications of deep learning are expanding quickly, ranging from early disease diagnosis to predictive analytics and personalized medicine. This is due to the development of predictive healthcare models in recent years that can precisely spot trends, patterns, and subtle abnormalities that even the most proficient professional may miss. However, despite the quick advancements in machine learning applications for health care, there is very little study that is being done in the field of multiple sclerosis disease detection. So, this thesis aims to address this challenge by accurately detecting multiple sclerosis lesions in 2D MRI images using deep learning models. This research explores various deep learning architectures, including a Convolutional Neural Network (CNN) built from scratch, a ResNet model trained from

scratch, and a Vision Transformer (ViT) also trained from scratch. A comparative analysis was conducted among these models to evaluate their performance in lesion detection. The best-performing model was CNN from scratch achieving an accuracy of 98.06%, while the Vision Transformer exhibited the lowest accuracy at 91.34%.

Keywords: Deep learning, predictive healthcare models, multiple sclerosis lesions, transfer learning

DEDICATION

I would like to dedicate my work to my beloved family and friends whose encouragement and understanding have been my strength throughout this journey. I am especially grateful to my advisor Dr. Kazim Sekeroglu for his constant support and guidance.

TABLE OF CONTENTS

DEDICATION	v
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER	
I. INTRODUCTION	2
II. DEFINITION OF TERMS	6
III. LITERATURE REVIEW	12
IV. DATASETS	18
V. METHODOLOGY	23
VI. RESULTS AND ANALYSIS	32
VII. CONCLUSION AND FUTURE WORK	61
REFERENCES	62

LIST OF TABLES

Table 1: Model comparison table (Patches).....	42
Table 2: Model comparison table (Slices)	52
Table 3: Comparison table for CNN (Patches vs Slices)	54
Table 4: Comparison table for Resnet (Patches vs Slices)	55
Table 5: Comparison table for ViT (Patches vs Slices)	57
Table 6: Comparison of our method with other literature reviews	60

LIST OF FIGURES

Figure 1: Artificial Neural Network [27].....	7
Figure 2: Convolutional Neural Network [28].....	8
Figure 3: Transfer Learning [29]	9
Figure 4: Resnet (Residual Network) [30].....	10
Figure 5: Vision Transformer (ViT) [31].....	11
Figure 6: Sample patch based 0 label images	20
Figure 7: Sample patch based 1 label images	21
Figure 8: Sample slice based 0 label images	21
Figure 9: Sample slice based 1 label images	22
Figure 10: CNN architecture for patch images	26
Figure 11: CNN architecture for slice images	27
Figure 12: Resnet architecture for patch and slice based images	28
Figure 13: ViT architecture for patch images	30
Figure 14: ViT architecture for slice images	31
Figure 15: Classification report for CNN (Patches)	32
Figure 16: Confusion Matrix for CNN (Patches)	33
Figure 17: Training vs validation accuracy and loss for CNN (Patches)	34
Figure 18: ROC curve for CNN (Patches).....	35
Figure 19: Classification report for Resnet (Patches)	36
Figure 20: Confusion matrix for Resnet (Patches)	36

Figure 21: Training vs validation accuracy and loss for Resnet (Patches)	37
Figure 22: ROC curve for Resnet (Patches)	38
Figure 23: Classification Report for ViT (Patches)	39
Figure 24: Confusion matrix for ViT (Patches).....	40
Figure 25: Training vs validation accuracy and loss for ViT (Patches)	40
Figure 26: ROC curve for ViT (Patches).....	41
Figure 27: Model performance comparison graph (Patches).....	42
Figure 28: Classification report for CNN (Slices)	44
Figure 29: Confusion matrix for CNN (Slices).....	45
Figure 30: Training vs validation accuracy and loss curve for CNN (Patches).....	45
Figure 31: ROC curve for CNN (Slices).....	46
Figure 32: Classification Report for Resnet (Slices)	47
Figure 33: Confusion matrix for Resnet (Slices)	48
Figure 34: Training vs validation accuracy and loss curve for Resnet (Slices).....	48
Figure 35: ROC curve for Resnet (Slices)	49
Figure 36: Classification report for ViT (Slices)	50
Figure 37: Confusion matrix for ViT (Slices).....	50
Figure 38: Training vs validation accuracy and loss curve for ViT (Slices)	51
Figure 39: ROC curve for ViT (Slices).....	51
Figure 40: Model performance comparison graph (Slices)	52
Figure 41: Comparison graph for CNN (Patches vs Slices)	53
Figure 42: Comparison graph for Resnet (Patches vs Slices).....	55
Figure 43: Comparison graph for ViT (Patches vs Slices)	56

CHAPTER I

INTRODUCTION

Multiple sclerosis (MS) is a demyelinating disease of the central nervous system (CNS) characterized by damage to the protective myelin surrounding the nerve fibers within the brain and spinal cord [2]. Myelin is a protective covering around neurons in the brain and spinal cord. It transmits signals from the brain and the rest of the body to control functions like vision, sensation and movement. The cause of MS is still unknown. There are still unproven theories about the causes of MS like environmental allergies, exposure to household pets, lack of vitamin D, smoking, obesity etc.

Since MS lesions vary in different shape, size, location and anatomical characteristics among individuals, accurately detecting lesions in both brain and spinal cord is a challenging task. Symptoms often overlap with other conditions, such as small vessel disease or neuromyelitis optica, complicating diagnosis. As MS does not have a single definitive test and traditional diagnostic approaches often rely on overlapping symptoms, early detection is very difficult.

MRI is a suitable technology for the diagnosis and prognosis of these disorders and monitoring effects of the treatment [3]. Different traditional machine learning approaches have been widely studied for MS diagnosis. These methods rely on feature engineering where features are extracted from MRI scans and then classified using conventional ML models. The most popular traditional machine learning approaches are Support vector machine (SVM), K-Nearest Neighbors (KNN) and Random Forest (RF). Despite the wide potential applications of these traditional models, all available algorithms have inherent

limitations. These models struggle to capture complex spatial and temporal relationships and localize features in MRI images. Traditional Machine learning models require handcrafted feature extraction, and they don't scale well with large datasets which require extensive preprocessing. Instead, compared to other ML algorithms, DL algorithms do not suffer from these limitations, as they eliminate the need for feature engineering by trying to learn the optimal set of features from data and they are bound to identify new and unexpected hidden data properties, which is a new and exciting research field [7]. This ability is particularly crucial in MS diagnosis, where lesions vary in size, shape, and intensity, making them difficult to detect using conventional methods. Deep learning models, particularly CNNs, transfer learning and transformer based models, allows for efficient feature extraction and improved model performance even with limited datasets which demonstrate superior performance by capturing intricate spatial relationships and distinguishing subtle abnormalities with high accuracy.

Convolutional neural networks (CNN), a deep learning method, can detect lesions early by learning patterns on brain magnetic resonance images (MRI). They have the ability to learn spatial hierarchies through convolutional layers. CNNs use convolution filters to extract relevant features from an image, automatically identifying subtle details within the lesion that may be difficult for the human eye to detect, like borders, color variations, and textures. So, training a CNN from scratch allows the model to learn MS specific patterns which requires large datasets and computational resources.

Transfer learning is a new and highly effective technique in deep learning which has already been trained in a large dataset called ImageNet. So, this approach starts with models which are pre trained on millions of images and which eventually fine tune on MS

specific data. Traditional deep learning approaches require labeled datasets and long training time to learn useful features from scratch. There are pretrained models like VGG, ResNet, Inception, DenseNet which have already learned hierarchical features like edges, texture, complex patterns. By using these features, transfer learning is faster and improves the model performance even with a limited number of datasets. This approach is really promising in the medical field where rapid detection of MS lesions is essential.

Transformer based models, especially Vision Transformers, are emerging as new alternatives to CNN and transfer learning models. Transformers process the images relying upon self-attention mechanisms. This mechanism allows for long-range dependencies and contextual information which are important for capturing complex patterns in MRI images. ViTs have been shown to perform competitively with CNNs on large-scale image classification tasks [13].

In conclusion, all three models have their own uniqueness in finding the multiple sclerosis lesion by analyzing MRI images. This study aims to research all three deep learning architectures, analyzing their effectiveness on MS datasets and finally evaluating their performance by improving the diagnostic capabilities.

Motivation

The primary goal of this project is to use computational techniques such as deep learning and medical imaging analysis for the diagnosis of multiple sclerosis (MS). This research also aims to work with patch-based datasets to capture localized characteristics of lesions where slice-based strategy often fails. This research seeks to identify the most effective approach between three models for both slices based and patch-based datasets.

So, the ultimate goal is to improve lesion localization and contribute to the development of robust intelligent systems.

CHAPTER II

DEFINITION OF TERMS

Deep learning is an artificial intelligence (AI) method that teaches computers to process data in a way inspired by the human brain [14]. Millions of neurons interconnect to form a human brain where new information is learned and processed. Similarly, artificial neurons also called nodes interpret the data using computational methods. The components of a deep neural network are the input layer, hidden layer and output layer. The input layer takes the data which passes into the hidden layer. The hidden layer processes the information at different levels and transforms the data using nonlinear functions. Deep learning networks have hundreds of hidden layers that they can use to analyze a problem from several different angles. The final output layer generates the model's prediction.

Artificial neurons are the fundamental units of deep learning models. Artificial neurons are also called perceptions. Firstly, a neuron receives input data. The input data can be raw numbers or pixel values or others depending upon the specific problem being solved. Each input has their weight which is the importance of that input. Inputs are then multiplied by their weights and summed up along with a bias term. A bias term is a parameter in a neural network that helps to shift the activation function. This helps to improve the model's learning capabilities and patterns. The weighted sum is passed through an activation function to introduce nonlinearity. The training process of deep learning models involves passing the data through the network layer by layer where each neuron calculates the output and passes it to the next layer. This process is called forward propagation. A loss function is used to compare the predicted output with the actual value,

eventually measuring the model's error. The minimization of error is done through the process called backward propagation. An optimization algorithm is used to update the model's weight. Using gradient descent, weights and biases are updated to minimize the error. The process is repeated for multiple iterations until the model achieves good accuracy. After training the data, the model can take new and unseen information to generate the predictions.

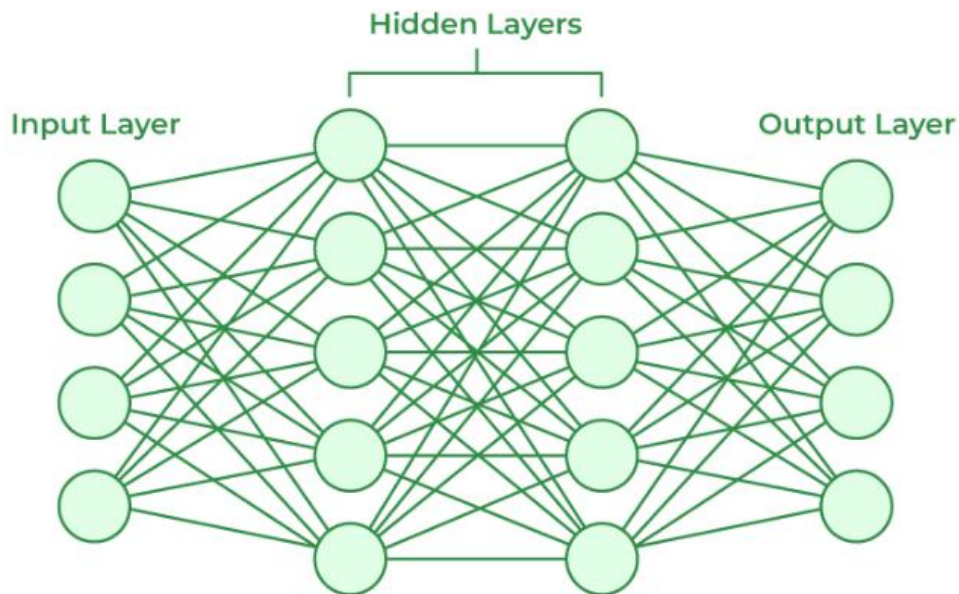


Figure 1: Artificial Neural Network [27]

CNN (Convolutional Neural Network) is one of the deep learning neural network architectures which is used to extract the feature from the grid-like matrix dataset. Convolutional layer is the first layer of convolutional neural network which is also considered as the core building block of CNN. This layer applies convolution operations

to extract local features from the input dataset. The convolution applies a sliding window function to a matrix of pixels representing an image. This sliding function is called filter or kernel. The applied filter is used to find specific patterns, shapes, edges, texture from an image. These filters slide over the images performing multiplications which are followed by summations to create a feature map. A RELU activation function is applied after each convolution operation to help the network learn non-linear relationships between features in an image. After this, a pooling layer is applied to pull the most important features from the matrix. As the network goes deeper, more complex patterns are captured. After the feature extraction process is complete, it goes through the last layer of CNN which is fully connected layers. These layers get a flattened one-dimensional matrix, and it performs the task of classification based on the features extracted from the previous layers. Finally, the final layer uses sigmoid or softmax activation function in classification tasks which generates the probability for each class.

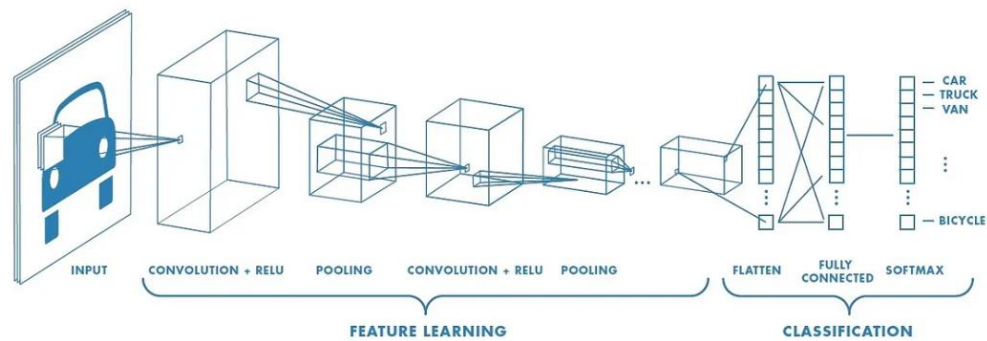


Figure 2: Convolutional Neural Network [28]

Transfer learning is another machine learning technique that uses pre-trained models rather than training the model from scratch. In other words, transfer learning uses what has been learned in one setting to improve generalization in another setting [15]. The traditional setting involves building models from scratch based on datasets where no knowledge is retained, and learning occurs independently. On the other hand, transfer learning uses an already trained model as a starting point for training where the model's knowledge is applied for a new task. This process involves selecting a well-trained model like VGG, ResNet, Inception etc. These models have already been trained in large scale datasets which are called ImageNet. This dataset contains over 14 million labeled images across 1000 categories. The lower layers capturing features like edges and texture are usually frozen to retain their learned knowledge. And the higher layers recognizing the task related patterns are fine-tuned. So, this method deals with difficulty related to acquiring larger datasets. Also, it helps to reduce model training time which eventually helps to improve performance.

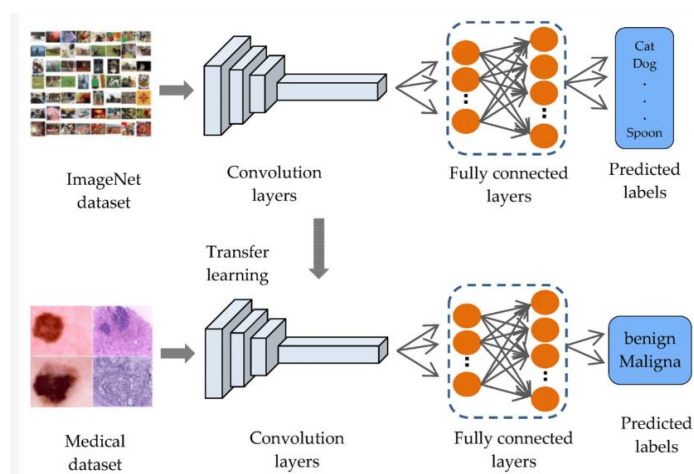


Figure 3: Transfer Learning [29]

A residual neural network also known as ResNet is a deep learning architecture which is specifically used to solve vanishing gradient problems in a neural network. This concept is called residual blocks. A technique called skip connections is used. The skip connection connects activations of a layer to further layers by skipping some layers in between. This forms a residual block. Resnets are made by stacking these residual blocks together [16]. The expression is called $H(x) = F(x) + x$, where $F(x)$ represents the residual function learned by the network and x is the input which represents identity mapping. ResNet architectures utilize bottleneck blocks. This uses $1 * 1$ convolutions to reduce computational costs. Among Resnet, ResNet50 remains the most popular for the real-world applications.

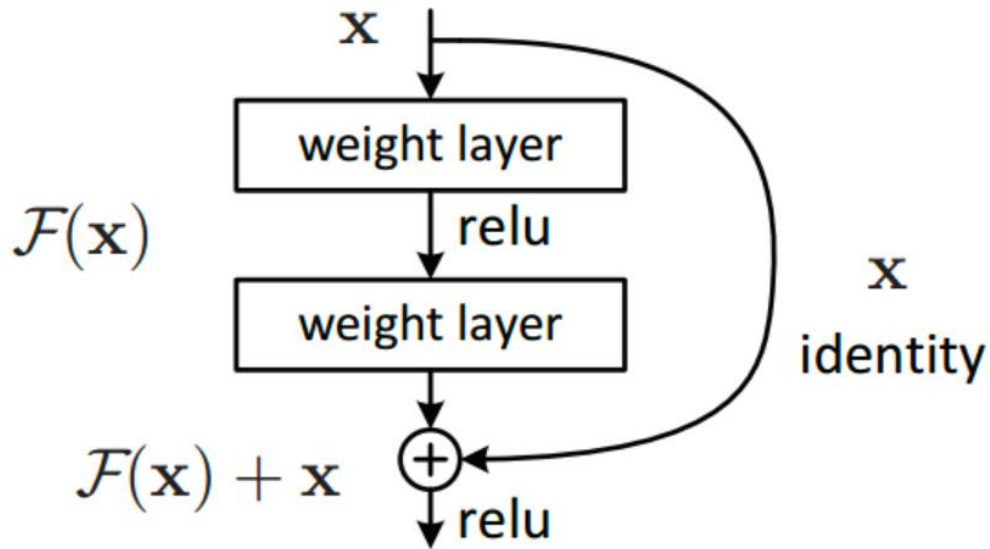


Figure 4: Resnet (Residual Network) [30]

Transformer is a deep learning architecture which applies self-attention mechanisms, differentially weighing significance of each part of the input data. Vision transformers (ViT) split an image into a number of small fixed size patches. The patches are then flattened and projected into a D-dimensional embedding space using a linear layer to generate lower-dimensional embeddings. The transformers require positional embeddings to gain spatial relationships between patches. The patch embeddings which are combined with positional embeddings which are then fed into a transformer model which applies multi head self-attention mechanisms to capture global relationships between patches. This approach allows models to model long range dependencies, making them effective in tasks like image classification, object detection and segmentation. This model has the ability to scale with large datasets which provide good performance when trained with sufficient data.

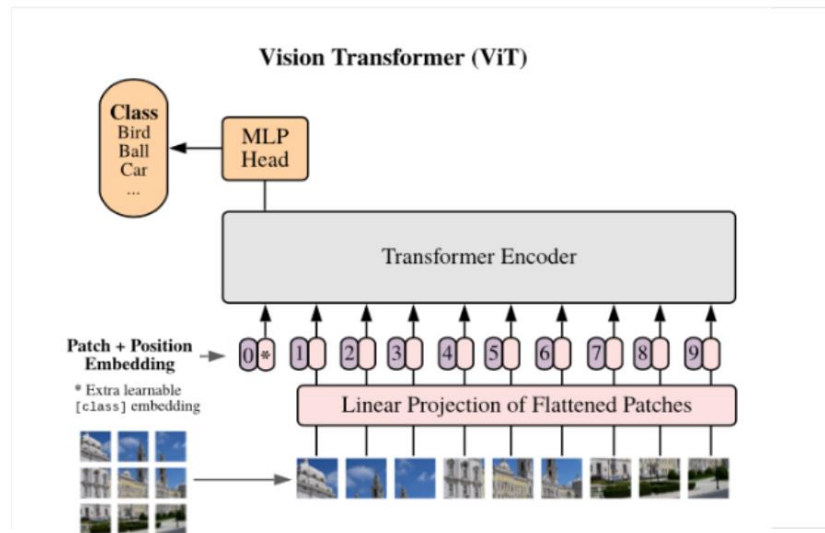


Figure 5: Vision Transformer (ViT) [31]

CHAPTER III

LITERATURE REVIEW

In the paper “Detection of Multiple Sclerosis Using Convolutional Neural Networks: A Comparative Study”, the authors compared the performance of two machine learning techniques, such as Random Forest (RF) and Support Vector Machine (SVM), with a deep learning Convolutional Neural Network (CNN) technique in diagnosing Multiple Sclerosis (MS) using MRI scans [4]. This study emphasizes the significant role of deep learning algorithms, particularly CNNs, in computer vision tasks related to medical imaging. It aims to improve predictive accuracy and assist in clinical diagnosis through effective classification methods.

The paper “Brain MRI based diagnosis of autoimmune diseases using deep learning” focused on diagnosing autoimmune diseases, particularly multiple sclerosis, through brain MRI analysis. It addresses the challenge of accurately interpreting MRI data to identify demyelination areas, which are crucial for diagnosing and monitoring the disease [5]. This paper utilizes CNN architecture for image classification. It extracts hierarchical features from the images through multiple convolutional layers, enabling the model to recognize patterns indicative of demyelination associated with multiple sclerosis.

Khaled et al. [6] introduced an ensemble learning-based approach for the detection of Multiple Sclerosis (MS), integrating deep learning techniques with ensemble methods to enhance diagnostic accuracy and efficiency. Initially, feature extraction is performed using the SWIN Transformer and MobileNetV3-small architectures to analyze MRI images from Kaggle MS dataset. These extracted features are then classified using ensemble

learning techniques, including CatBoost, XGBoost, and Random Forest, to achieve robust and accurate classification. This combination of feature extraction, ensemble learning, and deep learning techniques represents a comprehensive framework for MS detection [6].

Sabila Al Jannat [8] took on the detection of white matter lesions in 3D MRI to diagnose multiple sclerosis (MS) employing convolutional neural networks and transfer learning approaches for improving accuracy. Their dataset is the MS patient's data from the Laboratory of Imaging Technologies. Their approach is different as they try to classify different types of MS including one class with no MS which requires softmax activation function for training. The authors also experiment with the augmentation technique which is a well-rounded approach towards the dataset imbalance. The other preprocessing technique the authors used was the grayscale image was converted to RGB images to ensure color consistency during the training of convolutional neural networks. This research aims to automate the lesion detection process by reducing the time associated with clinical analysis.

Minhoe Kim [9] analyzed the link between white matter tract density index (TDI) and the disability in multiple sclerosis. Segmentation for the lesion was performed especially for the FLAIR MRI images and their TDI was calculated. This paper explored the relationship between EDSS (Expanded disability status scale) and TDI. This research suggested that the MS lesions with high white matter tract density are related to severe physical disability. For datasets they use the largest publicly available MS patient dataset which are obtained through manual lesion segmentation. They performed the TDI calculation which also supported the idea that the location of lesions rather than their size was crucial for disability in MS.

The paper, ‘BrainMorph: A Foundational Keypoint Model for Robust and Flexible Brain MRI Registration’ [10] introduced a tool for aligning brain images which handles multimodal, rigid, affine, and nonlinear transformations for both skull stripped and non skull stripped data. The main aim of this paper is for the brain MRI registration using keypoint detection. Image registration involves aligning two or more images of the same anatomical region obtained from different modalities or at different times. They performed training on 100000 3D volumes from 16000 subjects. The paper proposed a novel algorithm for groupwise registration based on detected keypoints. Their results indicate improved registration, especially in longitudinal studies.

Nada Haj Messaoud [2] takes an approach to segment out 7 classes of brain regions including Frontal Lobe, Occipital Lobe, Parietal Lobe, Temporal Lobe, Brain Stem, Cerebellum, and background with an unlimited number of slices for each patient. Their approach utilized the Unet inspired model to segment the lobes for the public and private datasets. This model extracted over 7200 features from MS lesions including surface, localization and the number of lesions. The performed segmented approach achieved a mean IoU score of 0.70. They also performed the Concat-U-Net method to segment the MS lesions. However, this model isn’t elaborated on. The authors also note the challenges they faced during segmentation which are variability in brain shapes and sizes and limited number of 2D slices. In terms of data, they combine imaging data with clinical data to predict disability levels in MS patients which is a different approach.

In the paper ‘Advances in differential diagnosis of cerebrovascular diseases in magnetic resonance imaging: a narrative review’ author Xin Li explores MRI’s role in diagnosing cerebrovascular disease (CVD). The paper discusses that multiple sclerosis

(MS) can cause various cerebral symptoms, including ophthalmoplegia, due to demyelinating lesions [12]. MRI that uses a large magnetic field of 7 tesla (T) is used to detect changes in myelin water and interstitial water in MS patients. In addition to that, this paper also notes that MS can lead to ataxic hemiparesis, characterized by specific MRI findings such as low-signal-intensity lesions in the internal capsule.

Ekmekyapar and Taşcı [17] developed an exemplar based deep learning AI model for multiple sclerosis detection using MobileNetV2 architecture. This study uses a different approach of converting images into smaller 28 by 28 patches which have improved data processing and feature extraction. The authors selected the most meaningful features using a technique called ImRMR and used KNN for classification of lesions. This method used by the authors outperformed several existing models in terms of accuracy and F1 score. The authors experimented with limited datasets, so they have emphasized the need for diverse datasets.

The paper, ‘Detection of multiple sclerosis using advanced deep learning strategies’ [18] focused on unexplored regions which are deep grey matter and mesial temporal lobe for the detection of MS lesions. So, the authors addressed the gap by studying a less commonly researched area which is often neglected. This paper proposed GLCM based feature extraction which can help to reduce overfitting. They performed the combination of GLCM with pretrained CNN models like ResNet, DenseNet and ShuffleNet within the ensemble framework. This paper has proved that ensemble deep learning methods have good potential in detecting MS lesions.

Siar and Teshnehlalab [19] proposed CNN based architecture for diagnosing and classification of Multiple sclerosis lesions, brain tumor and healthy dataset using MRI

images. Unlike other studies which focus only on MS lesions or brain tumors, this study focuses on MS and brain tumor classification which often appear similar on MRI images. So, the goal of this paper is to help reduce misclassification by using a deep CNN model. Twenty-five layers CNN was used along with a softmax classifier to classify three categories. This paper provided accuracy of 96.88% which has shown improvement in classification tasks.

Sujatha Krishnamoorthy [20] proposed an automatic intelligent system using medical of things for multiple sclerosis detection. The paper uses pretrained transfer learning models for detecting MS lesions along with hand crafted feature extraction methods using Local Binary Pattern (LBP) and Pyramid histogram of oriented gradients (PHOG). This combination of deep and hand-crafted features is unique, and this hybrid method improves detection. The authors proposed the Brownian firefly algorithm (BFA) to support feature optimization and avoid overfitting problems. Another contribution of this paper is the model tested on both MRI slices with and without skull sections.

The paper ‘Computational classifiers for predicting the short-term course of Multiple sclerosis’ [21] explored different machine learning classifiers like NNets, Bayesian models, logistic regression and decision trees to predict short term progress of multiple sclerosis. The paper uses a different strategy of combining Neural networks with predictors like EDSS, lesion volume on MRI. The main purpose of this paper was to address the challenge of prognostic predictions of MS lesions on each level which can be useful for the future. So, this paper shows the importance of machine learning in feature fusion to predict the progression in MS disease.

Eitel et al. [22] introduced a new layer type called patch individual filter (PIF) which trains higher level filters within predefined patches. This new approach used by the authors has outperformed a baseline CNN in small datasets by improving accuracy from 75.04% (baseline CNN) to 80.92% (PIF based CNN).

Gulay Macin [23] proposed a machine learning model called Exemplar Multiple Parameters Local Phase Quantization (ExMPLPQ) for MS detection. Along with this model feature selection is done using LPQ and Iterative neighbourhood component analysis (INCA) which are classified using Shallow k nearest neighbour (KNN) algorithm. This approach used by authors has outperformed 19 different pre-trained deep learning models.

CHAPTER IV

DATASETS

The datasets used in this study includes publicly available T1-weighted, T2 weighted, and FLAIR (fluid-attenuated inversion recovery) images from patients at MS-Clinic, Baghdad Teaching Hospital, Medical City Complex, Baghdad, Iraq. It consists of MRI data from 60 patients diagnosed with MS. The scans were conducted between 2019 and 2020 using a 1.5 Tesla scanner across 20 different medical centers. This dataset also includes manual lesion segmentation for all MRI images which were validated by three experts. Segmentation and validation of the manual lesion segmentation are done by three experts, two radiologists and one neurologist in slice-by-slice manner for three MRI sequences T1-w, T2-w and FLAIR [1]. All the images are in NIFTI format which is the file format usually used for neuroimaging.

Data Preprocessing

The publicly available dataset used in this research was extracted from the web. The dataset contained 60 folders with each patient's image data. Inside each folder, there were T1 weighted, T2 weighted and Flair images along with T1 segmented, T2 segmented and Flair segmented images. For this project, the focus was only on FLAIR and FLAIR segmented images which have been more effective in highlighting multiple sclerosis lesions. The 3D data contained axial, sagittal and coronal views. All these standard anatomical planes were used for this research. Initially, 3D volumetric MRI images were converted into individual 2D slices of images for each patient. Since the 3D images were in NIFTI format, a python library Nibabel was used for the conversion. After the

conversion, the 2D images were saved in .jpg format for further processing. Two separate folders were maintained for each patient, where one contained the original FLAIR slices and other contained their corresponding manually segmented lesion masks. In order to apply a sliding window approach for patch extraction, it was crucial to find the size of the largest lesion across the dataset. The largest lesion was detected to calculate the proper patch size that ensures full lesion coverage. All the segmented slices were visualized, and the lesion slices were analyzed manually. OpenCV was used to process the segmented mask and find the largest lesion. Using connected component analysis, distinct lesions were detected, and their areas were calculated. After the largest lesion was found, a bounding box was created around the largest lesion to know its dimension. The dimension of the largest lesion was found to be $146 * 81$ pixels. This dimension served as a reference for the size of a sliding window.

A sliding window with the same dimension of largest lesion along with the stride size of one fourth was applied to each segmented image. The goal of applying a sliding window to the segmented image was to identify and isolate the regions containing the lesions. As the sliding window moved across each segmented image, it generated patches of the fixed size $146 * 81$ with the defined stride. For each window position, the corresponding patch from the segmented image (i.e. the lesion mask) was analyzed to determine whether it contained a lesion. If a lesion was present within a patch from the segmented mask, the exact same spatial region was extracted from the original Flair images. So, the new dataset containing the patches image was created. Each patch was labeled based on the presence or absence of the lesion.

The classification process described above was applied across the Flair images of all 60 patients in the dataset. After preprocessing, the total patch-based dataset consisted of 248498 image patches where 205545 patches were labeled as non-lesions and 42953 patches were labeled as lesions. This patch level dataset along with slice dataset was used for training and evaluating the lesion detection models.

Also to compare the patch datasets with the slice dataset, all the slice data were resized to a uniform dimension of 465 by 465 pixels. This is required for training because the model expects all input images to be of consistent dimension. Instead of using interpolation-based resizing, zero padding was applied to preserve lesion features. There were 1461 slices of image in total where 664 slices were labeled as non-lesions and 797 slices were labeled as lesions.

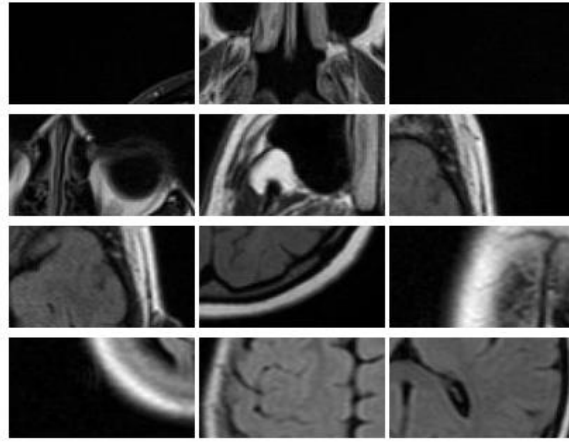


Figure 6: Sample patch based 0 label images

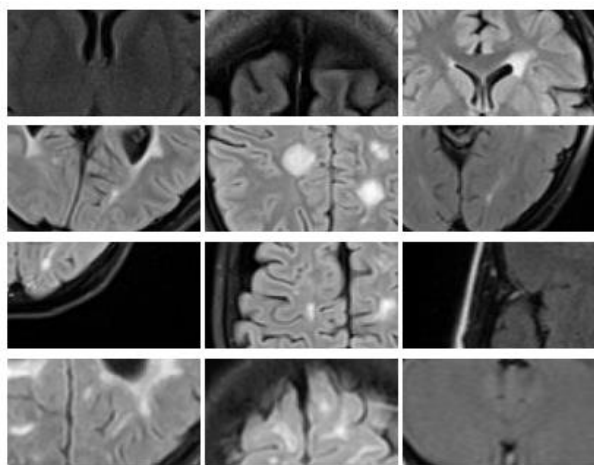


Figure 7: Sample patch based 1 label images

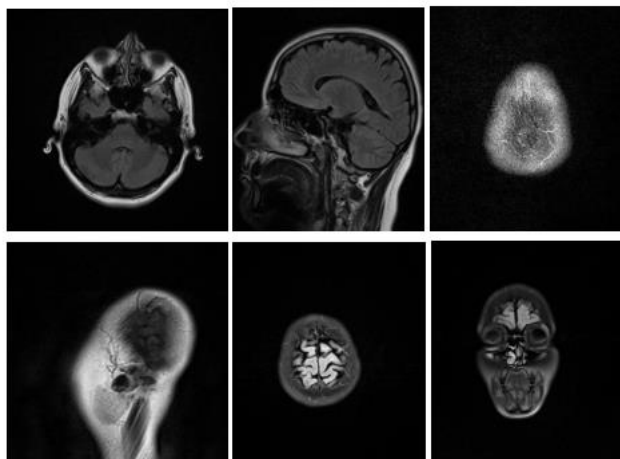


Figure 8: Sample slice based 0 label images

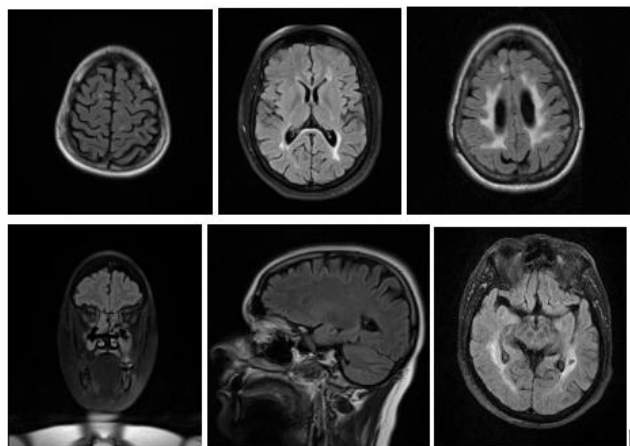


Figure 9: Sample slice based 1 label images

CHAPTER V

METHODOLOGY

Experimental Overview

This study tested the CNN model from scratch, transfer learning technique Resnet from scratch and Vision transformer (ViT) for multiple sclerosis detection. The dataset has MRI images which are grayscale. All of the image patches are 146 by 81 pixels in resolution. The slice images were of 465 by 465 resolution. 80% of the data is divided for training and 20% of data is divided for validation phase using TensorFlow's `image_dataset_from_directory()` function. To optimize the data loading process, AUTOTUNE was used for prefetching the data. This was used to enhance training efficiency. Furthermore, when training machine learning models, each dataset and model need a different set of hyperparameters, which are a kind of variable [24]. To determine the hyperparameters multiple variables must be experimented to fully understand the model's capability. So, hyperparameter tuning is a technique used to pick a set of parameters and run them through the models. The different combinations of variables are explored which can be helpful to avoid overfitting or underfitting and increase the model's performance. Some common hyperparameters that can be tuned are learning rate, batch size, optimizer, activation functions, kernel size, dropout rate and many more. So, this research used hyperparameter tuning to grasp the full potential of the model. Along with hyperparameter tuning, the research used checkpoints and early stopping callbacks from the Tensorflow keras library. Checkpoints were used to save the model's best state during the training based on validation performance. This also helped to prevent loss of progress because training took a lot of time, and this helped to overcome system crashes. Early

stopping was used in order to prevent overfitting and reduce unnecessary training time. During early stopping, validation accuracy was used as performance metrics and training was stopped if there was no improvement for 5 consecutive epochs.

CNN architecture

The research used CNN models from scratch and included hyperparameter tuning using Keras Tuner. The CNN model consisted of three main convolutional blocks which was followed by a fully connected layer and an output layer. Each block consists of multiple Conv2D layers along with the relu activation function followed by batch normalization, which is used to stabilize the training process, Max pooling of pool size 2 by 2 was used to down sample the input along with its spatial dimension and a dropout which is used to prevent overfitting by deactivating neurons during training. The tunable parameters used in the model were convolutional filters, kernel size, dropout rates across different layers, dense layers, L2 regularization strength and learning rate for adam optimizer. The first block included two convolutional layers where filter size of 32, 64 and 128 and kernel size of 3 and 5 were tuned. The second block included two convolutional layers where filter size of 64, 128 and 256 was tuned and kernel size of 3 was used. The third block included three convolutional layers where filters ranging from 128 to 512 were tuned and kernel size of 3 was used. The dropout of 0.2 and 0.5 was used to fine tune the model. After the convolutional block, the features were flattened and passed through a fully connected dense layer where neurons of 128, 256 and 512 along with l2 regularization strength of 1e-3, 1e-4, 1e-5 were fine tuned. The model was compiled using Adam optimizer and learning rate was hypertuned between 1e-5 and 1e-3. Binary cross entropy

was also used as the loss function along with metrics such as accuracy, precision, recall, and AUC.

To perform hyperparameter tuning, the model used the RandomSearch method from the keras tuner library. This approach randomly selects the combination of parameters from the defined range. The tuning object was set to val_accuracy ensuring the model with maximum validation accuracy is selected. The max_trials were set to 10 meaning the tuner would explore 3(10) different hyperparameters configurations. Tuner.search() method was initiated to do the hyperparameter tuning. Each trial was run upto 30 epochs along with early stopping.

After the tuning, the CNN architecture was created from scratch using the best parameters obtained during hyperparameter tuning. In the final training, the first convolutional block consisted of two conv2D layers with 128 filters (5*5 kernel) and 32 filters (3*3 kernel) respectively. This was followed by batch normalization and max pooling and dropout rate of 0.4. The second convolutional block consisted of two conv2D layers with 64 filters (3*3 kernel) and 128 filters (3*3 kernel), batch normalization, max pooling and dropout rate of 0.4. The third convolutional block consisted of three conv2D layers where each layers used 256 filters (3 *3 kernel), batch normalization, max pooling and dropout rate of 0.4. The extracted features were flattened, passed through a dense layer of 512 units and L2 regularization of ($\lambda = 0.001$) followed by batch normalization and a dropout of 0.4. Finally, the output layer consisted of a single neuron with sigmoid activation function.

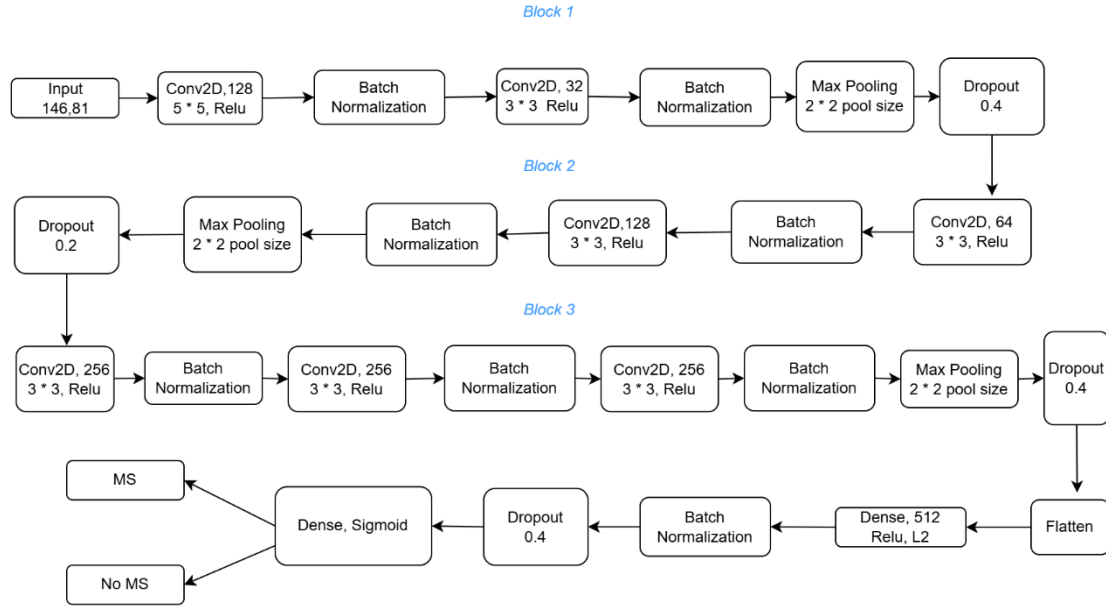


Figure 10: CNN architecture for patch images

A separate CNN architecture was designed for slice MRI images. The architecture consisted of three convolutional layers, each followed by max pooling. The convolutional layers and kernel size were hypertuned using tunable parameters. After flattening, it was passed through dense layers with tunable neurons 64, 128 and 256 and a tunable dropout rate of 0.3 and 0.6. At last, the learning rate was also hyper tuned. The best hyperparameters tuned using this CNN model were: conv1_filters: 16, conv1_kernel: 5, conv2_filters: 128, conv2_kernel: 3, conv3_filters: 128, conv3_kernel: 3, dense_units: 128, dropout_rate: 0.5, lr: 0.0001048345398481006. Based on these parameters the model was rebuilt and trained for 100 epochs.

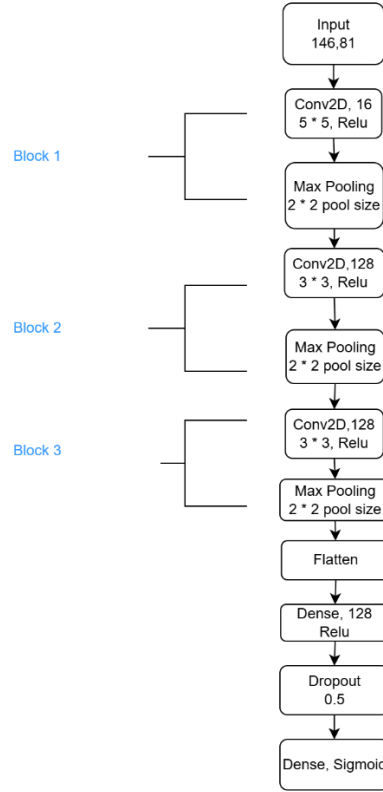


Figure 11: CNN architecture for slice images

Resnet architecture

The next architecture used for hyperparameter tuning was a Resnet Model. This model was used because it allowed for effective training of deeper neural networks. This model comes with residual connections, helps in mitigating gradient problems eventually helping to learn complex features. The resnet model begins with a convolutional layer of 64 filters (7 * 7 kernel) with stride as 2. This was followed by batch normalization and max pooling with pool size (3 * 3). This model used four residual blocks with filters ranging from 64 to 512 with stride as 2 to down sample feature maps. The residual block consists of two convolutional layers, each followed by batch normalization. A shortcut connection

is used in the residual block and if the spatial dimensions or channel depth changes between the input and output of a block, a 1×1 convolution is applied to the shortcut to match dimensions. After the residual block, the model was followed by a global average pooling layer which is used to reduce the number of parameters, preventing overfitting as well. Finally, a dense layer was used which was tuned from 64 to 256 numbers of units. Along with that dropout rate which was also treated as tunable parameters from value 0.3 to 0.7 was used. The final output consists of a single neuron with sigmoid activation for classification.

So, in the resnet model the three key hyperparameters: the number of units in the dense layer, dropout rate and learning rate for Adam optimizer were used. After the hyperparameter tuning, the best performing parameters were 192 units for the dense layer, 0.4 dropout rate and the learning rate was 0.001.

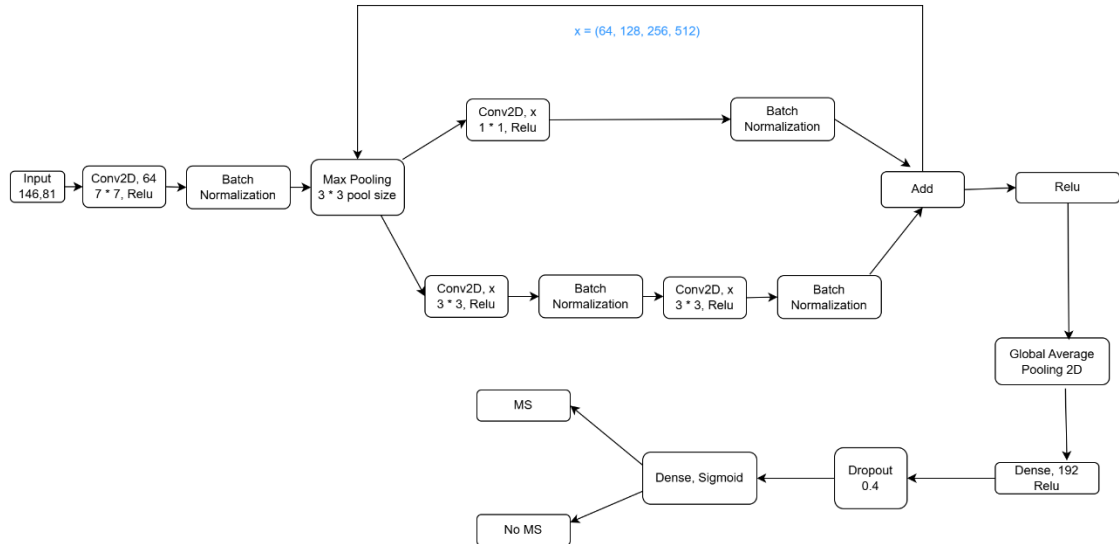


Figure 12: Resnet architecture for patch and slice based images

The same model was applied for the slices image. While hyper parameter tuning for the slices image, the best parameters obtained were 64 units for the dense layer, 0.4 dropout rate and 0.0001 learning rate.

ViT architecture

This research also implemented Vision Transformer (ViT) architecture for binary classification of MRI patch images. An input image of size $146 * 31$ with 3 channels was fed into the model where the model treated the image as smaller non overlapping image patches. A patch size of 13 was used to divide each image into 66 patches. These patches were linearly projected into 32 dimensional vectors using a dense layer. This formed the initial patch embedding layer. After this a technique called positional embedding was used. It is a technique that is used to describe the location or position of an entity in a sequence so that each position is assigned a unique representation [25]. So, the key reason for using positional embeddings was because this introduced spatial information which the transformers lack. So, the positional embeddings were added to the patch embedding layers. These embeddings were passed to five transformer encoders which consisted of 6 heads for multi head self-attention, 32 projectional dimensions, 192 transformer units, 0.15000000000000002 drop out units. All of the parameters for the training of the model were obtained through hyperparameter tuning. There were also the residual connections at two stages within each transformer block. By preserving the original input through residual connections, transformers ensure that gradients can flow directly through the network, alleviating the vanishing gradient problem commonly encountered in deep neural networks. Also, residual connections contribute to smoother optimization during training, enabling transformers to converge faster and achieve better performance. After going

through all the transformer encoders, a final layer normalization was applied with an epsilon value of $1e-6$. This was followed by the global average pooling layer. This aggregates the sequence of embeddings into a single vector. This vector is passed through a dropout layer and then finally it goes through the dense layer which uses sigmoid activation to classify a MS lesion. So, this architecture helps to model long range spatial relationships between patches by leveraging self-attention mechanisms which is useful for MS detection.

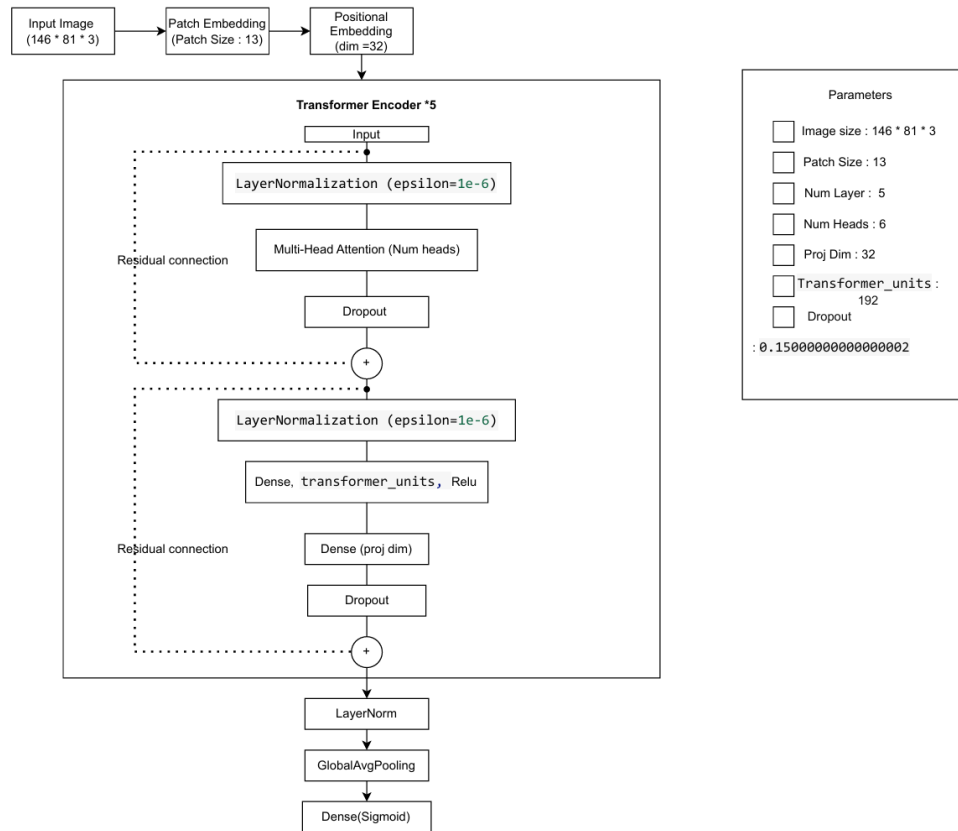


Figure 13: ViT architecture for patch images

ViT was used for slices of images too. The input of size 465 by 465 was divided into non overlapping patches of size 15 by 15 which resulted in 961 patches for each slice. Each patch was flattened and passed through a dense layer that projected into 64-dimensional space. A positional embedding of dimension 64 was added to the patch. The model then looped through 4 transformer encoder blocks. Each block contained layer normalization, Multi head attention with 4 heads and feed forward MLP with dense layers 128 and 64 units. Residual connections were applied too. After the transformer encoder, a global average pooling layer was applied which was followed by a dense layer with 64 units and Relu activation function. Finally, a dense layer with 1 unit was used for classification.

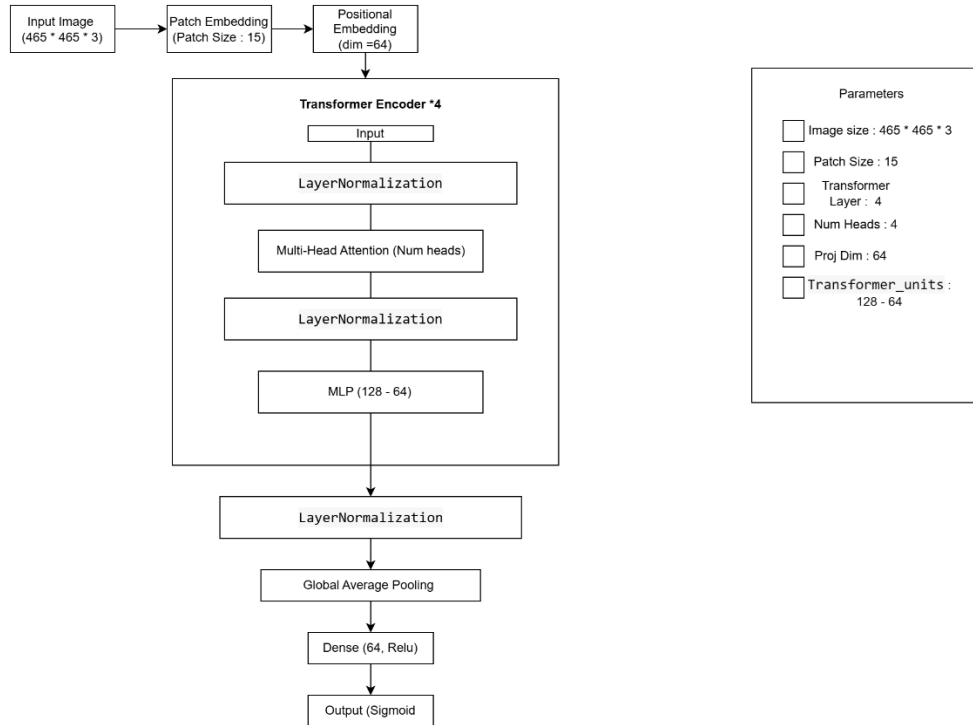


Figure 14: ViT architecture for slice images

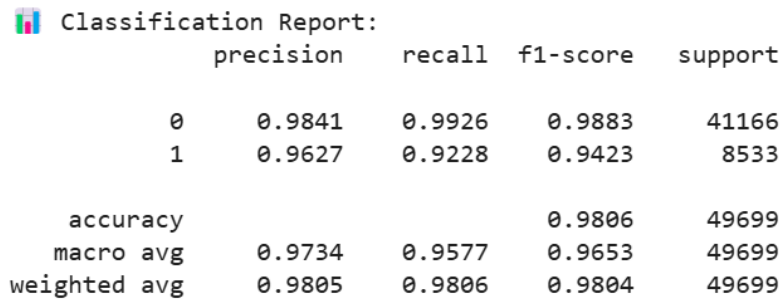
CHAPTER VI

RESULTS AND ANALYSIS

All three models included performance metrics such as accuracy, F1 score, precision and recall for the validation datasets. These metrics were calculated to provide complete assessments of each model's capabilities. Along with them, a confusion matrix, training vs validation accuracy curve, training vs validation loss curve and ROC curve were created.

CNN predictions on patch images

The CNN model achieved strong accuracy on patches images with an overall accuracy of 98.06%. For both lesion and non-lesion patches the model performed excellent precision, recall and f1 score. The macro averaged F1-score of 0.9653 reflects good balance between the classes while the weighted average F1-score of 0.9804 confirms consistent performance despite class imbalance.



Classification Report:				
	precision	recall	f1-score	support
0	0.9841	0.9926	0.9883	41166
1	0.9627	0.9228	0.9423	8533
accuracy			0.9806	49699
macro avg	0.9734	0.9577	0.9653	49699
weighted avg	0.9805	0.9806	0.9804	49699

Figure 15: Classification report for CNN (Patches)

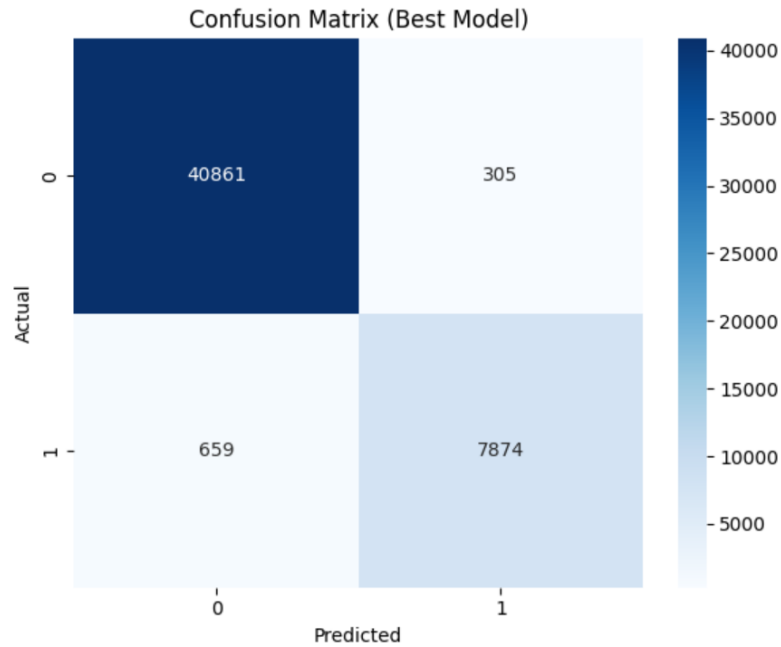


Figure 16: Confusion Matrix for CNN (Patches)

The confusion matrix of CNN model describes its strong classification on patches image. The above results indicate a low false positive and false negative rate, particularly showing the model's ability to accurately detect lesions while maintaining high specificity for non-lesion patches.

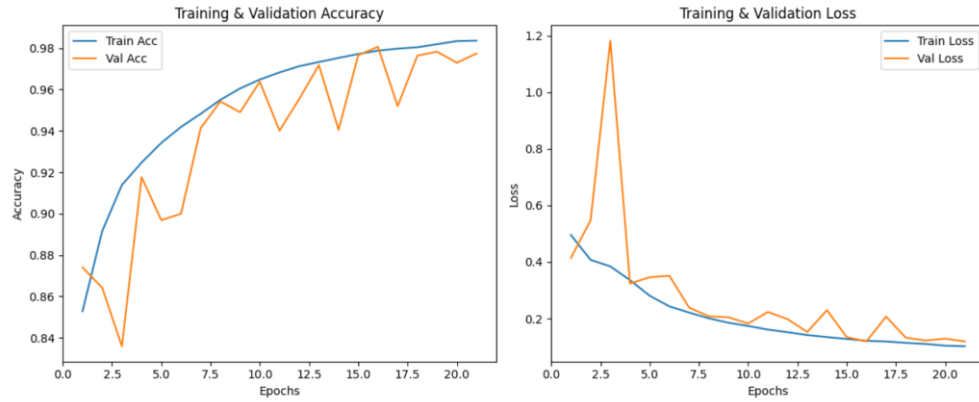


Figure 17: Training vs validation accuracy and loss for CNN (Patches)

The training vs validation accuracy curve describes the increasement in the accuracy over 20 epochs with both converging to the values above 97%. Initially the model was trained for 50 epochs but it early stopped to 20 epochs. It is also noted that the validation accuracy closely tracks training accuracy which shows there might be minimal overfitting.

Similarly, the training vs validation loss curve describes the decrement over 20 epochs. Although there was a spike on epoch 2, it continuously started to decline afterwards.

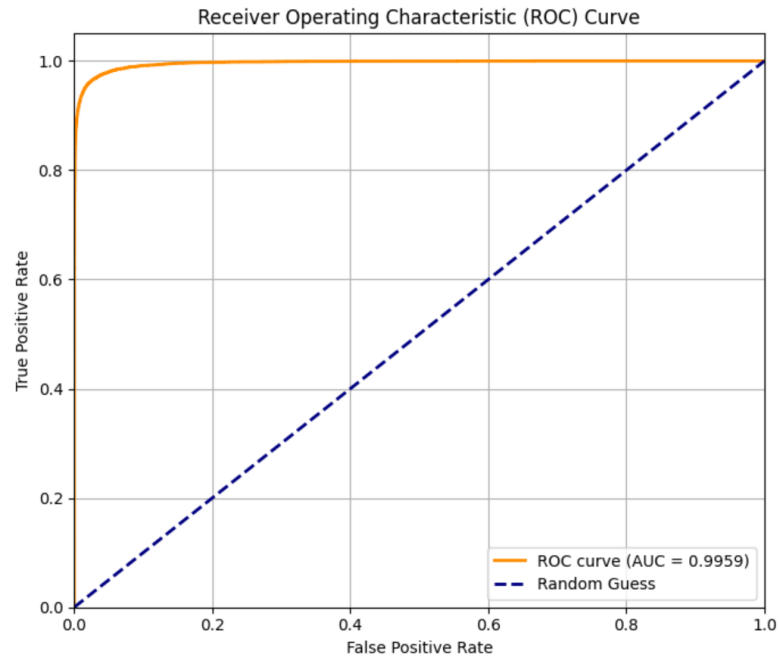


Figure 18: ROC curve for CNN (Patches)

The ROC curve illustrates that the model achieved Area under the curve (AUC) of 0.9959, which is very close to 1. This high value confirms the model's ability to accurately distinguish between positive and negative cases.

Resnet predictions on patch images

Classification Report:				
	precision	recall	f1-score	support
0	0.9597	0.9808	0.9701	41166
1	0.8963	0.8014	0.8462	8533
accuracy			0.9500	49699
macro avg	0.9280	0.8911	0.9082	49699
weighted avg	0.9488	0.9500	0.9489	49699

Figure 19: Classification report for Resnet (Patches)

The Resnet model achieved overall accuracy of 95.00% on patch classification. However, it showed lower recall on lesion patches (0.8014), indicating some missed detections.

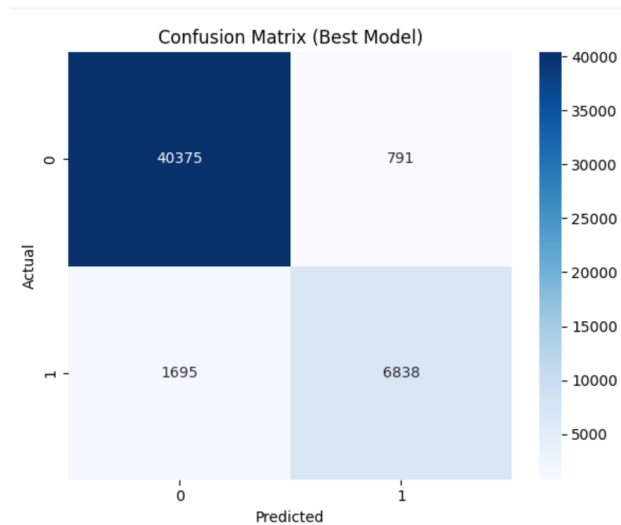


Figure 20: Confusion matrix for Resnet (Patches)

The confusion matrix shows that 40375 non-lesion images and 6838 lesion images were correctly classified to their respective classes. However, 1695 lesion patches were misclassified as non-lesions which might have reflected lower recall for class 1.

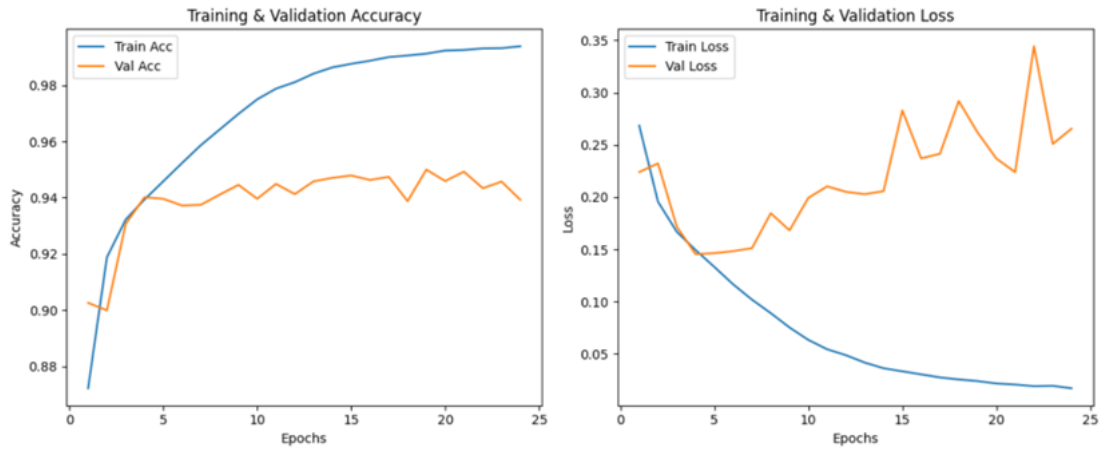


Figure 21: Training vs validation accuracy and loss for Resnet (Patches)

The model was trained for 50 epochs but since early stopping was implemented, it stopped earlier. The training accuracy of the Resnet Model increased rapidly. However, the validation accuracy stopped around 94 or 95% which might suggest potential overfitting. Similarly, the loss curve of training gradually decreased while validation fluctuated a number of times and increased after several epochs.

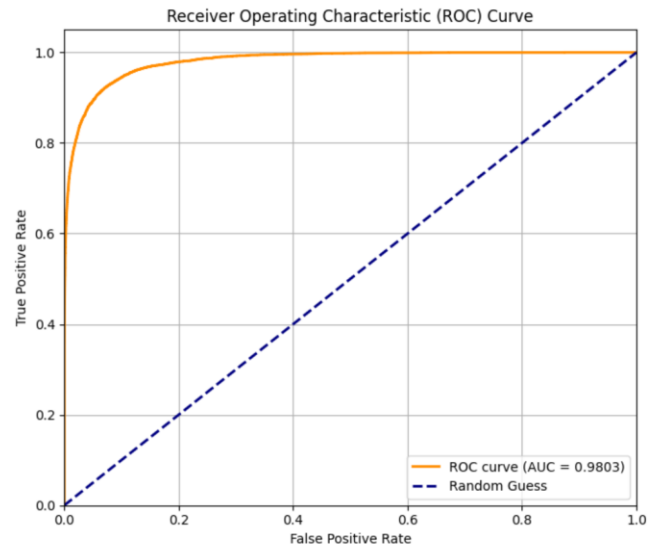
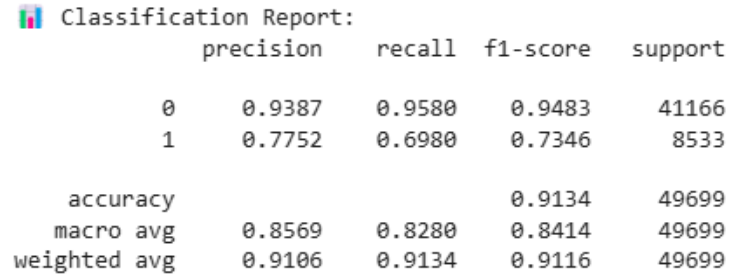


Figure 22: ROC curve for Resnet (Patches)

The Area under the curve (AUC) value is 0.9803 for the Resnet model. This indicates the model can effectively differentiate between lesion and non-lesion patches.

Vision transformers predictions on patch images



Classification Report:				
	precision	recall	f1-score	support
0	0.9387	0.9580	0.9483	41166
1	0.7752	0.6980	0.7346	8533
accuracy			0.9134	49699
macro avg	0.8569	0.8280	0.8414	49699
weighted avg	0.9106	0.9134	0.9116	49699

Figure 23: Classification Report for ViT (Patches)

The vision transformer model achieved accuracy of 91.34%. It represented a decrease in its performance in lesion patches which shows 0.6980 recall value and 0.7346 f1-score. This suggests that the lesion patches were misclassified. On the other hand, for the non-lesion patches recall and f1 score was 0.9580 and 0.9483 respectively which shows that the model was correctly identifying negative cases.

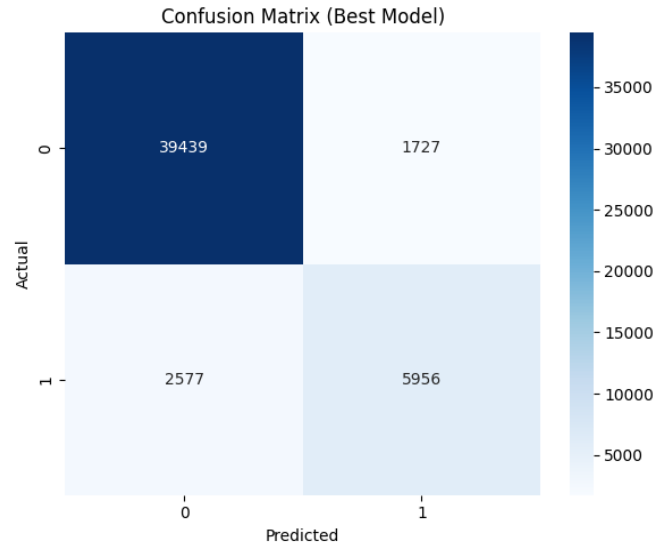


Figure 24: Confusion matrix for ViT (Patches)

The confusion matrix of the model suggests that it misclassified 1727 non-lesions as lesions indicating false positives and 2577 lesions as non-lesions indicating false negatives. This reflects the lower recall in the classification report.

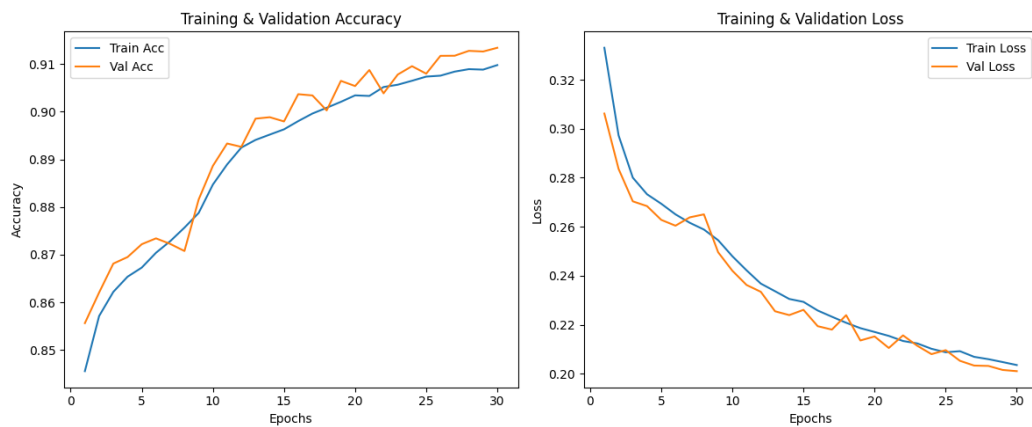


Figure 25: Training vs validation accuracy and loss for ViT (Patches)

The model was trained for 30 epochs, and early stopping was not implemented in this model. By the end of training, both the training and validation accuracy increased over 90%. Both training and validation loss curve experience declined over 30 epochs.

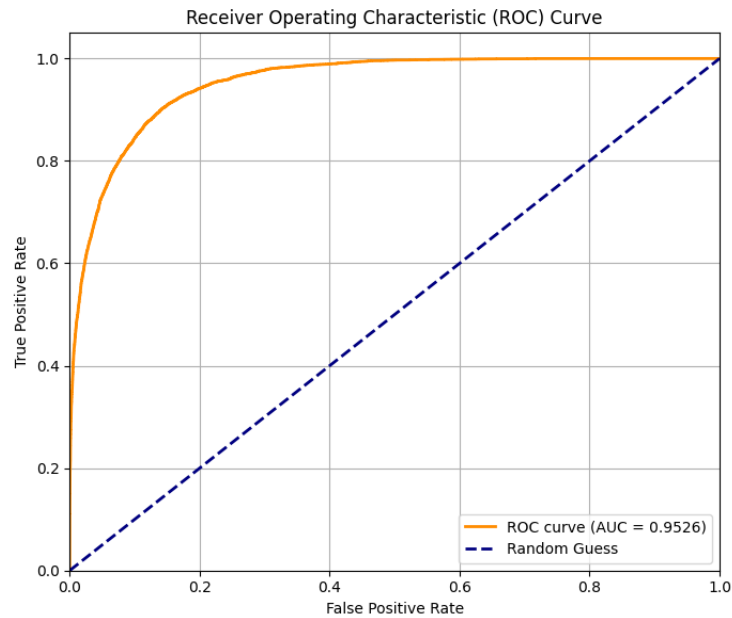


Figure 26: ROC curve for ViT (Patches)

The ROC curve of the vision transformer model shows the curve rising steeply towards the top left corner indicating AUC of 0.9526.

Comparison between the models for patches images

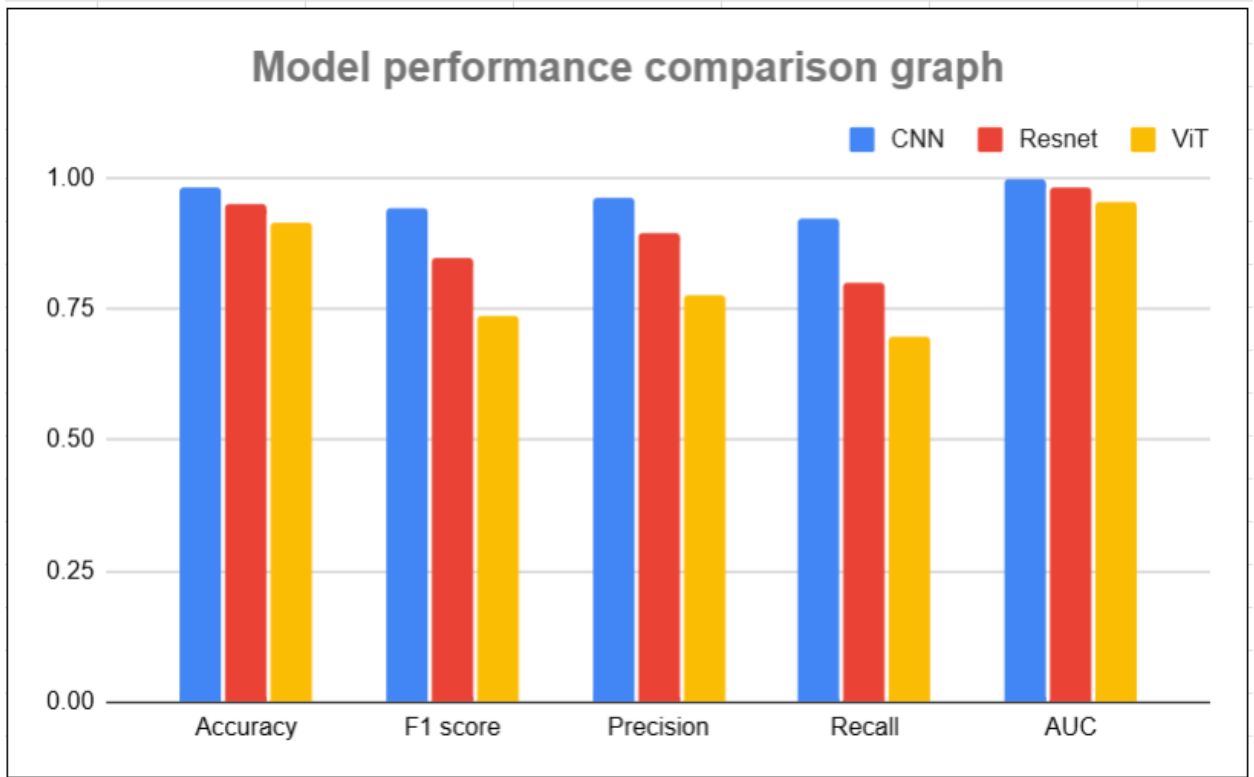


Figure 27: Model performance comparison graph (Patches)

Table 1: Model comparison table (Patches)

Model	Accuracy	F1- score	Precision	Recall	AUC
CNN	0.9806	0.9423	0.9627	0.9228	0.9959
ResNet	0.95	0.8462	0.8963	0.8014	0.9803
ViT	0.9134	0.7346	0.7752	0.698	0.9526


From figure 27 and table 1, it is clear that CNN model outperforms other models based on all metrics such as accuracy, F1 score, Precision, Recall and AUC with the value of 0.9806, 0.9423, 0.9627, 0.9228 and 0.9959 respectively. This proves that the CNN model is most effective in patch-based classification of MRI images due to its localized feature extraction. The convolutional filters are better at detecting localized lesions making it suitable for lesion detection tasks.

The Resnet model followed the CNN model with the metrics value given above. It performed moderately even though it had more complex architecture than CNN. It experienced the lowest recall in comparison with its accuracy, this suggests that it sometimes fails to identify the patches.

Lastly, the Vision transformer gave the poorest performance amongst all. Even though it has special properties to identify long range dependencies and capture global relationships, it lagged behind in all the metrics. This suggests that the transformer may not be as effective as CNN model in detecting patch-based images. CNN and resnet have hierarchical feature extraction and local receptive fields which ViTs lack due to which they may not perform well in tasks requiring accurate localization of small lesions even if they have long range dependencies.

In conclusion, the comparison shows that localization is important in patch-based analysis and CNN outperformed all other models because it focused on local regions.

CNN predictions on slice images



Classification Report:				
	precision	recall	f1-score	support
0	0.9167	0.8582	0.8864	141
1	0.8734	0.9262	0.8990	149
accuracy			0.8931	290
macro avg	0.8950	0.8922	0.8927	290
weighted avg	0.8944	0.8931	0.8929	290

Figure 28: Classification report for CNN (Slices)

This classification report summarizes the performance of CNN model for slice-based images. The model achieved overall accuracy of 89.31%. Class 1 had a precision of 87.34%, F1 score of 89.90% and recall score of 92.62%. The results show that CNN model performed well on slice-based images. However, when compared to patch-based images all the key metrics including accuracy were lower for slice-based images. This suggests that patch-based approach was better especially for localizing the features.

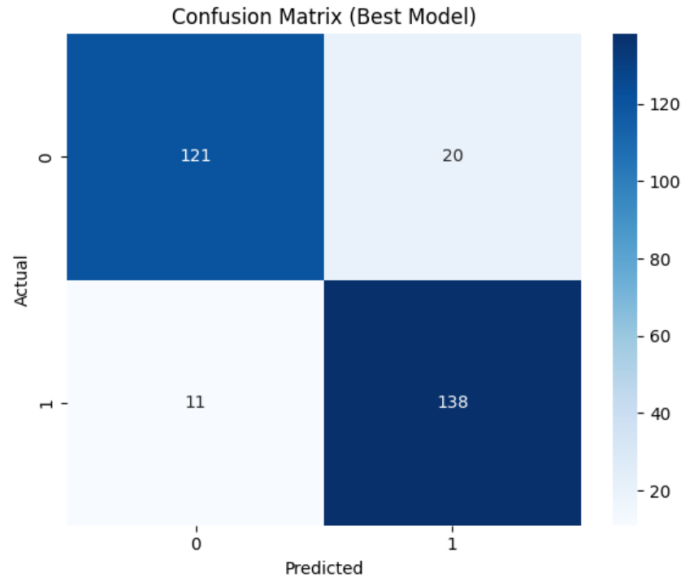


Figure 29: Confusion matrix for CNN (Slices)

The figure above shows the confusion matrix for CNN based approach. Based on the matrix, 20 samples misclassified as 1 when they were actually 0. On the other hand, 11 samples of lesion images were incorrectly classified as non-lesions.



Figure 30: Training vs validation accuracy and loss curve for CNN (Slices)

The training and validation curves for accuracy and loss clearly shows the signs of overfitting. The samples were initially trained for 50 epochs but due to early stopping, it stopped at 30 epochs where training accuracy reached 100% and training loss decreased to 0. The validation accuracy fluctuates around 85 to 90% and the validation loss increases after a few epochs. This shows that the model learns the training data very well, but it struggles to classify the hidden data.

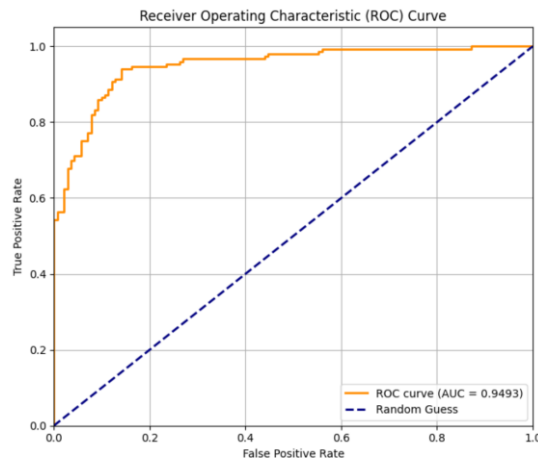
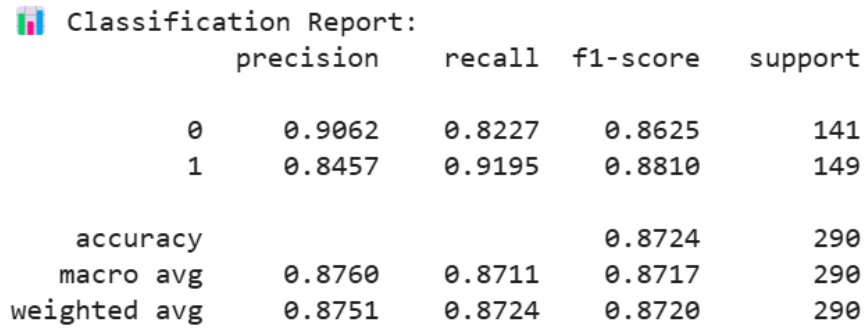


Figure 31: ROC curve for CNN (Slices)

The figure shows that the Area under the curve (AUC) is 0.9493.

Resnet predictions on slice images



Classification Report:					
	precision	recall	f1-score	support	
0	0.9062	0.8227	0.8625	141	
1	0.8457	0.9195	0.8810	149	
accuracy			0.8724	290	
macro avg	0.8760	0.8711	0.8717	290	
weighted avg	0.8751	0.8724	0.8720	290	

Figure 32: Classification Report for Resnet (Slices)

The classification report says that the overall accuracy of the Resnet Model is 87.24%. For class 1, the model has slightly lower precision of 84.57% than class 0 and higher recall and f1 score with 91.95% and 88.10% respectively.

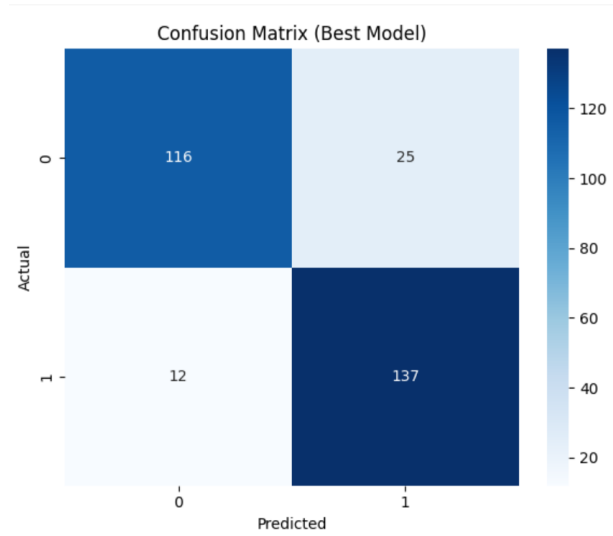


Figure 33: Confusion matrix for Resnet (Slices)

The confusion matrix shows that the model is performing well especially with class 1. The FP and FN class with 25 and 12 samples suggests some misclassification between the classes.

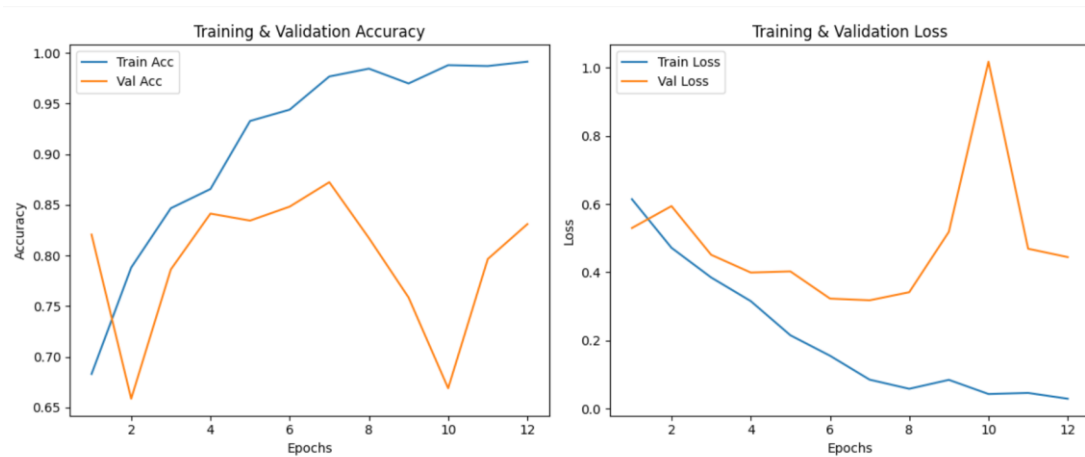


Figure 34: Training vs validation accuracy and loss curve for Resnet (Slices)

Based on the figure above the training accuracy performs well reaching upto 100% but in contrast validation accuracy fluctuates and reaches its lowest point at epoch 2 and epoch 10. For the validation curve, the training loss steeps low, but validation loss decreases at first and increases after 6th epoch reaching a spike at epoch 10. This suggests that the model is overfitting.

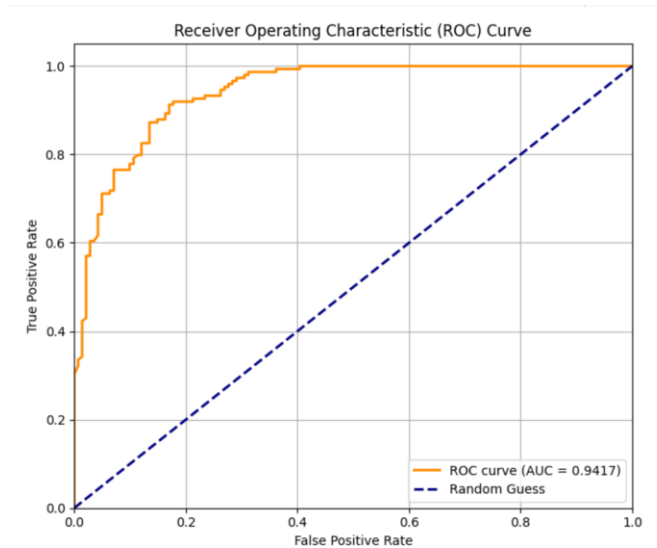


Figure 35: ROC curve for Resnet (Slices)

The ROC curve above shows that Area under the curve value is 0.9417.

Vision predictions on slice images

Classification Report:				
	precision	recall	f1-score	support
0	0.8488	0.5177	0.6432	141
1	0.6699	0.9139	0.7731	151
accuracy			0.7226	292
macro avg	0.7594	0.7158	0.7081	292
weighted avg	0.7563	0.7226	0.7104	292

Figure 36: Classification report for ViT (Slices)

This classification report suggests the weakest performance by the vision transformer on slices images with overall accuracy of 72.26%. While the recall for class 1 was highest with 91.39%, the other metrics precision, and f1 score was lower than class 0.

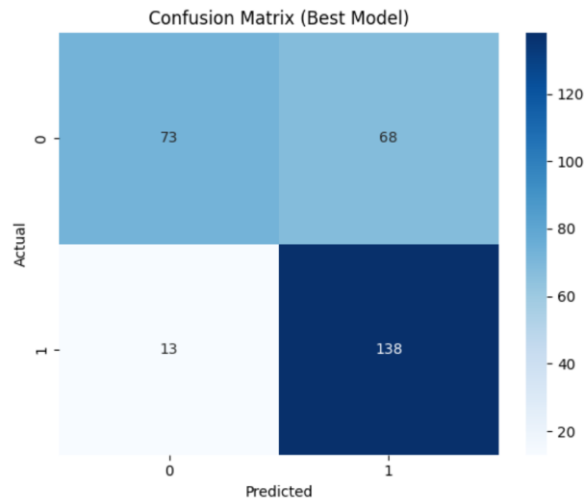


Figure 37: Confusion matrix for ViT (Slices)

The confusion matrix shows there is a high number of samples for false positives which are 68 and 13 were identified as false negatives. This figure clearly shows there is an imbalance in performance seen in the classification report.

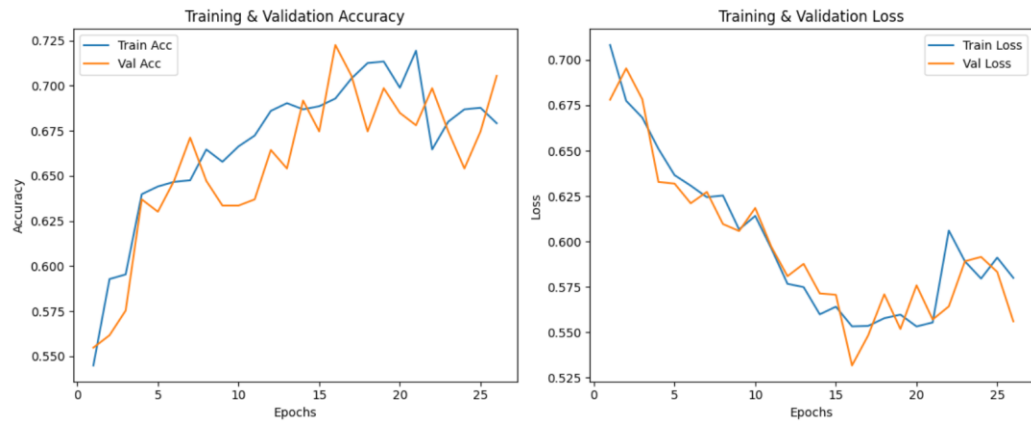


Figure 38: Training vs validation accuracy and loss curve for ViT (Slices)

The figure shows that the training and validation curve shows that they are closely aligned to each other. This shows that the model is not overfitting.

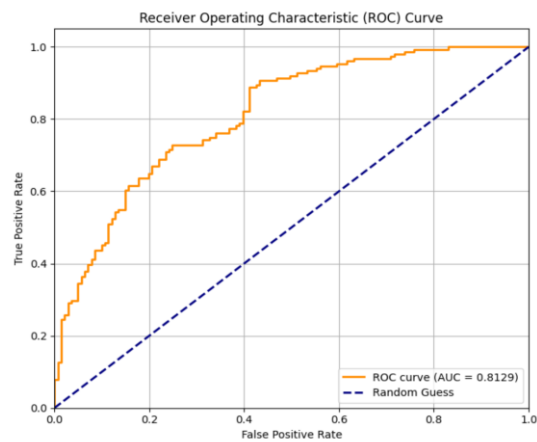


Figure 39 : ROC curve for ViT (Slices)

The area under the curve for the vision transformer model is 0.8129.

Comparison between the models for slice image

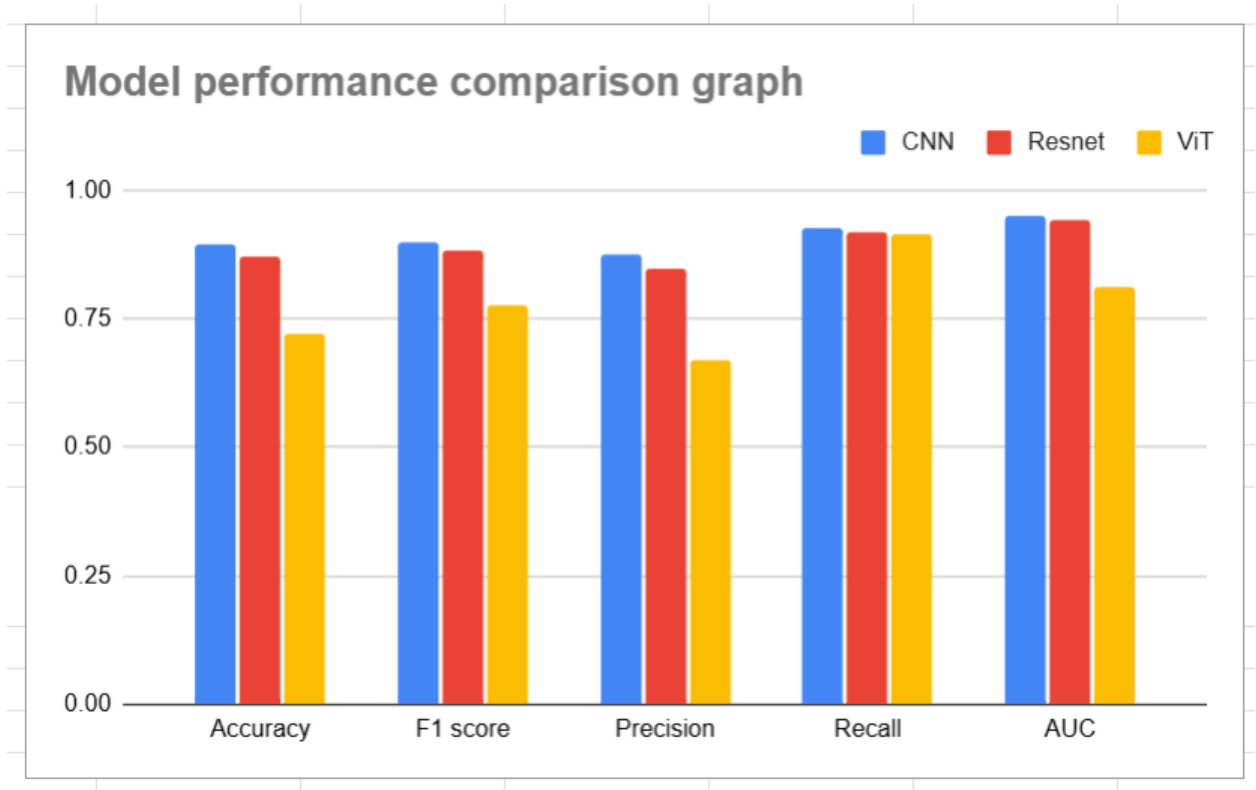


Figure 40: Model performance comparison graph (Slices)

Table 2: Model comparison table (Slices)

Model	Accuracy	F1-score	Precision	Recall	AUC
CNN	0.8931	0.899	0.8734	0.9262	0.9493
ResNet	0.8724	0.8815	0.8469	0.9195	0.9417
ViT	0.7226	0.7743	0.6699	0.9139	0.8129

The figure above shows the model comparison for slices images. In comparison with patch-based images, it is difficult to focus on localized features in slice images. Despite this, the CNN model still performs the best among the other two models for all the performance metrics. The high recall and f1 score value tell that CNN has its strength in correctly identifying true positive values.

Similar to patch-based approach, Resnet performed moderately even if they have residual connections. The Vision transformer (ViT) had the lowest performance. Though its recall is high, the precision is lowest which shows that it indicates a high number of false positives.

CNN model's performance

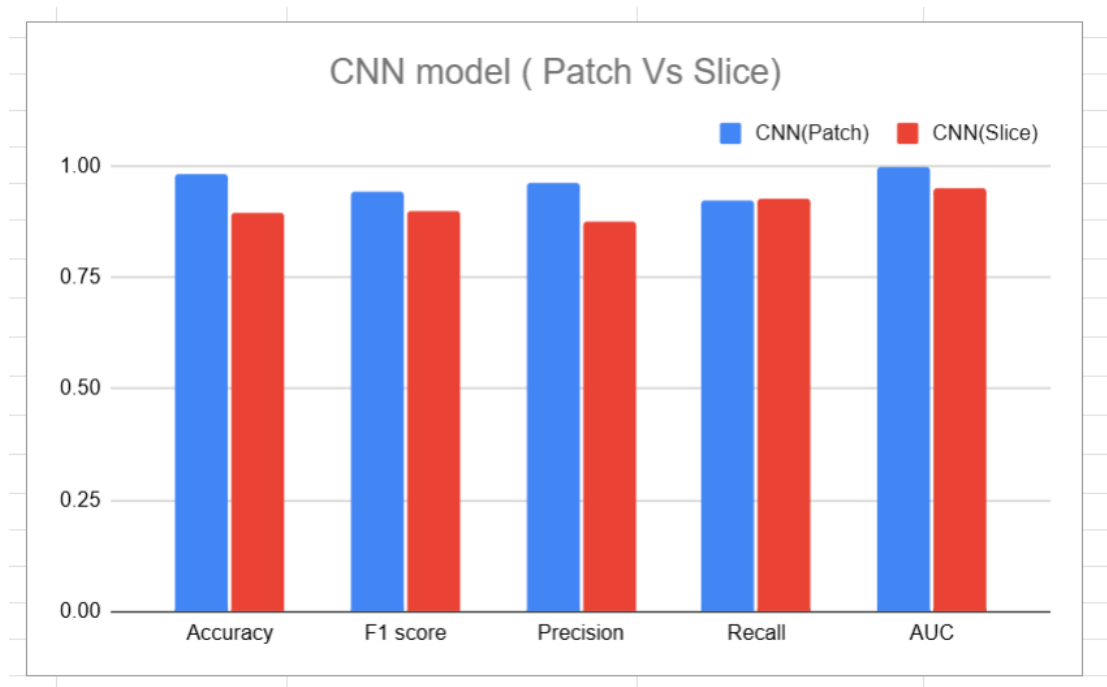


Figure 41: Comparison graph for CNN (Patches vs Slices)

Table 3: Comparison table for CNN (Patches vs Slices)

	CNN model's performance (Patch vs Slice)	
	CNN(Patch)	CNN(Slice)
Accuracy	0.9806	0.8931
F1 score	0.9423	0.899
Precision	0.9627	0.8734
Recall	0.9228	0.9262
AUC	0.9959	0.9493

The bar graph and table clearly show that CNN model performs better in patch-based images than slice images. All the performance metrics are higher in patch-based approach. In contrast the performance drops in slice images because it introduces noise and dilutes lesion signals, making it difficult to localize the lesions. This performance introduces the importance of localized feature learning in medical images. Also, the similar values of recall indicated that CNN is strong in capturing true lesion cases.

Resnet model's performance

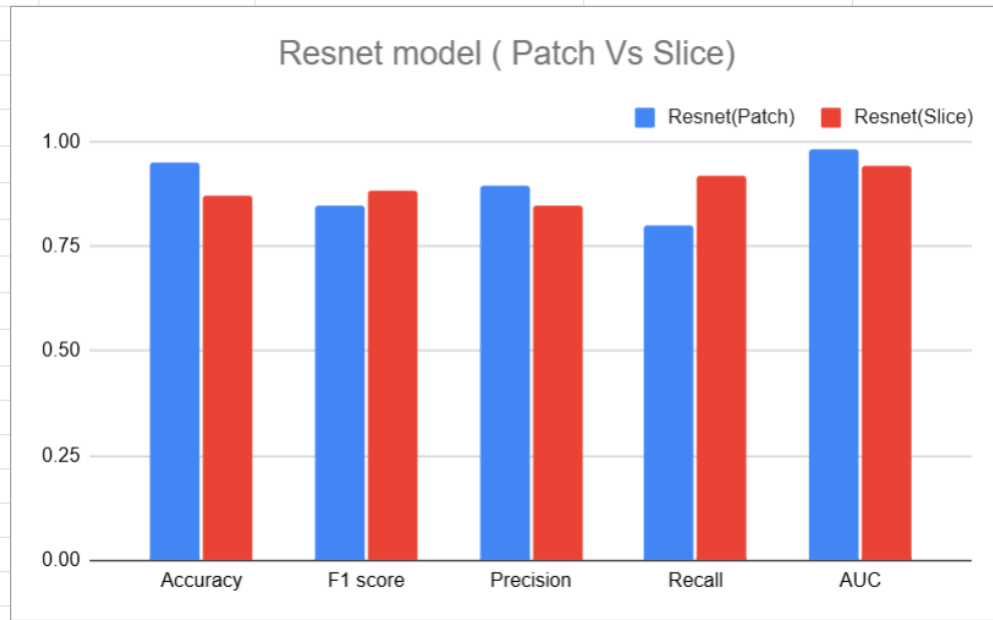


Figure 42: Comparison graph for Resnet (Patches vs Slices)

Table 4: Comparison table for Resnet (Patches vs Slices)

	Resnet Model's performance (Patch vs Slice)	
	Resnet(Patch)	Resnet(Slice)
Accuracy	0.95	0.8724
F1 score	0.8462	0.8815
Precision	0.8963	0.8469
Recall	0.8014	0.9195
AUC	0.9803	0.9417

The figure above shows that only for metrics like accuracy, precision and AUC has higher value in patch images than slice images. This indicates that the model correctly identifies lesions with few false positive images. Since Resnet architecture also uses convolutional neural networks, it shows its strength in capturing localized regions. On the other hand, F1 score, and Recall are higher for slice-based images which identifies more true positive images. This may be due to the global context present in slice images. Overall, this model performs well in both patch and slice images.

Vision transformer's performance

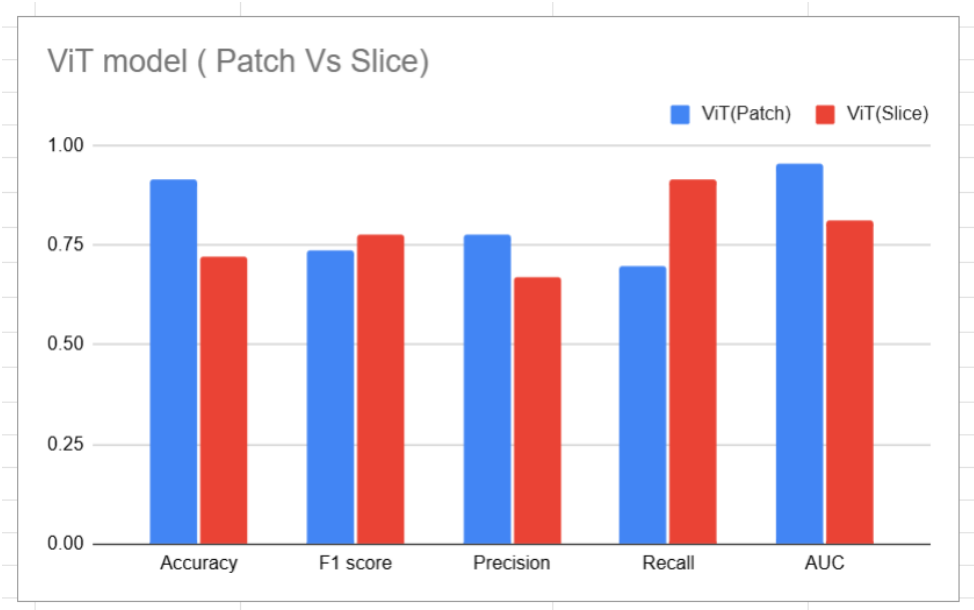


Figure 43: Comparison graph for ViT (Patches vs Slices)

Table 5: Comparison table for ViT (Patches vs Slices)

	ViT model (Patch Vs Slice)	
	ViT(Patch)	ViT(Slice)
Accuracy	0.9134	0.7226
F1 score	0.7346	0.7743
Precision	0.7752	0.6699
Recall	0.698	0.9139
AUC	0.9526	0.8129

The findings illustrate that ViTs performed better in classification tasks for patch images, however it had lowest recall with 0.698 which indicates it missed a lot of true positive images. But on the slices images, recall was highest with 0.9139 which means it captured lots of lesions images. But it came at a cost of lowest precision of 0.6699 which indicates a larger false positive rate and worse class separation. The model's global attention mechanism may have contributed to its tendency to over-detect lesions in the context of the entire image.

ACCURACY ASSESSMENT IN THE CONTEXT OF EXISTING LITERATURE

Other literature reviews focused on slice images, but this approach adopted a patch based approach for training. Each patch was obtained from original slices of MRI images where these patches focused on localized features. They enable the model to focus on rich details which can be better for lesion diagnosis by improving efficiency. They allow the model to extract small textures, edges and patterns which might be difficult in slices because they may contain irrelevant anatomical structures which might distract the model. A comparison was made based on performance accuracy between my model and existing literature reviews. Three research papers were used for the comparison.

The paper ‘Brain MRI based diagnosis of autoimmune diseases using deep learning’ [5] declared the highest accuracy of 100% using CNN model from scratch to distinguish healthy individuals with MS. The healthy dataset was obtained from Kitware data whereas the data with lesions used the same datasets as my research. The authors employed a CNN architecture with six convolutional layers each followed by Relu activation function and three max pooling layers. The feature map was finally flattened and then followed by three fully connected layers with the final layer used for classifying MS or healthy. On the other hand, this model is more compact, efficient and light weight than the model used in reference paper. Although the accuracy is less than accuracy from paper this method focuses more on localized lesions on patch-based input. The same preprocessing method was used where the 3D images were converted to 2D slices of images.

The second comparison was done with the paper titled, ‘Detection of Multiple Sclerosis Using Convolutional Neural Networks: A Comparative Study’ [4]. This paper used both transfer learning and machine learning models such as VGG 16, Random Forest and Support vector machine with highest accuracy of 98.44% in modified VGG 16. Three different datasets were used where one of the datasets was the same as ours dataset. While all of their models rely on the global context of the entire slice, this research focuses on localization on patch-based images.

Lastly the paper ‘Detection of Multiple Sclerosis using Deep Learning’ [8] also employs a transfer learning algorithm VGG 16 to classify full slice based MRI images reaching accuracy of 98.24%. Other transfer learning methods were applied using VGG 19 and Resnet 50 which achieved accuracy of 95.96% and 79.92% respectively. In comparison with this research’s method, the paper not only focused on MS disease, but it also focused on classifying other diseases such as RR, PR, SP, CIS with healthy ones. The dataset was also not similar as it used the data from Laboratory of Imaging Technology Different dataset and it only used 30 MS patient data and enhanced through data augmentation. So, the comparison with this paper highlights the difference in global classification and localized detection.

Table 6: Comparison of our method with other literature reviews

Reference papers	Dataset	Models Used	Accuracy
[5]	Healthy data : Kitware dataset Data with MS : Mendeley Data	CNN from scratch	1
[4]	First : Ozal University Second : Mendeley Data Third : Open Big Healthy Brain	VGG16 VGG19 Random Forest Support Vector Machine	0.9844 0.9771 0.96 0.92
[8]	Laboratory of Imaging Technology Different dataset	VGG16 VGG9 ResNet50	0.9824 0.95.96 0.7992
My Work	Mendeley Data	CNN from scratch Resnet Vision Transformer	0.9806 0.95 0.9134

CHAPTER VII

CONCLUSION AND FUTURE WORK

The detection of Multiple sclerosis is really challenging. A lot of slice-based approaches have been performed which showed limitations in localizing lesions because it focused on global features. That is why improving lesion localization was very important due to which the research focused on a patch-based approach. This method looked quite promising as it focuses on localized features which improved classification performance in comparison with slice images. Various deep learning approaches have been applied in detecting Multiple sclerosis detection for patch images. Among them CNN approach outperformed Resnet and Vision Transformer models even in slice-based images. To grasp the full potential of each model hyperparameter tuning was also used.

While this work has given promising results, there are several limitations that must be acknowledged. Firstly, the patches datasets were limited which might have affected models' generalizability. Secondly, the model is heavily dependent upon the lesion mask for patch extraction which introduces supervised bias. In addition to that, this approach focuses on classification without integrating slice wise localization of lesions. To address these limitations, future work could focus on improving detection by automatic patch extraction process by removing its dependency to lesion masks. This could be done through unsupervised learning. Another work that can be done is mapping the detected lesion patch to their corresponding slice and marking that particular slice as an MS lesion. This could help in improving accuracy in slice detection methods. Also, limited datasets could be addressed using data augmentation or other techniques.

REFERENCES

- [1]A. M. Muslim et al., “Brain MRI dataset of multiple sclerosis with consensus manual lesion segmentation and patient meta information,” *Data in Brief*, vol. 42, p. 108139, 2022, doi: 10.1016/j.dib.2022.108139.

- [2] N. H. Messaoud, R. Ayari, A. B. Abdallah, and M. H. Bedoui, “Automated brain lobe segmentation and feature extraction from multiple sclerosis lesions using deep learning,” *Proceedings Copyright*, vol. 532, pp. 540, 2024.

- [3]K. Memon, N. Yahya, M. Z. Yusoff, H. Hashim, S. S. A. Ali, and S. Siddiqui, “Image Pre-processing for Differential Diagnosis of Multiple Sclerosis using Brain MRI,” in *Proc. 2023 2nd Int. Conf. Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, Vellore, India, 2023, pp. 1–6, doi: 10.1109/ViTECoN58111.2023.10157177.

- [4] N. Afifi, A. T. Abdel-Hamid, and B. A. Abdullah, “Detection of Multiple Sclerosis Using Convolutional Neural Networks: A Comparative Study,” in *Proc. 2023 10th Int. Conf. Soft Computing & Machine Intelligence (ISCMI)*, Mexico City, Mexico, 2023, pp. 77–80, doi: 10.1109/ISCMI59957.2023.10458634.

- [5] D. Amanatidis, G. Chatzisavvas, and M. Dossis, “Brain MRI based diagnosis of autoimmune diseases using deep learning,” in Proc. 2022 7th SEEDA-CECNSM, Ioannina, Greece, 2022, pp. 1–5, doi: 10.1109/SEEDA-CECNSM57760.2022.9932959.
- [6] K. A. A. Ismail, A. K. Dutta, and A. R. W. Sait, “Ensemble Learning-based Multiple Sclerosis Detection Technique Using Magnetic Resonance Imaging,” J. Digit. Radiol., vol. 3, no. 6, 2024, doi: 10.57197/JDR-2024-0078.
- [7] R. Bonacchi, M. Filippi, and M. A. Rocca, “Role of artificial intelligence in MS clinical practice,” NeuroImage: Clinical, vol. 35, p. 103065, 2022, doi: 10.1016/j.nicl.2022.103065.
- [8] S. A. Jannat, T. Hoque, N. A. Supti, and M. A. Alam, “Detection of Multiple Sclerosis using Deep Learning,” in Proc. 2021 AsianConf. Innovation in Technology (ASIANCON), Pune, India, 2021, pp. 1–8, doi: 10.1109/ASIANCON51346.2021.9544601.
- [9] M. Kim, J. W. Seo, M. S. Kim, K. H. Lee, and M. Kim, “White matter tract density index is associated with disability in multiple sclerosis,” Neurobiology of Disease, vol. 198, p. 106548, 2024.

[10] A. Q. Wang, R. Saluja, H. Kim, X. He, A. Dalca, and M. R. Sabuncu, "BrainMorph: A Foundational Keypoint Model for Robust and Flexible Brain MRI Registration," arXiv preprint, arXiv:2405.14019, 2024.

[12] X. Li et al., "Advances in differential diagnosis of cerebrovascular diseases in magnetic resonance imaging: a narrative review," Quant. Imaging Med. Surg., vol. 13, no. 4, pp. 2712–2734, 2023, doi: 10.21037/qims-22-750.

[13] GeeksforGeeks, "Vision Transformers vs Convolutional Neural Networks (CNNs)," GeeksforGeeks. [Online]. Available: <https://www.geeksforgeeks.org/vision-transformers-vs-convolutional-neural-networks-cnns/>. [Accessed: March 15, 2025].

[14] Amazon Web Services, "What is deep learning?," Amazon. [Online]. Available: <https://aws.amazon.com/what-is/deep-learning/>. [Accessed: March 15, 2025].

[15] IBM, "Transfer learning," IBM. [Online]. Available: <https://www.ibm.com/think/topics/transfer-learning>. [Accessed: March 15, 2025].

[16] GeeksforGeeks, "Residual networks (ResNet) – Deep learning," GeeksforGeeks. [Online]. Available: <https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/>. [Accessed: March 15, 2025].

- [17] T. Ekmekyapar and B. Taşcı, “Exemplar MobileNetV2-Based Artificial Intelligence for Robust and Accurate Diagnosis of Multiple Sclerosis,” *Diagnostics*, vol. 13, no. 19, p. 3030, 2023, doi: 10.3390/diagnostics13193030.
- [18] M. Divya, J. Dhilipan, and A. Saravanan, “Detection of Multiple Sclerosis Using Advanced Deep Learning Strategies,” *J. Balkan Tribol. Assoc.*, vol. 29, no. 1, 2023.
- [19] H. Siar and M. Teshnehlab, “Diagnosing and Classification Tumors and MS Simultaneous of Magnetic Resonance Images Using Convolution Neural Network,” in *Proc. 2019 7th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, Bojnord, Iran, 2019, pp. 1–4, doi: 10.1109/CFIS.2019.8692148.
- [20] S. Krishnamoorthy et al., “Automatic intelligent system using medical of things for multiple sclerosis detection,” *Comput. Intell. Neurosci.*, vol. 2023, no. 1, p. 4776770, 2023.
- [21] B. Bejarano et al., “Computational classifiers for predicting the short-term course of Multiple sclerosis,” *BMC Neurology*, vol. 11, p. 1, 2011.
- [22] F. Eitel, J. P. Albrecht, F. Paul, and K. Ritter, “Harnessing spatial MRI normalization: patch individual filter layers for CNNs,” *arXiv preprint, arXiv:1911.06278*, 2019.

[23] G. Macin et al., “An accurate multiple sclerosis detection model based on exemplar multiple parameters local phase quantization: ExMPLPQ,” *Appl. Sci.*, vol. 12, no. 10, p. 4920, 2022.

[24] Amazon Web Services, “What is hyperparameter tuning?,” [Online]. Available: <https://aws.amazon.com/what-is/hyperparameter-tuning/>. [Accessed: March 15, 2025].

[25] M. Saeed, “A gentle introduction to positional encoding in transformer models, Part 1,” *Machine Learning Mastery*, Jan. 6, 2023. [Online]. Available: <https://machinelearningmastery.com/a-gentle-introduction-to-positional-encoding-in-transformer-models-part-1/>. [Accessed: March 15, 2025].

[27] GeeksforGeeks, “Artificial neural networks and its applications.” [Online]. Available: <https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/>. [Accessed: March 15, 2025].

[28] Saturn Cloud, “A comprehensive guide to convolutional neural networks: The ELI5 way.” [Online]. Available: <https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/>. [Accessed: March 15, 2025].

[29] A. A. Mukhlif, B. Al-Khateeb, and M. A. Mohammed, “Incorporating a novel dual transfer learning approach for medical images,” *Sensors*, vol. 23, no. 2, p. 570, 2023.

[30] A. Arora, “Understanding ResNets: A Deep Dive into Residual Networks with PyTorch,” Weights & Biases Reports, [Online]. Available: <https://wandb.ai/amanarora/Written-Reports/reports/Understanding-ResNets-A-Deep-Dive-into-Residual-Networks-with-PyTorch--Vmlldzo1MDAxMTk5>. [Accessed: March 15, 2025].

[31] Papers with Code, "Vision Transformer," Papers with Code. [Online]. Available: <https://paperswithcode.com/method/vision-transformer>. [Accessed: March 15, 2025].