Detection of Alzheimer's Disease using Deep Learning

by

Mahmudul Hasan
15201041
Syed Zafrul Hassan
15301064
Tanzina Hassan Azmi
15201043
Emtiaz Hossain
15101069

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
December 2019

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Mahmudul Hasan 15201041	Syed Zafrul Hassan 15301064
Tanzina Hassan Azmi 15201043	Emtiaz Hossain 15101069

Approval

The thesis/project titled "Detection of Alzheimer's Disease using Deep Learning" submitted by

1. Mahmudul Hasan (15201041)
2. Syed Zafrul Hassan (15301064)
3. Tanzina Hassan Azmi (15201043)
4. Emtiaz Hossain (15101069)

Of Fall, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on December 8, 2019.

Examining Committee:

Supervisor:
(Member)

Mohammad Zavid Parvez, PhD
Assistant Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Md. Ashraful Alam, PhD
Assistant Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam,PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Mahbubul Alam Majumdar, PhD
Professor and Chairperson(CSE), Interim Dean, School of Sciences
Department of Computer Science and Engineering
Brac University

Ethics Statement (Optional)

This is optional, if you don't have an ethics statement then omit this page

Abstract

Machine Learning has been on top of its form over the last few years. It covers a vast area of predictive web browsing, email and text classification, object detection, face recognition etc. Among all of the other applications of machine learning, deep learning has gained more popularity over the last several years. It is helping researchers in the field of biomedical problems like detection of different types of diseases such as Cancer, Alzheimer, Malaria, Blood cell detection etc. Deep learning is a subset of machine learning algorithms that is used for classification, image processing etc. by extracting features. In our research, we used Convolutional Neural Network (CNN) for classification of Alzheimer patients and healthy patients from Magnetic Resonance Imaging (MRI) data. The dataset (OASIS-1) contains 416 subjects classified into non-demented and mild to moderate Alzheimer's disease. The classification of this type of medical data is very significant for creating a prediction model or system to examine the presence of the disease in different subjects or to estimate the phase of the disease. Classification of Alzheimer's disease has always been difficult and selecting the distinctive features is the most complicated part of it. By using different CNN architectures like InceptionV3, Xception, MobileNetV2, VGG16, VGG19 we have classified Alzheimer's patients from healthy subjects by calculating different model accuracy, confusion matrix and ROC curve from their MRI data. Among all the models, the basic CNN and the InceptionV3 provide the best accuracy up to 90.62%. This research shows us how different CNN architectures perform on our MRI data of Alzheimer's subjects and healthy subjects in case of classification and helps us to find the best models for the detection of Alzheimer's disease.

Keywords: Alzheimer; Machine Learning; Deep Learning; CNN; MRI; OASIS-1; Confusion Matrix; ROC curve

Dedication (Optional)

A dedication is the expression of friendly connection or thanks by the author towards another person. It can occupy one or multiple lines depending on its importance. You can remove this page if you want.

Acknowledgement

To begin with, our heartfelt gratitude to Almighty Allah who gave us a chance to work on this thesis and pull it off successfully. We have put our best efforts to reach our goal on time with efficiency. It has been an amazing learning experience for us. Moreover, we wholeheartedly show our gratitude towards our supervisor Mohammad Zavid Parvez Sir for his guidance and contribution. Without his proper counseling, we would have not been able to complete our thesis work. In every step of our research, he gave us necessary assistance and motivation to achieve our desired goal. Furthermore, special gratitude to our co-supervisor Md. Ashraful Alam Sir who shared his knowledge and experience with us in order to help us achieve our goal. In addition, we are also very thankful to our family, friends and loved ones who supported us in our research. Last but not the least, we appreciate BRAC University for giving us the opportunity and also providing us all the necessary resources to achieve our thesis objective.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$ADNI$ Alzheimer's Disease Neuroimaging Initiative

ANN Artificial Neural Network

AUC Area Under The Curve

CNN Convolutional Neural Network

DNN Deep Neural Network

$GLCM$ Gray-Level Co-occurrence Matrix

HC Healthy Control

MCI Mild Cognitive Impairment

ML Macine Learning

MRI Magnetic Resonance Imaging

NC Normal Control

PET Positron Emission Tomography

$ReLU$ Rectified Linear Unit

RNN Recurrent Neural Network

ROC Receiver Operating Characteristic

SVM Support Vector Machine

VFI Voting Function Intervals

VGG Visual Geometry Group

Chapter 1

Introduction

This chapter provides an overview of the purpose of our research and a general idea of our whole thesis.
Deep learning is part of machine learning (ML) algorithms based on artificial neural networks (ANN). Moreover, many architectures of Deep learning such as Deep Neural Networks (DNN), Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) have been successfully used in natural language processing, computer vision, audio recognition, speech recognition, medical image analysis, where in some cases they have produced better results than human. Furthermore, a neural network is a series of algorithms that recognize relationships in a set of data through a process which is very similar to human brain operation. In our research We have used CNN because of the efficient recognition algorithm. This algorithm is very effective for pattern recognition and image processing. It takes images as input and builds a model that processes the images by extracting the features and recognize a pattern. By using the pattern CNN tries to identify the similarities of a new input as accurately as possible This algorithm is very popular because of its simple structure, adaptability and less training parameters and this algorithm reduces complexity of a network model. We used Brain MRI images from OASIS-1 dataset for the classification of the Alzheimer's subjects and healthy subjects in order to build a predictive and trained model using many CNN architectures like VGG, InceptionV3, MobileNetV2.

1.1 Motivation

Alzheimer's has become one of the major diseases throughout the world. There are approximately 50 million people all around the world suffering from Alzheimer's disease and different types of dementia. A report from the United Nations states that, this number is more than Columbia's population. Someone in the world develops dementia every 3 seconds. If there are not any breakthroughs discovered, this rate can surpass 152 million by the year 2050. Moreover, there is no cure for Alzheimer's disease but if can be detected in the early stages then with proper treatment the patients will have options available to cope with the symptoms and improve the quality of life. That is why we are working on predicting Alzheimer's disease. On the other hand, deep learning has become popular more than ever in eminent research fields like facial and speech recognition, machine translation, natural language processing, etc. And the reason why deep learning stand out among others is because of its

accuracy. Deep learning produces very high-level accuracy that are superior to human specialists in most cases. Convolutional Neural Network (CNN) is one of the most popular and widely used deep learning architectures. In the last decade, the use of Convolutional Neural Network (CNN) of neural network has increased rapidly for various computational tasks e.g. image recognition. Convolutional Neural Network (CNN) uses convolution operation for classification and prediction and gives high accuracy. So, for higher accuracy we are using Convolutional Neural Network (CNN) and its different architectures for predicting if the person has Alzheimer's or not.

1.2 Aims and Objectives

The major goals of our thesis:

- Training the MRI data of subjects consisting of young, middle aged and older adults of OASIS-1 dataset for building train and test models.
- Applying the basic CNN and other architectures of CNN like InceptionV3, VGG16, VGG19, Xception, MobileNetV2 to predict if the subject has Alzheimer's disease or not.
- Analyzing the results of various models and finding the model with maximum accuracy to find out the best model to predict the availability of Alzheimer's disease of a person.

1.3 Thesis Overview

Chapter 1 is the introduction of our thesis work. Our motivations and objectives to do this thesis are also mentioned here as well as a short overview of the methodology that has been followed by us.

Chapter 2 consists of the literature review where we have demonstrated the background study that we have done for this thesis.

Chapter 3 consists of workflow, pre-processing and data labelling for our thesis.

Chapter 4 consists of pre-trained models and implementation of our overall thesis.

Chapter 5 contains the conclusion and future work.

Chapter 2

Literature Review

This chapter consists of the literature review, in which we explained the background study we did for our thesis.

2.1 Convolutional Neural Network

Neural networks fig: 2.1 consist of various algorithms which are used for recognizing patterns [37]. They are used for classification and clustering. They use machine perception, clustering or labeling raw input for interpreting sensory data. They recognize patterns which are numerical and contained in vectors, so all the real-world data such as images, text, sound, etc. has to be translated. The structure of neural network consists of one input layer, one or more hidden layers and one output layer. These layers can be dense layer, convolutional layer, pooling Layer, recurrent layer, normalization layer, rectified linear unit layer, fully connected layer etc. Convolutional Neural Networks fig: 2.2 is basically a class of neural networks which uses convolutional layers. Convolutional layers are based on mathematical operation called convolution [35]. It is one kind of linear operation. Instead of general matrix multiplication, Convolutional Neural Network uses convolution in at least one of their layers. It operates on two images or two signals respectively in 2D and 1D. One as basically the input, and the other as a filter which is called kernel. So basically, as input convolution takes two images and the third one is produced as output [19]. Convolutional Neural Networks consist of an input layer,

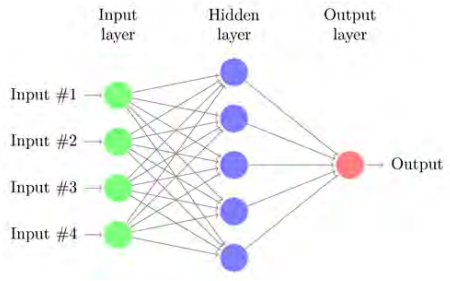

Figure 2.1: Neural Network (NN) [20].

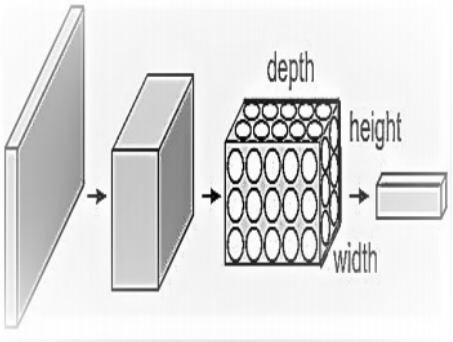

Figure 2.2: Convolutional Neural Network (CNN) [36].

multiple hidden layers, and an output layer fig: 2.3. Typically, the hidden layers are consisting of convolutional layers, pooling layers, normalization layers and fully connected layers. The layers have 3 dimensions which are width, height and depth. It is mainly used for image processing, image classification, image segmentation and other auto correlated data. It also has other uses in video recording, text analysis, speech recognition, natural language processing, text classification as well as various AI systems.

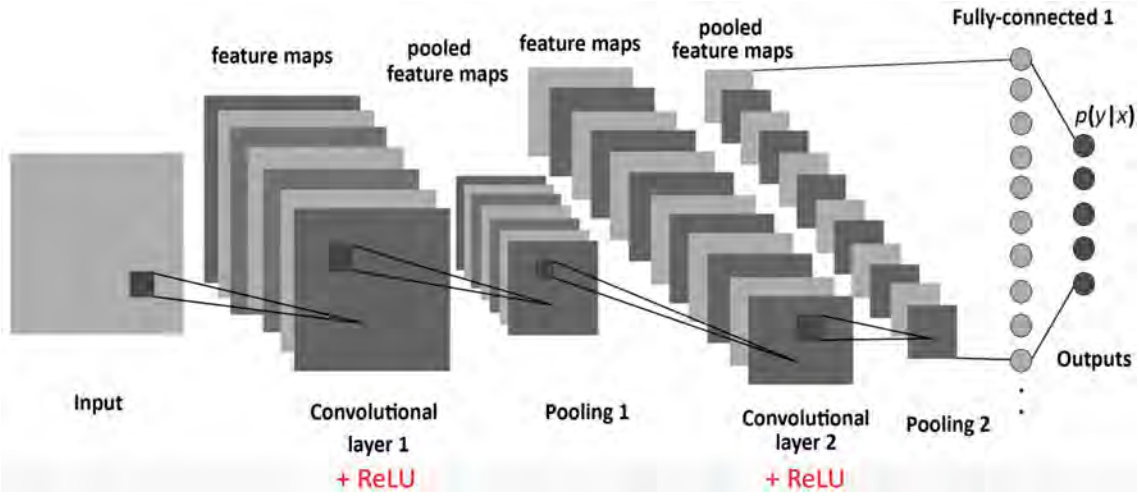

Figure 2.3: CNN Architecture [20].

2.2 How CNN Works

2.2.1 Input Layer

This layer works based on image data only. There are three dimensions of an image (width x height x depth) which is a matrix of pixel values. For instance, if we have an image of (32x32x3) then the input layer will hold raw input of image that has value of (width=32, height=32, depth=3) [14]. Here depth stands for RGB channels. Moreover, input layer has to be divisible by 2. For instance, 32, 64, 224, 384, 512 etc. In order to work with input layer, the three-dimensional matrix must be reshaped into a single column. For instance, if an image has a dimension 24x24=576, before passing it into the input layer it must be converted into 576x1. If there are 'n' training examples then dimension of input will be (576, n).

2.2.2 Convolution Layer

This is the layer where the convolution process happens and the Convolutional Neural Network learns. Convolution layer does most of the computational heavy lifting. It is the core building block of a Convolutional Network. It has some parameters and hyper parameters. These parameters are made up with filters/kernels/K. Convolution layers extract features using these filters and then learns from it. For this reason, this layer is usually known as feature extractor layer. Input images are compared segment by segment in order to distinguish the similarities and the differences. These segments are called features. The convolution layer takes one or more

features fig: 2.4 from the input images and then using image matrix it creates dot product and one or more matrix. If we have an image of 5x5 and the pixel values are 0, 1 and the filter matrix is 3x3 then the 3x3 filter matrix will multiply with 5x5 image matrix which is known as "Feature Map" [21]. The filter moves from left to right with a specific stride value till it parses the full width. Then it moves down to the beginning left side of the image with the same stride and repeats the process until the full image is traversed [23]. The main goal of this convolution process is extracting the high-level features like edges. It can also perform various operations such as blur and sharpen, edge detection by applying filters.

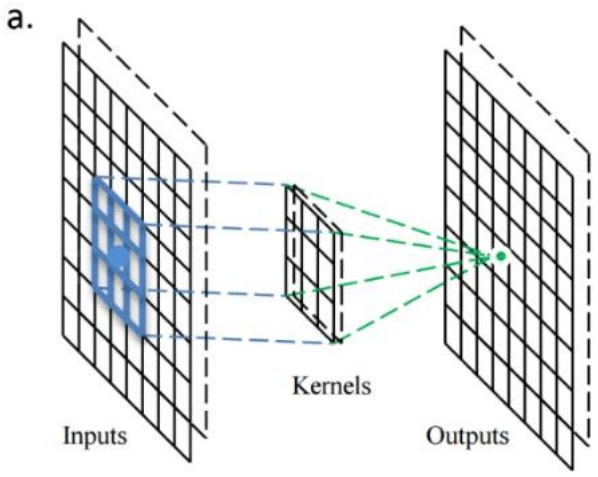

Figure 2.4: Creating a feature map from a convolutional kernel.

If we have a volume of size

$$W1 \times H1 \times D1$$

we need four hyper parameters which are [36]:

- number of filters K
- their spatial extent F
- the stride S
- the amount of zero padding P

It will produce a volume of size

$$W2 \times H2 \times D2$$

where [36]:

$$\begin{aligned} W2 &= \frac{W1 - F + 2P}{S + 1} \\ H2 &= \frac{H1 - F + 2P}{S + 1} \\ D2 &= D1 = K \end{aligned} \tag{2.1}$$

2.2.3 Padding

Sometimes the filter does not accurately match with the input image. Then there are two options fig: 2.5.

- **Zero-Padding:** It means padding the image with zeros so that it accurately fits.
- **Valid-Padding:** It means the part of the image where the filter did not fit will be dropped. That means it only keeps the valid part of the image.

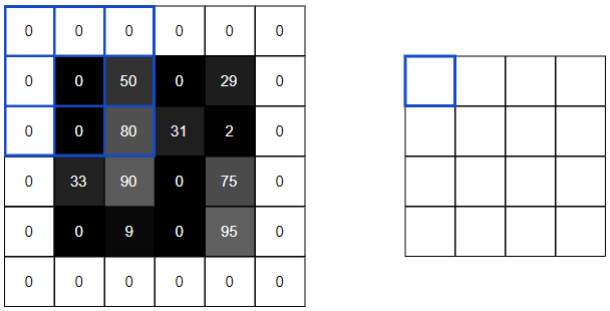

Figure 2.5: A 4x4 input convolved with a 3x3 filter to produce a 4x4 output using same padding [35].

2.2.4 Rectified Linear Unit Layer (ReLU)

ReLU stands for Rectified Linear Unit. It is an activation function. ReLU's goal is to impose non-linearity. The output of ReLU is [21], [17]:

$$ReLU = max(0, x) \tag{2.2}$$

Which means after having the resultant values, it successfully removes the values that are negative from an activation map by setting them to 0 and will keep the positive number unchanged fig: 2.6. This layer does not contain any parameters or additional hyperparameters. ReLU activation function has become the default activation function of many types of neural networks, although there are other activation functions like Tanh and Sigmoid function. But ReLU is usually preferred as it makes the model easier and faster to train and often achieves better performance.

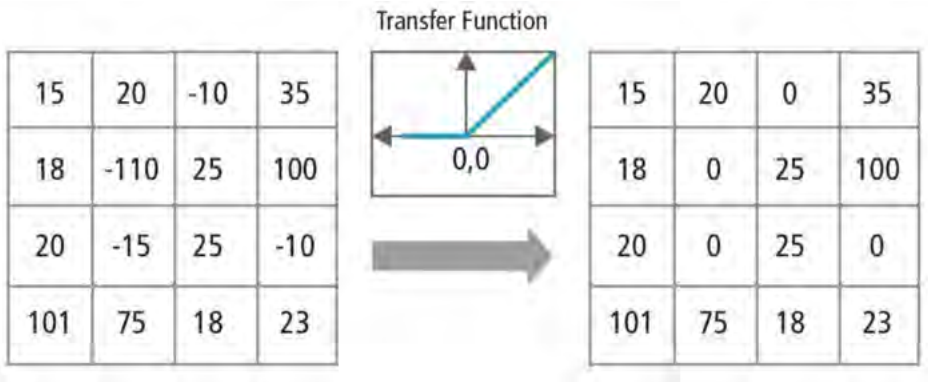

Figure 2.6: ReLU operation [21].

2.2.5 Pooling Layer

It is inserted repeatedly in ConvNet in order to reduce the size of the volume when the image is too large so that it makes the computation fast, prevents from over fitting and reduces memory. There are various kinds of pooling layers such as Max Pooling fig: 2.7, Average Pooling, Sum Pooling. Max pooling takes the largest value from the feature map, Average pooling takes the average by calculating for each patch of the feature map, Sum pooling takes the sum of all elements in the feature map. The most popular and common type of pooling layer is max pooling. There are two hyper parameters for pooling layer, one is Filter(F) and another is Stride(S).
If the dimension of the input is (W1 x H1 x D1) where their spatial extent F, the stride S then it produces a volume of size (W2 x H2 x D2) [36], Where-

$$\begin{aligned} W2 &= \frac{W1 - F}{S + 1} \\ H2 &= \frac{H1 - F}{S + 1} \\ D2 &= D1 \end{aligned} \tag{2.3}$$

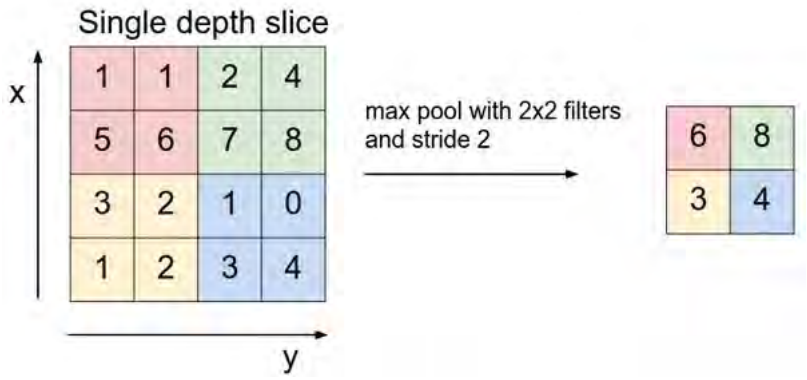

Figure 2.7: Max Pooling [21].

2.2.6 Fully Connected Layer

In this layer, the input from the past layers are flattened from matrix into vector. After flattening, the volume of the previous layer is fed into the fully connected layer like a neural network. By looking at the output of the previous layer this layer decides which features mostly match a particular class. It works on high level features that have particular weights. For this reason, fully connected layer provides the correct probabilities for the different classes as it computes the products between the weights and the previous layer. By using the activation function outputs are classified as a dog, cat, car, etc.

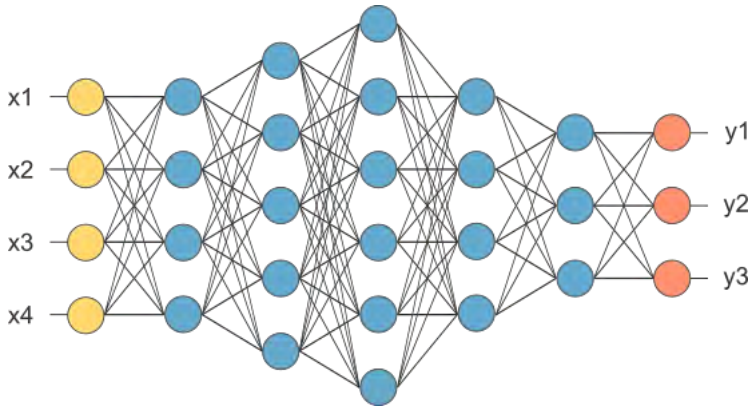

Figure 2.8: Fully connected layer [21].

2.3 Previous Works

We have found many research papers on classification and detection of alzheimer disease where they have used different datasets and different techniques. For example, Kloppel et al. detected Alzheimer's Disease by using patient's MRI data and he has used linear SVM to achieve his goal [2]. However, Averson used two different types classifier which are SVM binary classifier and multi-class classifier to detect Alzheimer's Disease from brain MRI images [8]. He has used Dimensional reduction and variations methods for analysis of structural MRI data. There was another related work done by Vemuri et al. who have used SVM also to find differences between Alzheimer's patients and healthy subjects and he has developed three different types of classifiers using demographic, genotype data and MRI [3]. On the other hand, Gray used FDG-PET images from ADNI dataset and random forest classification in order to generate a model which diagnose Alzheimer disease [5]. From the past few years, deep learning models like CNN has been used to detect pattern and image processing. So, we have found some many research work of Alzheimer disease based on deep learning. Gupta et al. has classified MRI instances into three types which are Healthy Control (HC), Mild Cognitive Impairment (MCI) and Alzheimer's Disease (AD) using ADNI dataset. He developed a sparse autoencoder model to find out bases from natural images and then to extract features from

the dataset he used convolution [6]. By using the MRI scan of the brain from ADNI data set Payan and Montana predicted Alzheimer disease using sparse autoencoders and 3D convolutional neural network (CNN). In their experiments they used sparse autoencoders with sparsity constraints to invest whether sparse autoencoder can find out necessary filters to perform conv operations. They have trained an auto-encoder to a series of 3D patches that were selected randomly. After the training they created a 3D convolution network so that it can take an MRI scan as input. Then they have utilized the weights as filter. After that they obtained a conv layer consisting of 150 three-dimensional feature maps by applying convolutions with all the bases. They have applied a 5*5 max pooling operation for minimizing the volume of the characteristic maps of convolutionary surface. They had also developed a 2D convolutional neural networks (CNN) model that show very similar results as well [10]. However, Liu and Shen used both unsupervised and supervised techniques to distinguish Alzheimer's Disease and MCI patients based on MRI images on a pre-trained large CNN [7]. Alzheimer disease can be treated with modern technology if the disease is diagnosed earlier. By keeping the idea in mind, Hosseini-Asl et al. developed a deep 3D CNN model which can detect generic features to predict Alzheimer disease using ADNI dataset [11]. Similarly, Sarraf S. used fMRI data and CNN to identify Alzheimer's disease [12]. They have used the famous CNN architecture LeNet-% to identify Alzheimer's patients from healthy subjects. They have achieved 96.85% accuracy rate. On the other hand, Cheng and Liu used CNN and RNN to diagnose Alzheimer disease from PET images [13]. With their proposed model they have achieved promising classification performance of an AUC of 95.28% for classification of AD vs NC and 83.90% for classification of MCI vs. NC respectively. Throughout our thesis work, we have found a different sort of work on Alzheimer disease. That research was to develop a novel data mining system with three different types of classifiers which are SVM, Bayes statistics, and voting function intervals (VFI) to derive a pattern matching quantitative index to predict the possibility of mild dementia to Alzheimer's disease [4]. In Addition, Jyoti and Yanqing presented another work based on CNN to detect and classify Alzheimer's disease [15]. Their work was based on OASIS dataset. They have developed a s global state of the network that includes 3 models of DenseNet [9]. Then they have used 3 build patches from 3 physical planes for each MRI data. In the proposed network they used the patches as input. For input, the distinct models take MRI image and create the representation. With their proposed model they have achieved accuracy of 93.18%. Another OASIS dataset base work was done by Amulya et al. who used GLCM method for classification of Alzheimer's disease [38].

Chapter 3

Workflow and Dataset Description

3.1 Workflow

This chapter provides a brief idea regarding our workflow. In flowchart 3.1 we have described our working process from beginning to end. At first we collected our dataset from official OASIS website [39]. After fetching data we came to know that there are images of different size. Here comes our data pre-processing part. There are two parts in our image pre-processing - Image resizing and Image denoising. After pre-processing we labeled our data for binary classification and fixed our sample size. After that we split our data in train data and test data. We moved to our next part which is preparing our self-designed model and other CNN models for training. After that we checked results like accuracy, precision recall, confusion matrix and area under ROC curve for different models.

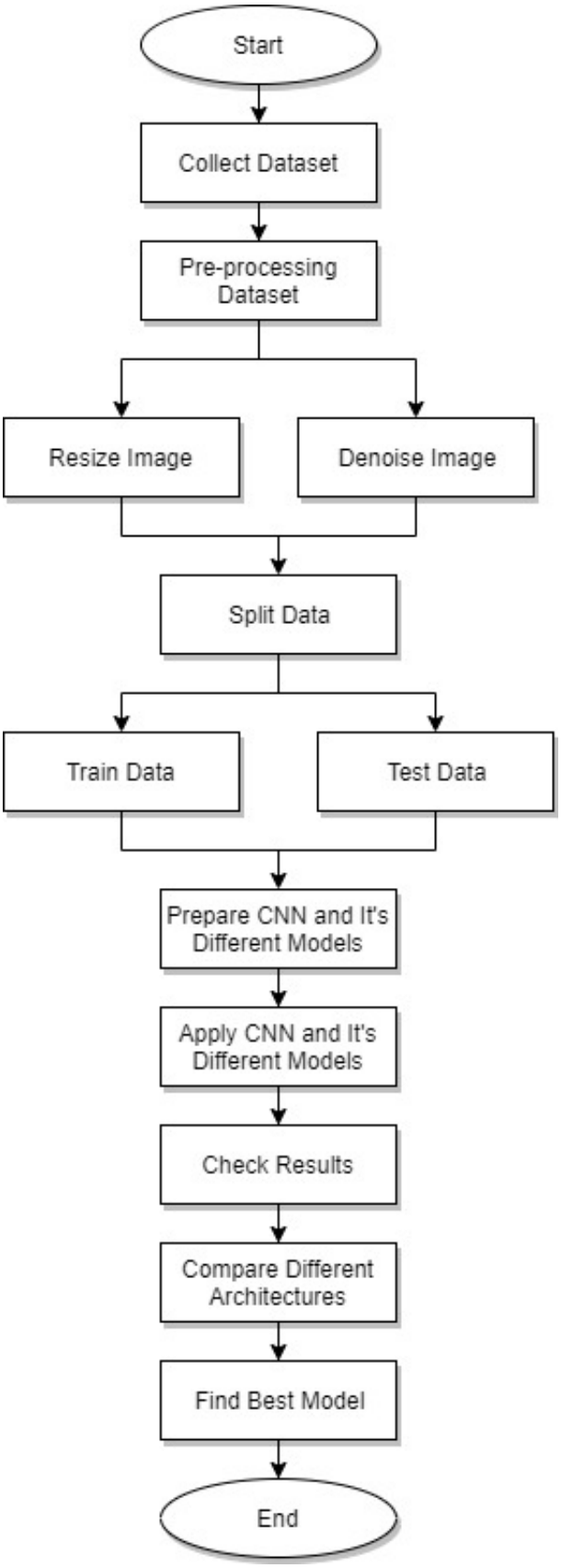

Figure 3.1: Proposed System.

3.2 Data-set Description

3.2.1 Summary

OASIS stands for Open Access Series Of Imaging Studies. The goal of Oasis is to provide neuro-imaging datasets of the brain that are free available for data analysis and distribution. In our research we have used OASIS-1 dataset. This dataset contains MRI data of 416 subjects. These subjects age range from 18 years old to 96 years old. For every individual, three or four MRI scans are included that had been acquired in single meeting. The subjects include both men and women and all of them are right-handed. 100 out of the 416 subjects that are aged over 60 have been diagnosed with Alzheimer's disease (AD) ranging from very mild to moderate level. In addition, for 20 non-demented subjects, a reliability data set is included that contains images of the following visit within 90 days of their initial session [1].

3.2.2 Additional data

Clinical, demographic,and derived anatomic measures are located in the spreadsheets files (oasis-cross-sectional.csv).

Clinical: Clinical Dementia Rating (CDR) corresponds to :
0 = non-demented;
0.5 = very mild dementia;
1 = mild dementia;
2 = moderate dementia.
All subjects with dementia (CDR>0) were diagnosed with probable Alzheimer's disease.

Table 3.1: OASIS-I CSV Dataset Example.

ID	M/F	Hand	Age	Educ	CDR	eTIV	nWBV	ASF	Delay
OAS1_0001_MR1	F	R	74	2.0	0.0	1344	0.743	1.306	NaN
OAS1_0002_MR1	F	R	55	4.0	0.0	1147	0.810	1.531	NaN
OAS1_0003_MR1	F	R	73	4.0	0.5	1454	0.708	1.207	NaN
OAS1_0004_MR1	M	R	28	NaN	NaN	1588	0.803	1.105	NaN
OAS1_0005_MR1	M	R	18	NaN	NaN	1737	0.848	1.010	NaN

Demographics: Gender (Male/Female), Age, Handedness (Right handed), Education (Educ). Education codes correspond to :
1 = less than high school graduation,
2 = high school graduation,
3 = some college,
4 = college graduation,
5 = beyond college.

Table 3.1 is a chunk of the CSV file of our dataset. It contains the information of 5 subjects such as male or female, handedness, age, education, clinical dementia rating (CDR), etc.

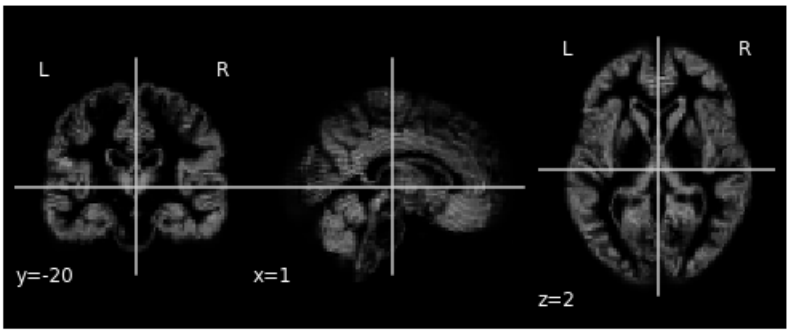

Figure 3.2: OASIS-I MRI Image Data Example.

Figure 3.2 is MRI images of a subject out of 416 subjects of our dataset. These MRI images are in .NII format.

3.3 Image Pre-processing

In the image pre-processing part, we have done image resizing and image denoising using openCV3.

3.3.1 Image Resizing using OpenCV (Python)

Image resizing is used to refer scaling of images. Scaling comes useful in many image processing as well as machine learning applications. It helps in reducing the number of pixels from an image and that has several advantages For example, It can reduce the time of training of a neural network as more is the number of pixels in an image more is the number of input nodes that in turn increases the complexity of the model. It also helps in zooming in images. Many times, we need to resize the image i.e. either shirk it or scale up to meet the size requirements. OpenCV provides us several interpolation methods for resizing an image.

Choice of Interpolation Method for Resizing –

- cv2.INTER_AREA: This is used when we need to shrink an image.
- cv2.INTER_CUBIC: This is slow but more efficient.
- cv2.INTER_LINEAR: This is primarily used when zooming is required. This is the default interpolation technique in OpenCV.

We resized our images in both train and test data to get better accuracy into same size with shape 91,109,91

3.3.2 Image Denoising using OpenCV3

Image denoising is one of the most complicated part of image processing. Denoising means the elimination of noise from an image and calculate the real image. There are different reasons for causing noise in image and often some are impossible to neglect. As a result, denoising plays a vital role in various applications. For instance, image segmentation, image restoration, etc. For image denoising, a lot of algorithms are offered but the issue itself still remains a challenge till this date. It becomes even more challenging when the level of noise is very high and thus images are in very bad condition.
An image contains two major types of noise:

1. **Salt and pepper noise:** Pixels in these types of images vary in intensity and color from the adjacent pixels. The main nature is that a noisy pixel's value has no connection with the adjacent pixel's color. Usually, these noises will impact a tiny amount of image pixels only. The name salt and pepper noise comes from the white and dark dots that the image includes when it is viewed.
2. **Gaussian noise:** Every pixel of these images is changed by a little extent from its actual value. A histogram depicts usual noise distribution by plotting

a pixel value's distortion amount against frequency. Although there are a lot of distribution that are feasible, Gaussian distribution is comparatively a better model generally. As sum of various noises has a tendency to go towards Gaussian distribution.

In our data we used a function called **cv2.fastNlMeansDenoisingColored()**. It is the execution of **Non-local Means Denoising** which is a denoising algorithm. That's how it is defined:

cv2.fastNlMeansDenoisingColored(src[,dst[,h[,hColor[,templateWindow Size[,searchWindowSize]]]]])

Here,

- **src:** The input image is of 8-bit containing 3-channel.
- **dst:** The type and size of the output image is exactly similar to input.
- **h:** Parameter to control the brightness component's filter power. The bigger the value of h the more precisely eliminates noise. Although it eliminates some details of image. The smaller the value of h the more it sustains details. Then again it also sustains some noise.
- **templateWindowSize:** Template patch size for computing weights. Odd number is preferable. Suggested value is 7 pixels.
- **searchWindowSize:** Window size for computing weighted average. Odd number is preferable. Has linear impact on performance: the bigger the search-WindowSize - the bigger the denoising time. Suggested value is 21 pixels.

3.3.3 Output

These are the picture before fig:3.3 and after denoising fig:3.4

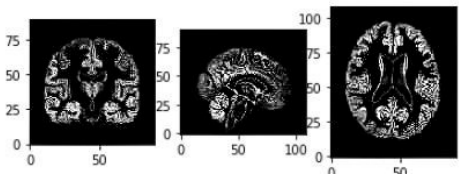

Figure 3.3: Before Denoising.

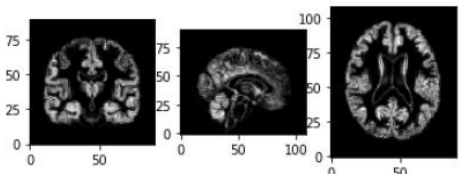

Figure 3.4: After Denoising.

3.4 Data Labelling

As we have mentioned earlier in our oasis dataset, we have a total number of 416 patients in which we have both demented and non-demented patients. In demented patients we have clinical dementia rating (CDR) of below 1, equals to 1 and above 1. It is to mention that in oasis dataset for every patient there are two image files. For our neural network we have used binary classification. In order to do that we first process the data using image-processing and then labeled the data in demented with CDR 1 and non-demented with CDR 0 by taking information from the csv file of the oasis dataset. It is important to note that for binary classification number of data in both of the part of the classification should be equal for better accuracy. Specially if the sample size is small. There were 28 patients who was demented with CDR 1. In that case we had 56 image files of demented with CDR1.So to make it equal we also took 56 image files in non-demented part. In both of the cases we split data at a percentage of 80% for train data and 20% for test data fig:3.5. So, in train dataset we have 40 image files in both demented and non-demented part. In test file we have 16 image files in both demented and non-demented part.

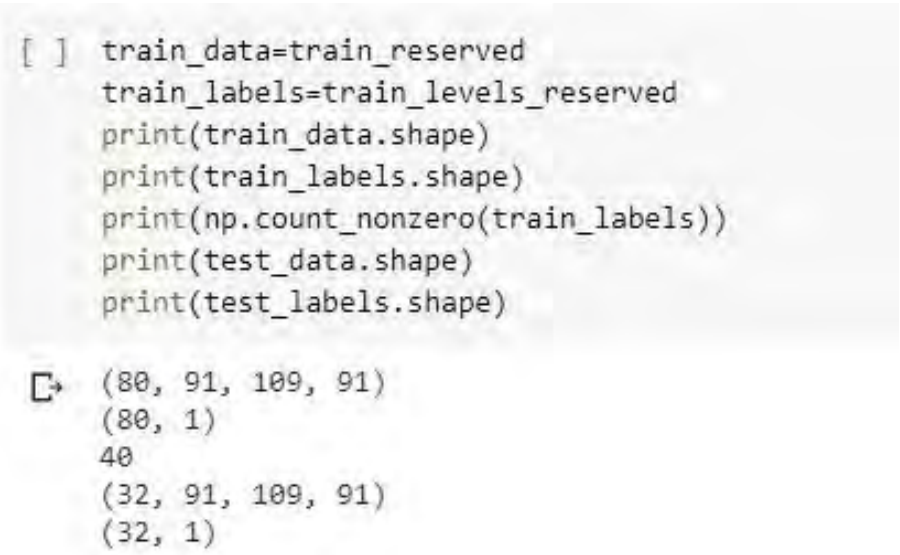

Figure 3.5: Labelled Data.

Chapter 4

Implementation and Result

This chapter provides a complete overview of the models that we have implemented and the outcome of our thesis.

4.1 Applying proposed CNN Architectures

After pre-processing the data, we applied our self-designed CNN model and other CNN models to calculate the accuracy, Confusion matrix from which we will see precision, recall and F1 score, false positive rate area under ROC curve, model accuracy and model loss graph.

Accuracy equation [31]:

$$\begin{aligned} ACC &= \frac{TP + TN}{P + N} \\ &= \frac{TP + TN}{TP + TN + FP + FN} \end{aligned} \tag{4.1}$$

Where-

- ACC = Accuracy
- TP = True Positive
- TN = True Negative
- P = Condition Positive
- N = Condition Negative

F1 score equation [31]:

$$\begin{aligned} F1 &= 2 \times \frac{PPV \times TPR}{PPV + TPR} \\ &= \frac{2TP}{2TP + FP + FN} \end{aligned} \tag{4.2}$$

Where-

- PPV = Positive Predictive Value/ Precision
- TPR = True Positive Rate
- TP = True Positive
- FP = False Positive
- FN = False Negative

ROC Curve: ROC is a probability curve that is used for performance measurement for classification problem at various thresholds settings.
The graph fig: 4.1 depicts 3 ROC curves that represent 3 categories which are: excellent, good, and worthless. The test's accuracy relies on how properly it distinguishes the test group from those with and without the disease. Area under the ROC curve measures the accuracy. A test is perfect if an area represents 1. Similarly, a test is worthless if an area represents 0.5. The traditional academic point system is the rough guide that is used for classification of the diagnostic test accuracy.

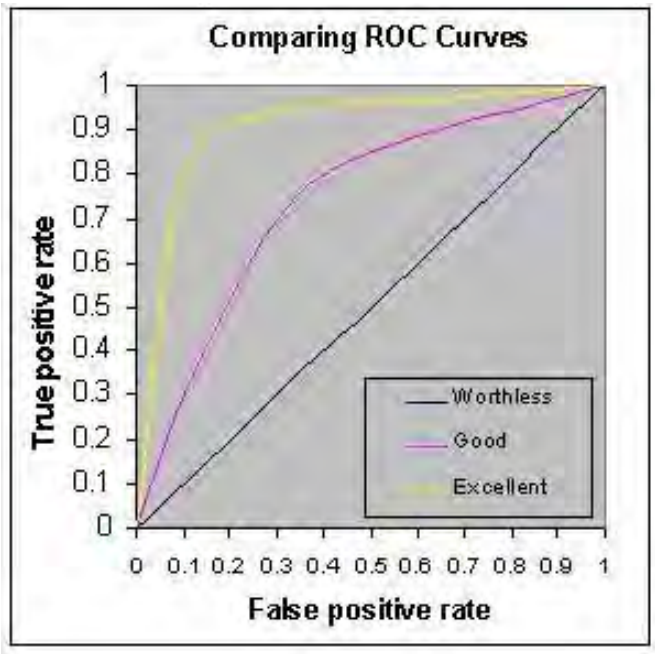

Figure 4.1: Comparing ROC Curve [40].

Depending on the area under the ROC curve, the following categories are defined in table: 4.1

Table 4.1: ROC Curve Score.

Score	Decision
0.9-1.0	Excellent(A)
0.8-0.9	Good (B)
0.7-0.8	Fair (C)
0.6-0.7	Poor (D)
0.5-0.6	Fail (F)

Precision equation [31]:

$$\begin{aligned} PPV &= \frac{TP}{TP + FP} \\ &= 1 - FDR \end{aligned} \tag{4.3}$$

Where-

- FDR = False Discovery Rate
- PPV = Positive Predictive Value/ Precision
- TP = True Positive
- FP = False Positive

Recall equation [31]:

$$\begin{aligned} PPV &= \frac{TP}{TP + FN} \\ &= 1 - FDR \end{aligned} \tag{4.4}$$

Where-

- FDR = False Discovery Rate
- PPV = Positive Predictive Value/ Precision
- TP = True Positive
- FN = False Negative

4.2 Self designed CNN architecture

4.2.1 Self designed CNN architecture Concept

Convolutional layer selection

In our self-designed CNN model, we used Conv2D. This layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs. When using this layer as the first layer in a model, provide the keyword argument input shape (tuple of integers, does not include the batch axis). Our input shape was 91 109 91. We used total 4 conv2D layer in our model.

Pooling layer selection

In this model we have used Maxpooling2D.There are different pooling layer. Max-Pooling is one of most commonly used. There is another commonly used pooling layer which is AvGPooling2D. If we need to detect patterns from all over the image, in that case average pooling is a better option as it takes average value of the pixel from the defined matrix. But as we are detecting infections from a body part of a patient which only a portion of the whole image, we are using max pooling. Besides, max pooling is better choice here because it extracts the sharpest features of an image. It extracts the most important features like edges of patterns whereas, average pooling extracts features smoothly. If there is no defined parameters keras takes pool size 2x2 and a stride value of 2 by default. However, we have also used the pool size 2x2 which means from a 2x2 matrix, max pooling will take the highest pixel value and replace it with one single pixel for the output image. With each Conv2D layer we have used a Maxpool2D layer. Which means there are 4 MaxPool2D layer.

Flatten Layer

In our model After using the pooling layer we used a flatten layer to flatten the whole network. Flattening transforms the whole pooled feature map matrix into a single column. Then this is being passed to the neural network for further processing.

Dense Layer

After the flatten layer we have used two dense layers. Dense layer is a linear operation on the layer's input vector. Dense layer is also known as Fully connected Layer.

Activation Function

In our model we have used two different activation functions.

- Sigmoid function
- ReLU function

Sigmoid Function: Expression of Sigmoid function is [17]-

$$f(x) = \frac{1}{1 + exp(-x)} \tag{4.5}$$

This is a curve which is s-shaped and it ranges from 0 and 1. Although the activation function is easy to apply it has a problem that is called vanishing gradient problem. And the updates of gradient go very distant in various directions as its not zero centered output. Thus makes it difficult in case of optimization.

RelU Function Expression of ReLU is [17]-

$$ReLU = max(0, x) \tag{4.6}$$

This function is easy to use and effective. It also doesn't have vanishing any gradient problem. These reasons make this function suitable for our models. Although there is one problem which is its usable only in case of hidden layers. But its performance is better than the other activation functions.
In our model we have used RelU function with each Conv2D layer and with one Dense Layer. We have used Sigmoid function with another Dense layer.

4.2.2 Result of our self-designed CNN architecture

After our model was ready, we calculated the following things from our data:

Accuracy

In our model with 50 epochs and batch size 10 we have an accuracy of 90.62%

Precision, Recall and f1 interpretation:

Precision: We have got precision in demented data 100% and non demented 84% data which represents good performance of the model.
Recall: We have got recall of 81% in demented data and recall of 100% non-demented data which is good for this model.
F1 score: In our case, F1 score is 90% for demented data and 91% for demented data which is good for this model.

Table 4.2: Precision, Recall and f1 Score for basic CNN.

	Precision	**Recall**	**f1-Score**
Demented	1.00	0.81	0.90
Non-Demented	0.84	1.00	0.91

Confusion Matrix Interpretation:

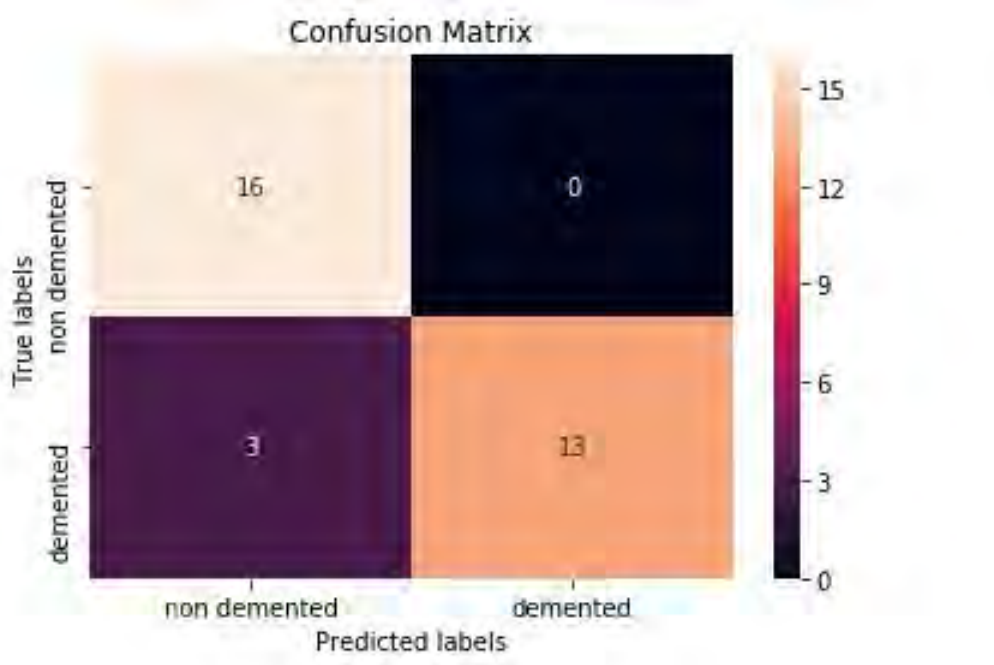

Figure 4.2: Confusion matrix interpretation of CNN architecture.

If we interpret this confusion matrix fig: 4.2, we can say that from our test data among 16 non demented images it predicted 16 properly. No wrong prediction here. On the other hand, among 16 demented images it predicted 13 properly and wrong predicted 3. Overall performance of the model is very good we can say from this performance analysis.

Area under ROC curve:

In our model the percentage of the area under ROC curve is 99.21% fig: 4.3 according to table: 4.1 which falls under very good category.

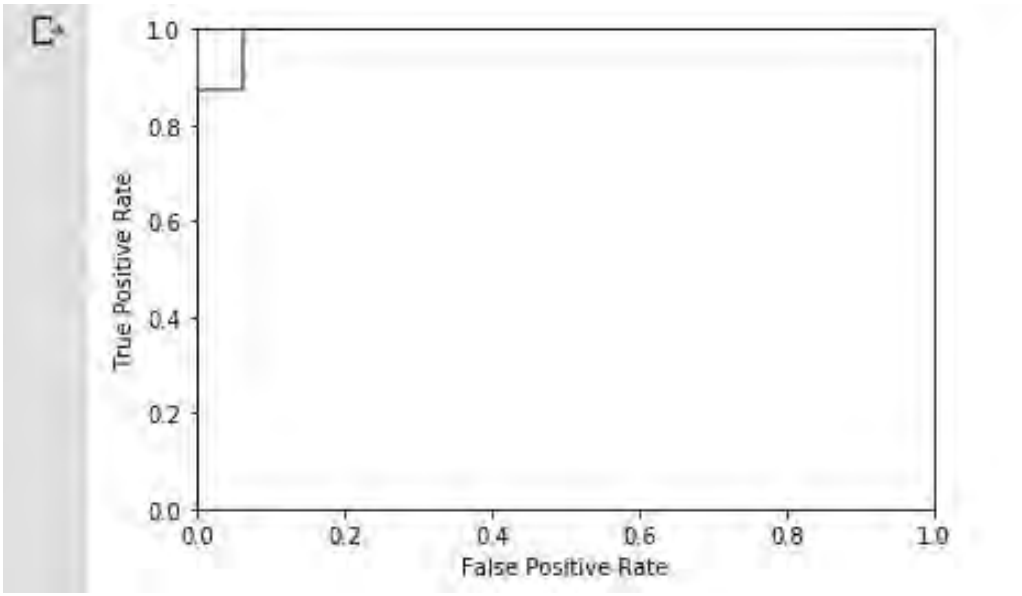

Figure 4.3: Area under ROC curve of CNN architecture.

Model accuracy and model loss:

From the model accuracy fig: 4.4 and model loss fig: 4.5 we can see that the model is not underfit or overfit. Which means model has no high biasness. Which we can also determine from the confusion matrix.
Overall from accuracy, confusion matrix fig: 4.2, ROC curve, model accuracy and model loss figure we can say that performance of the model is good.

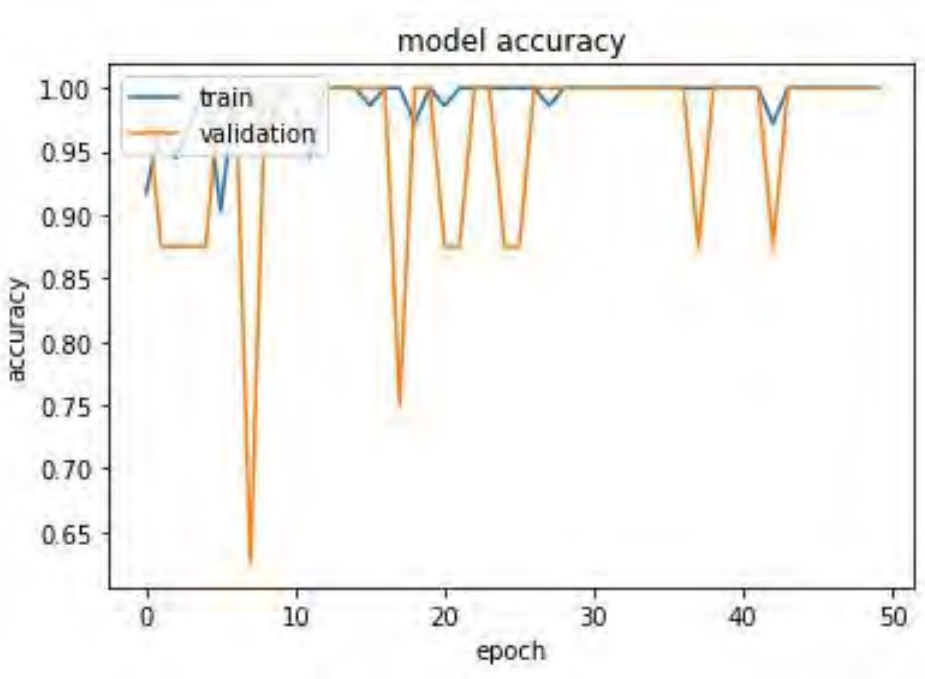

Figure 4.4: Model Accuracy of CNN architecture.

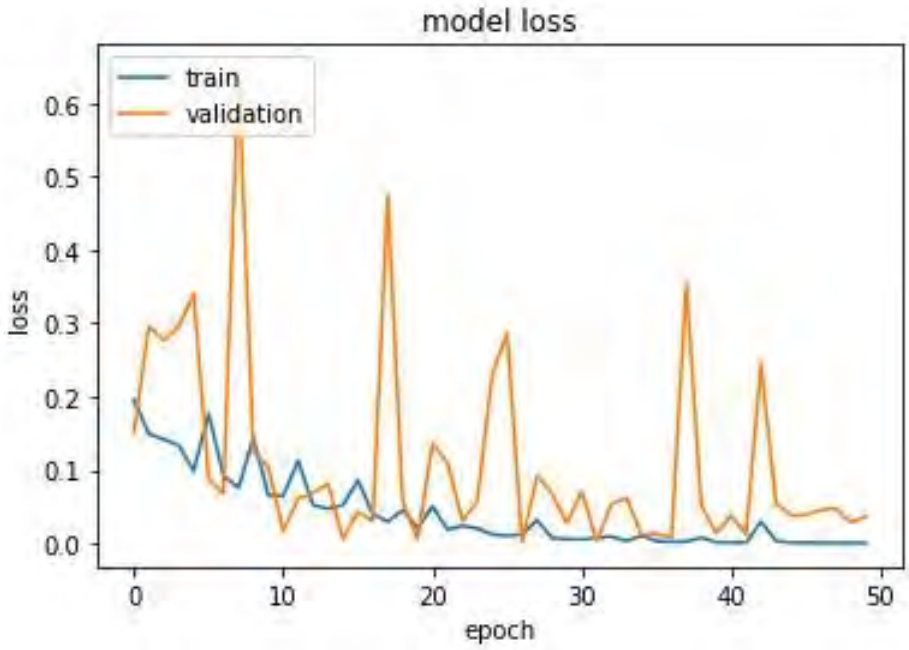

Figure 4.5: Model Loss of CNN architecture.

4.3 Inception

4.3.1 Inception concept

The inception network was a significant creation in the improvement of CNN classifiers. It is a very deep convolutional neural network architecture. After its innovation, most popular convolutional neural networks' convolutional layer stacked deeper in order to get improved performance. It's a complex architecture because it uses various tricks to improve its accuracy and speed. It has 1x1 convolution in the network. And instead of fully connected layers it uses global average pooling [25]. Also, for the same input it has different types and sizes of convolutions. The size of the noticeable part of an image can have large variation. Because of this large variation, choosing the correct kernel size is difficult. For globally distributed information, a larger kernel size is preferable and for locally distributed information, a smaller kernel is preferable. Moreover, over-fitting is another big problem for very deep networks. Gradient updates are difficult to pass through the network. In these cases, inception becomes handy. In inception, multiple sized filters operate on the same level.

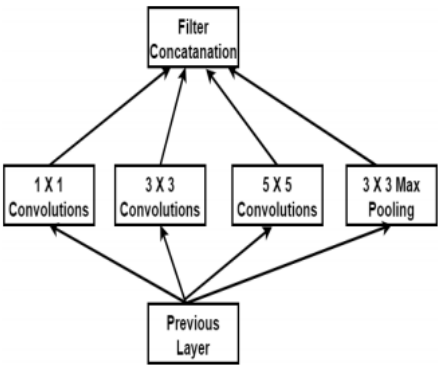

Figure 4.6: Naive version of Inception [22].

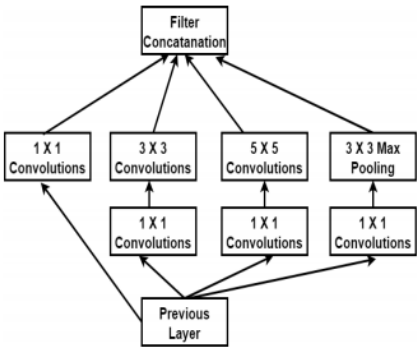

Figure 4.7: Dimension reductions of Inception[22].

3 different sizes (1x1 , 3x3, 5x5) of filters are used for performing convolution on an input [22]. Adding an extra 1x1 convolution, it limits the number of input channels. It reduces the model size and thus the over-fitting problem is also reduced [25]. The max pooling is introduced before the 1x1 convolution. The outputs are fed into the next module after being concatenated.The rapid evolution of inception lead to implementation of several versions of inception which are Inception-v1, Inception-v2, Inception-v3, Inception-v4.

4.3.2 InceptionV3 concept

Inception-v3 is the third version of Inception architecture. It is the upgraded version of inception-v2. The architecture fig: 4.8 was first introduced in 2015 [16]. It is a popular CNN module that is used for pattern identification of images. It has a smaller weight than VGG and ResNet [16]. This architecture is 42-layers deep and obtains lower error rate [26]. Auxiliary classifiers that were suggested InceptionV1 were modified in InceptionV3. In InceptionV1 auxiliary classifiers were used to have deeper network whereas in Inception-v3 it was used as regularizer. The contribution of the auxiliary classifiers weren't much until near the end of the training process, when the accuracy reaches near a saturation state. They especially functioned as regularizers when there are Dropout operations or BatchNorm present. So, Inception-v3 was updated keeping all the updates of Inception-v2 and the additions were such as RMSProp Optimizer, Factorized 7x7 convolutions, BatchNorm in the Auxiliary Classifiers, Label Smoothing, etc.

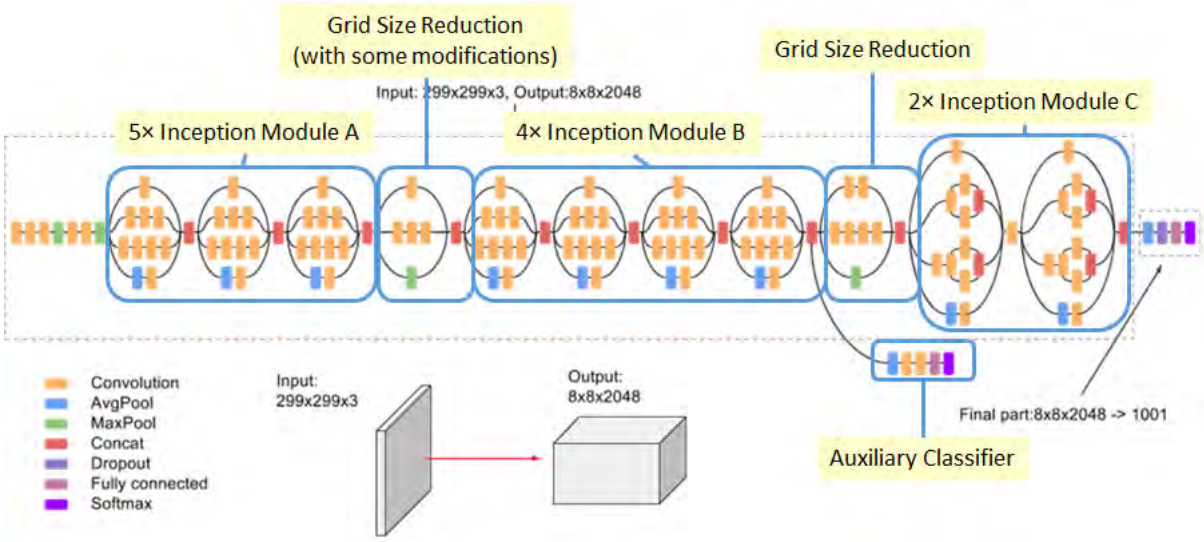

Figure 4.8: InceptionV3 Architecture [26].

4.3.3 Result of InceptionV3 architecture in our Dataset

Accuracy

In InceptionV3 model with 50 epochs and batch size 10 we got the accuracy of 90.62%

Precision, Recall and f1 interpretation:

Though the accuracy is same as basic CNN but we got a slightly different result when we plot the confusion matrix.

Precision: We have got precision in demented data 93% and non demented 88% data which represents good performance of the model.

Recall: We have got recall of 88% in demented data and recall of 94% non-demented data which is good for this model.

F1 score: In our case, F1 score is 90% for demented data and 91% for demented data which is good for this model.

Table 4.3: Precision, Recall and f1 Score for InceptionV3.

	Precision	**Recall**	**f1-Score**
Demented	0.93	0.88	0.90
Non-Demented	0.88	0.94	0.91

Confusion matrix interpretation:

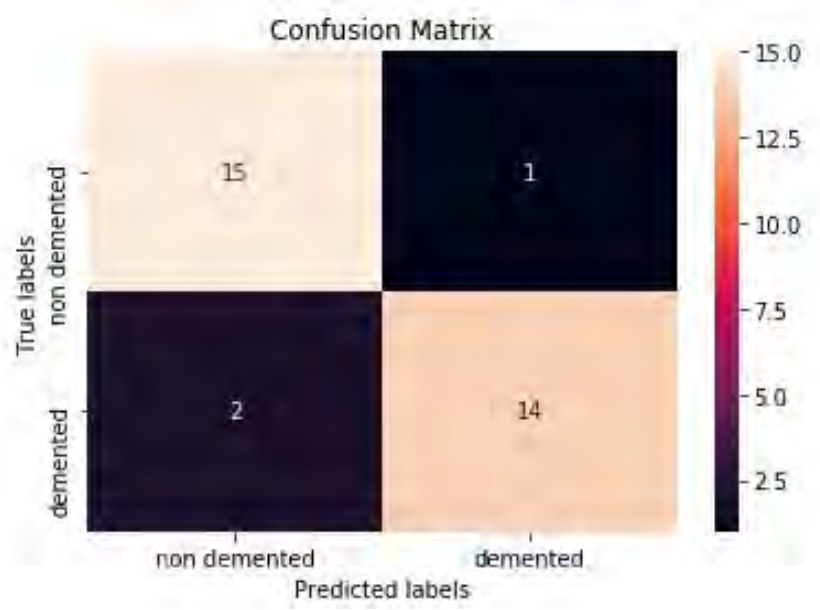

Figure 4.9: Confusion matrix interpretation of InceptionV3.

If we interpret this confusion matrix fig: 4.9, we can see that among 16 non demented images it predicted 15 correctly and 1 incorrectly. On the other hand, among 16 demented images it predicted 14 correctly and 2 incorrectly. Though the accuracy was similar to our self-designed architecture performance analysis is slightly different. We also got different result in case of precision and recall, ROC curve, model accuracy and model loss.

Area under ROC curve:

In our model the percentage of the area under ROC curve is 96.48% fig: 4.10 according to table: 4.1 which falls under very good category

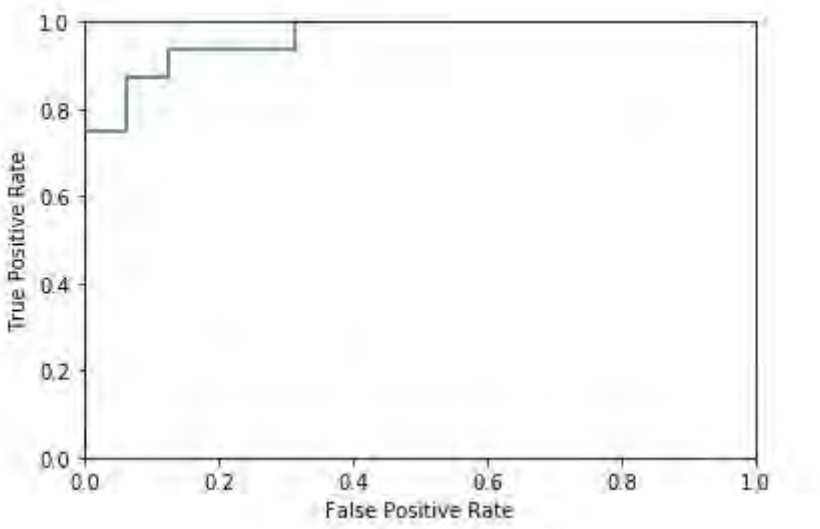

Figure 4.10: Area under ROC curve of InceptionV3.

Model accuracy and model loss:

From the model accuracy fig: 4.11 and model loss fig: 4.12 we can see that the model is not underfit or overfit. Which means model has no high biasness. Which we can also determine from the confusion matrix fig: 4.9.
Overall from accuracy, confusion matrix, ROC curve, model accuracy and model loss figure we can say that performance of the model is good.

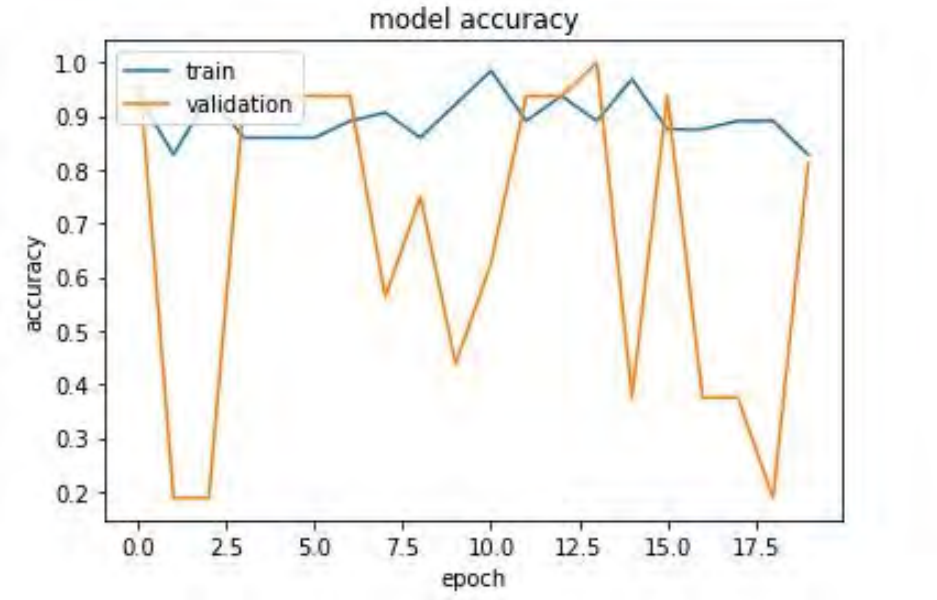

Figure 4.11: Model Accuracy of InceptionV3.

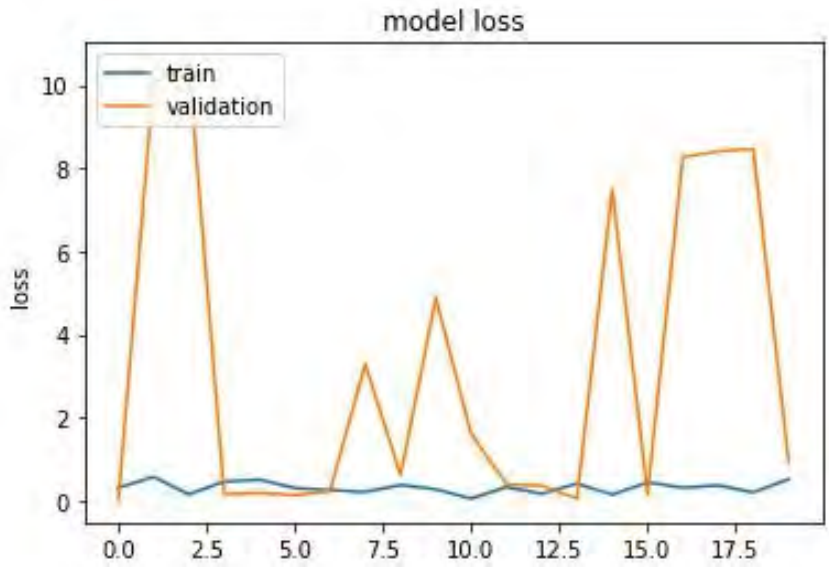

Figure 4.12: Model Loss of InceptionV3.

4.4 Xception

4.4.1 Xception concept

Extreme version of Inception (Xception) is an extended version of the inception architecture [28]. It is a deep convolutional neural network architecture. It has 36 convolutional layers that forms the feature extraction base of the network [34]. It involves depth-wise separable convolutions fig: 4.13 that replaces the standard inception modules. The depth-wise separable convolution layers forms a linear stack that has residual connections. Depth-wise convolution refers to channel-wise nxn spatial convolution [28]. That means for 5 channel there will be 5 nxn spatial convolution. And point-wise convolution refers to 1x1 convolution that is being used for changing dimension. In Xception, the modified depth-wise separable convolution fig: 4.14 is used. It is actually the point-wise convolution which is followed by a depth-wise convolution. That means, 1x1 convolution will be prior to any nxn spatial convolutions.This modification is inspired by the inception module. There is no intermediate ReLU non-linearity in the modified depth-wise separable convolution [28]. Xception outperforms InceptionV3, ResNet, VGG on various occasions.

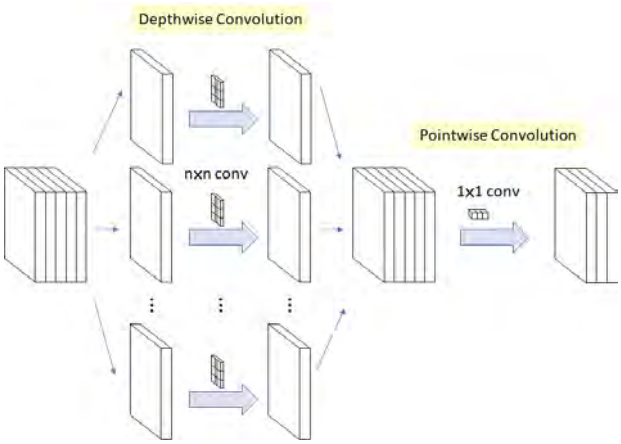

Figure 4.13: Original Depth-wise Separable Convolution [28].

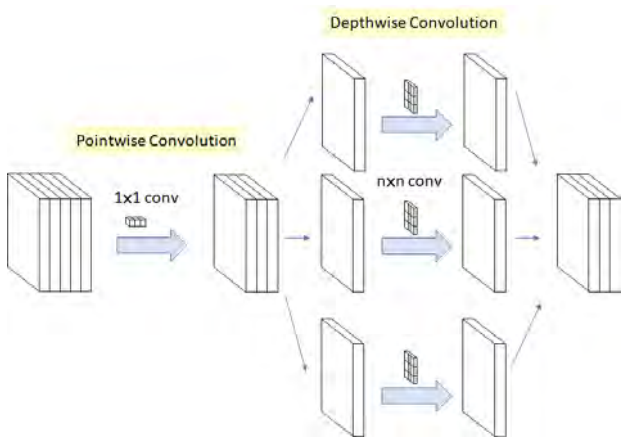

Figure 4.14: The Modified Depth-wise Separable Convolution [28].

4.4.2 Result of Xception architecture in our Dataset

Accuracy

In Xception model with 50 epochs and batch size 10 we got an accuracy of 84.375%
Precision, Recall and f1 interpretation:
Precision: We have got precision in demented data 100% and non demented 76% data which represents good performance of the model.
Recall: We have got recall of 69% in demented data and recall of 100% non-demented data which is good for this model.
F1 score: In our case, F1 score is 81% for demented data and 86% for demented data which is good for this model.

Table 4.4: Precision, Recall and f1 Score for Xception.

	Precision	**Recall**	**f1-Score**
Demented	1.00	0.69	0.81
Non-Demented	0.76	1.00	0.86

Confusion matrix interpretation:

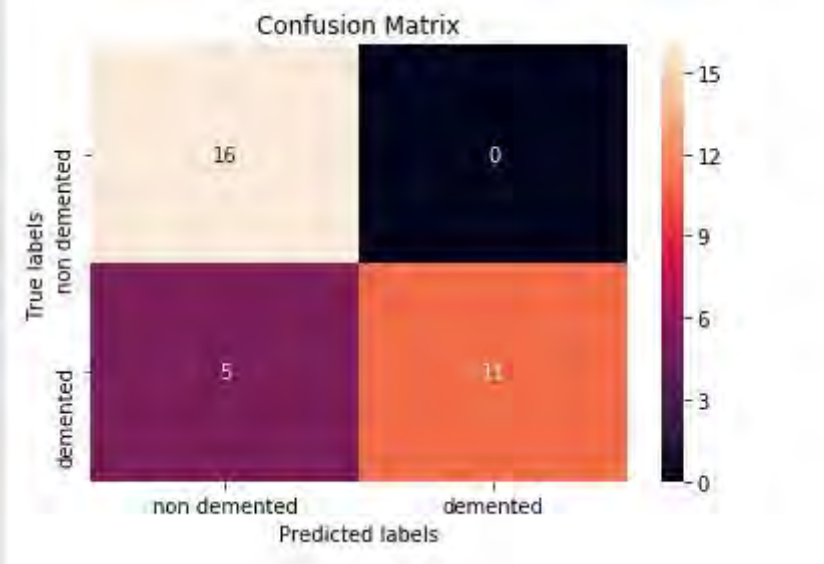

Figure 4.15: Confusion matrix interpretation of Xception.

If we interpret this confusion matrix fig: 4.15, we can see that among 16 non demented images it predicted 16 correctly and 0 incorrectly. On the other hand, among 16 demented images it predicted 11 correctly and 5 incorrectly.

Area under ROC curve:

In this model the percentage of this area that is under this ROC curve 96.48% fig: 4.16. According to fig: 4.1 which also falls under very good category.Though area under the roc curve is similar to inception but the graph is slightly different.

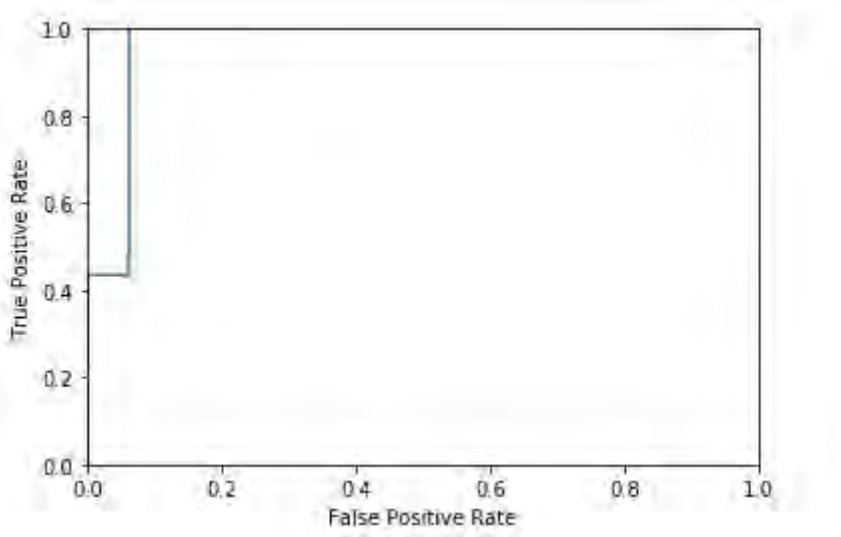

Figure 4.16: Area under ROC curve of Xception.

Model accuracy and model loss:

From the model accuracy fig: 4.17 and model loss fig: 4.18, we can see that the model is not underfit or overfit. Which means model has no high biasness. Which we can also determine from the confusion matrix fig: 4.15.
Overall from accuracy, confusion matrix ROC curve, model accuracy and model loss figure we can say that performance of the model is good.

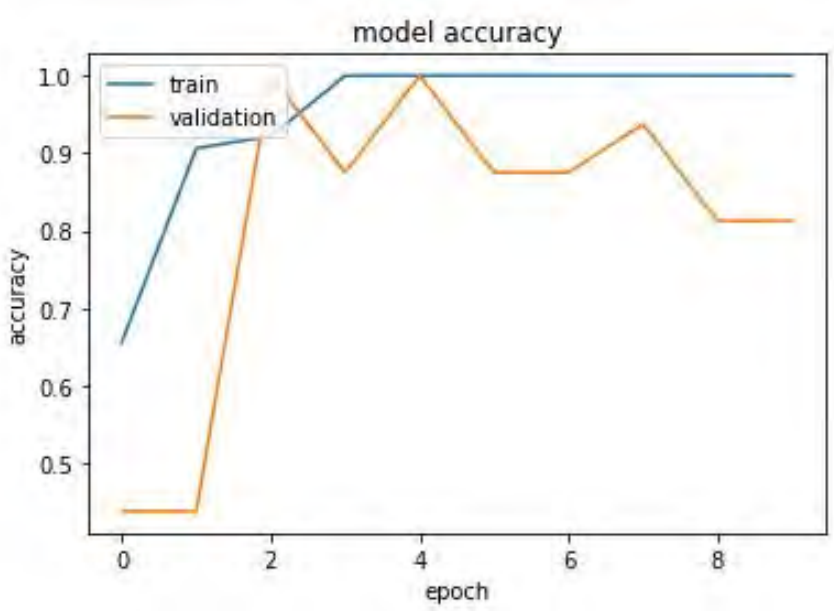

Figure 4.17: Model Accuracy of Xception.

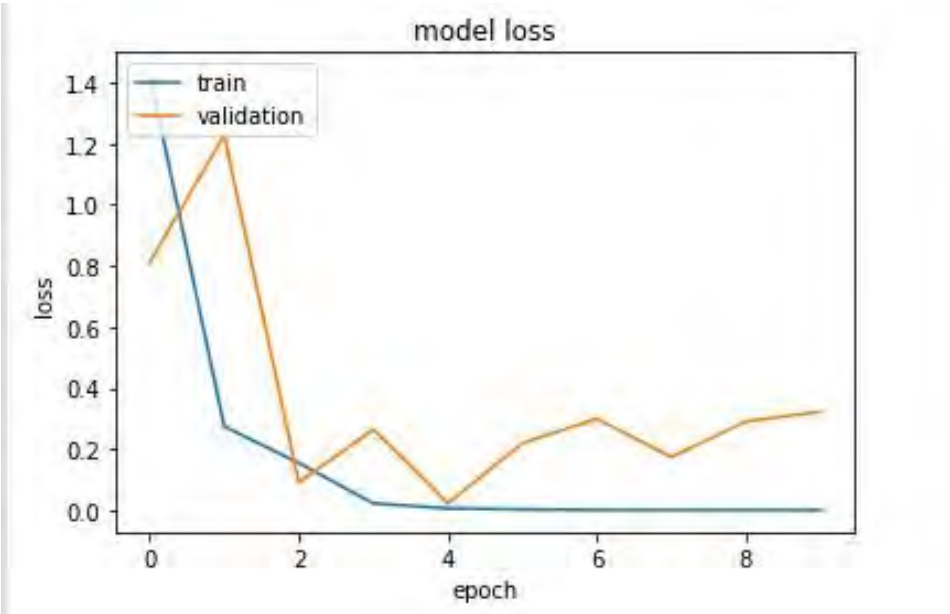

Figure 4.18: Model Loss of Xception.

4.5 MobileNet

4.5.1 MobileNet concept

MobileNet is specifically handy in embedded version applications and mobile. This architecture fig: 4.19 is basically a stack of separable convolution modules. These separable convolution modules are composed of Conv1x1 and depth-wise convolution [29]. The convolution in spatial and channel domains are performed by these separation convolution modules. The reduction of model size and complexity is the main function of depth-wise separable convolution [27]. The two main features of MobileNet are smaller model size (less parameters) and smaller complexity (fewer additions and multiplications). Two parameters respectively Width Multiplier α and Resolution Multiplier ρ are used for easily tuning MobileNet.

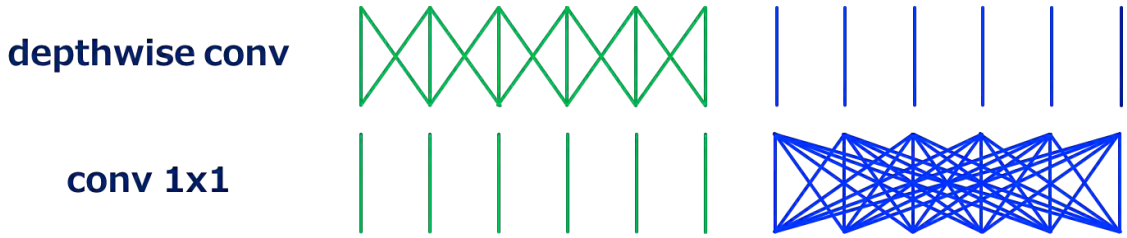

Figure 4.19: MobileNet Architecture [29].

4.5.2 MobileNetV2 concept

MobileNetV2 fig: 4.20 is a remarkably improved version of MobileNetV1. It improves the state-of-the-art for mobile visual recognition including object detection, classification and semantic segmentation [24]. It was built from the basic idea of MobileNetV1. It uses depth-wise separable convolution which is the same as MobileNetV1. However, two new architectures are introduced in this architecture which are linear bottlenecks between layers and shortcut connections between the bottlenecks [24]. The inputs and outputs of the model are encoded by the bottlenecks. And the model's ability to change from lower-level concepts (e.g. pixels) to higher level descriptors (e.g. image categories).The MobileNetV2 has an improved module called inverted residual structure [32]. As a result, it has better accuracy and faster training and non-linearities are removed from the narrow layers.

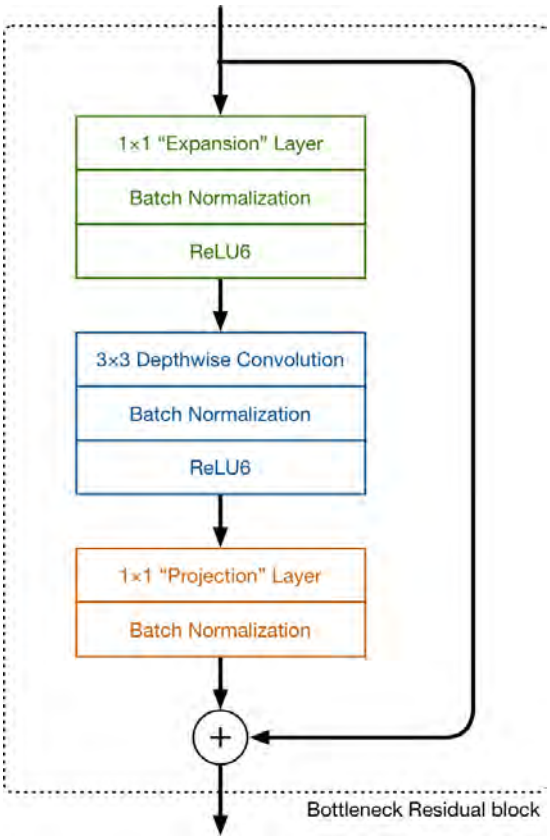

Figure 4.20: MobileNetV2 [18].

4.5.3 Result of MobileNetV2 architecture in our Dataset

Accuracy

In MobileNetV2 model with 50 epochs and batch size 10 we got an accuracy of 81.24%.

Precision, Recall and f1 interpretation:

Precision: We have got precision in demented data 92% and non demented 75% data which represents good performance of the model.

Recall: We have got recall of 69% in demented data and recall of 94% non-demented data which is good for this model.

F1 score: In our case, F1 score is 79% for demented data and 83% for demented data which is good for this model.

Table 4.5: Precision, Recall and f1 Score for MobileNetV2.

	Precision	**Recall**	**f1-Score**
Demented	0.92	0.69	0.79
Non-Demented	0.75	0.94	0.83

Confusion matrix interpretation:

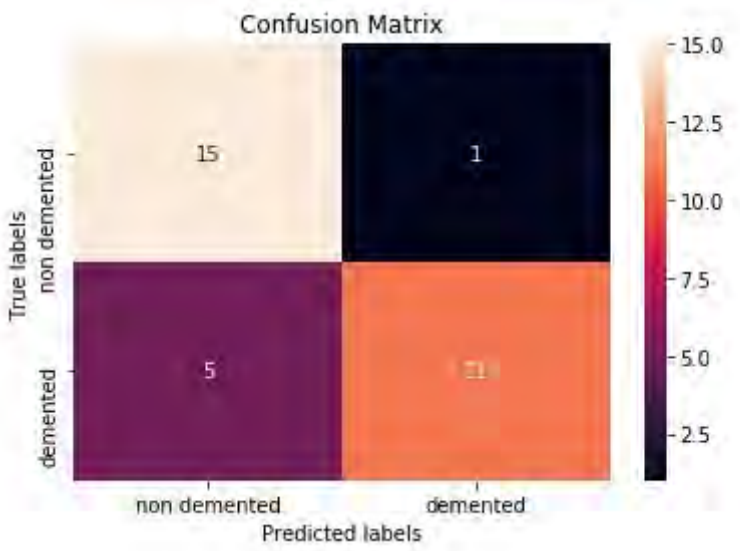

Figure 4.21: Confusion matrix interpretation of MobileNetV2.

If we interpret this confusion matrix fig: 4.21, we can see that among 16 non demented images it predicted 15 correctly and 1 incorrectly. On the other hand, among 16 demented images it predicted 11 correctly and 5 incorrectly.

Area under ROC curve:

In our model the percentage of the area under roc curve is 85.15% fig: 4.22. According to table: 4.1 which falls under good category.

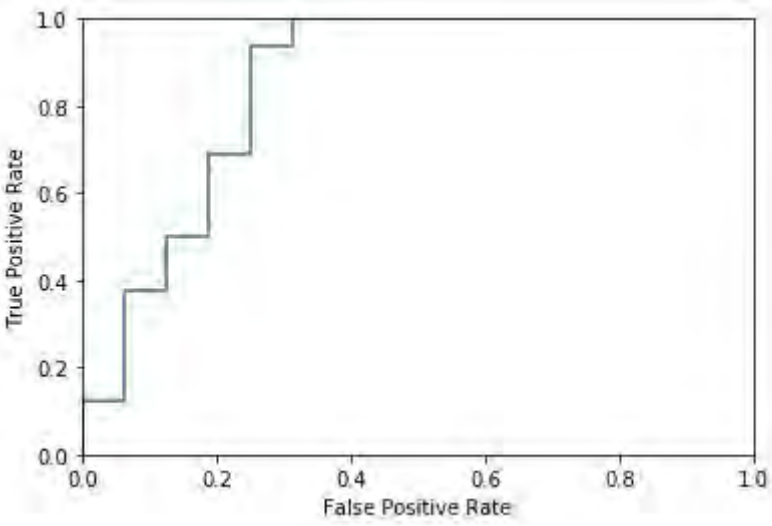

Figure 4.22: Area under ROC curve of MobileNetV2.

Model accuracy and model loss:

From the model accuracy fig: 4.23 and model loss fig: 4.24, we can see that the model is not underfit or overfit. Which means model has no high biasness. Which we can also determine from the confusion matrix fig: 4.21].
Overall from accuracy, confusion matrix roc curve, model accuracy and model loss figure we can say that performance of the model is good.

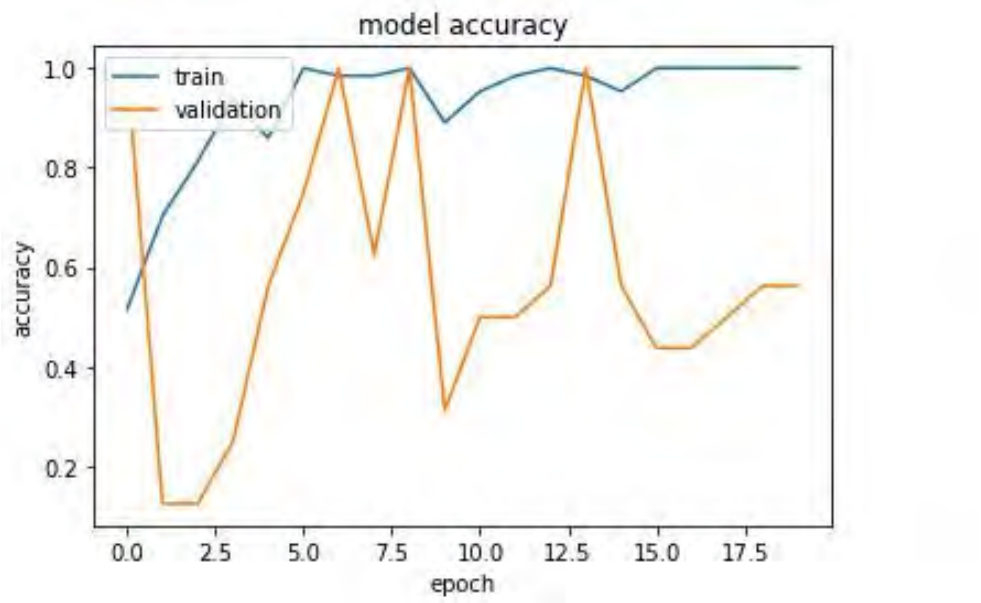

Figure 4.23: Model Accuracy of MobileNetV2.

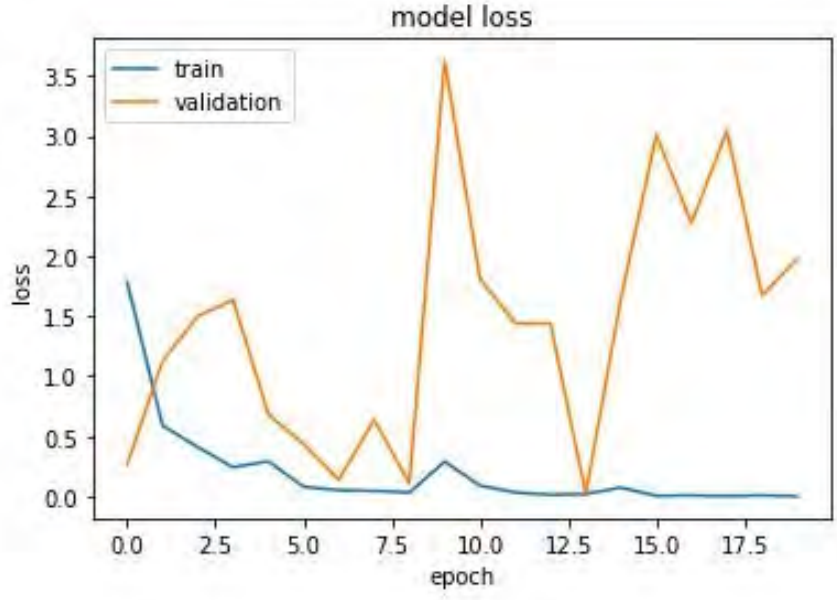

Figure 4.24: Model Loss of MobileNetV2.

4.6 VGG

4.6.1 VGG Concept

VGG neural network is one of the most famous models for image classification and object recognition. It is also known for its simplicity. While AlexNet emphasized on the strides and smaller window sizes, VGG focused on another very significant feature of CNNs that is depth. It has 19 layers. To increase depth, it uses 3x3 convolutional layers and stack them on top of each other. Max pooling handles the reduction of volume size fig: 4.25.

Input: It takes RGB images of 224x224 pixel size.

Convolutional Layers: Very small (3x3 ; smallest size yet capable of capturing left,right,up,down) receptive field are used in the convolutional layers of VGG. There are 1x1 convolution filters in VGG that functions as the linear transformation of input. It is followed by a ReLU unit. 1 pixel is fixed for the convolution stride for preserving the spatial resolution after convolution [33].

Fully-Connected Layers: There are 3 fully-connected layers in VGG that respectively has 4096, 4096 and 1000 channels.

Hidden Layers: Hidden layers of VGG use ReLU. Local Response Normalization (LRN) fails to increase any particular accuracy instead increases training time and memory consumption. For this reason, VGG does not use LRN in general. VGG uses small receptive fields that is 3x3 with a stride of 1 because there are three ReLU, so it has more discriminative function. It has fewer parameters which is 27 times the number of channels [33]. 1x1 convolutional layers are incorporated by VGG in order to make more non-linear decision function without the alteration of the receptive fields. Moreover, a huge number of weight layers are possible in VGG because of small-size convolution filters. VGG outperforms many other models and one of the most popular architectures for image-recognition.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 4.25: VGG configuration [33].

4.6.2 VGG16 Concept

The "16" in VGG16 means that it has 16 weight layers in the network fig: 4.26. It was a very famous model and considered very deep in 2014. It replaces the large kernel-sized filters of AlexNet (11 and 5 respectively in the 1st and 2nd convolutional layer) with several 3x3 kernel-sized filters that are put one on top of another in order to make improvements over AlexNet [30]. Training VGG16 is very difficult and tedious task. So usually smaller version is trained first to make things easier. This process is known as pre-training [16]. It is also very time consuming. The entire network is needed to be trained before it can perform as an initialization a deeper network. Moreover, the size of VGG16 is over 533MB because of its number of fully-connected layers and depth which also make deploying it a tedious task. However, VGG16 is still used in many image-classification of deep neural network but the smaller architectures like GoogleNet are more preferable

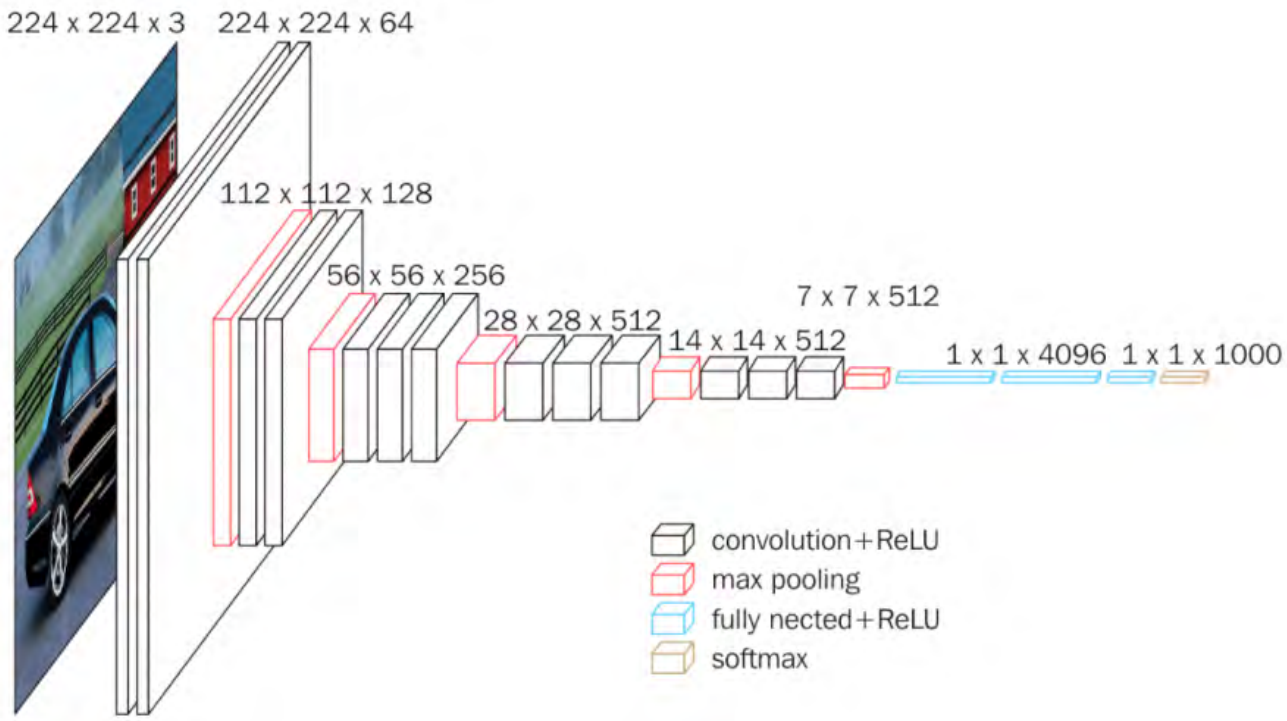

Figure 4.26: VGG16 Architecture [30].

4.6.3 Result of VGG16 and VGG19 architecture in our Dataset

Accuracy

Among all the CNN architectures that we applied VGG16 and VGG19 gave us the worst result in case of accuracy, precision and recall, confusion matrix, ROC curve, model accuracy and model loss. With 50 epochs and batch size of 10 both gave us accuracy of only 50.00%

Precision, Recall and f1 interpretation:

Precision: We have got precision in demented data 0% and non demented 58% data which represents poor performance of the model.

Recall: We have got recall of 0% in demented data and recall of 100% non-demented data which is very poor for this model.

F1 score: In our case, F1 score is 0% for demented data and 67% for demented data which is extremly poor for this model.

Table 4.6: Precision, Recall and f1 Score for VGG16 and VGG19.

	Precision	**Recall**	**f1-Score**
Demented	0.00	0.00	0.00
Non-Demented	0.58	1.00	0.67

Confusion matrix interpretation:

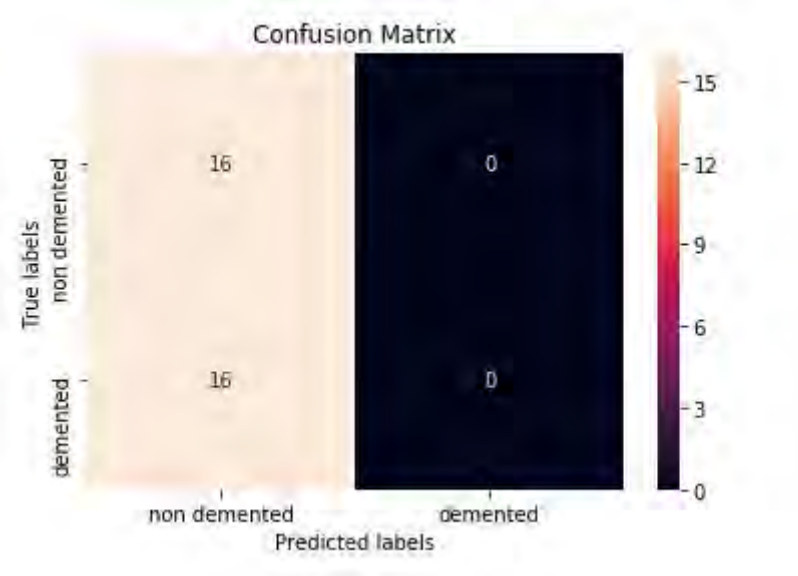

Figure 4.27: Confusion matrix interpretation of VGG16 and VGG19.

As we can see result of confusion matrix fig: 4.27 of both VGG16 and VGG19 is not good at all.If we interpret this confusion matrix, we can see that among 16 non demented images it predicted 16 correctly and 0 incorrectly. On the other hand, among 16 demented images it predicted 0 correctly and 16 incorrectly.

Area under ROC curve:

In our model the percentage of the area under ROC curve is 50% fig: 4.28. According to table: 4.1 which falls under worthless category.

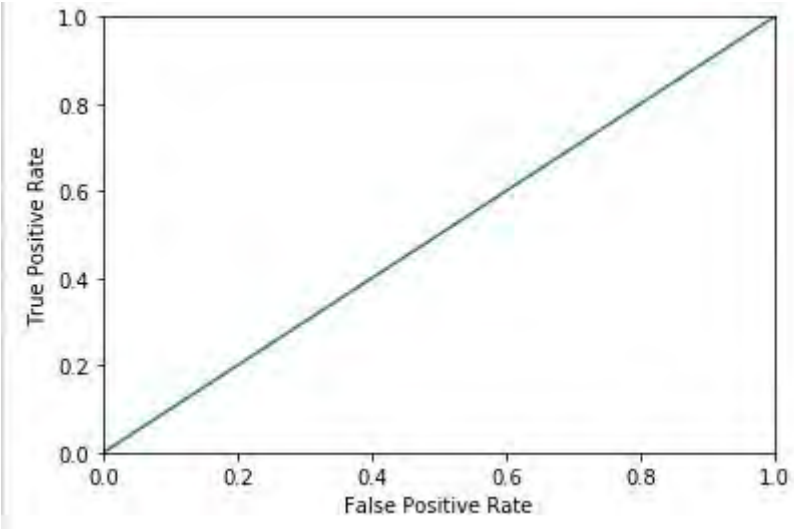

Figure 4.28: Area under ROC curve of VGG16 and VGG19.

Model accuracy and model loss:

From the model accuracy fig: 4.29 and model loss fig: 4.30, we can see the model is under-fit. It means the model has ha high biasness which we can also see from the confusion matrix fig: 4.27]. Model is highly biased to non-demented data as it predicted all of them correctly but predicted all the demented data incorrectly.

Overall from accuracy, confusion matrix, ROC curve, model accuracy and model loss figure we can come to the conclusion that VGG16 and VGG19 model does not perform well on our data at all.

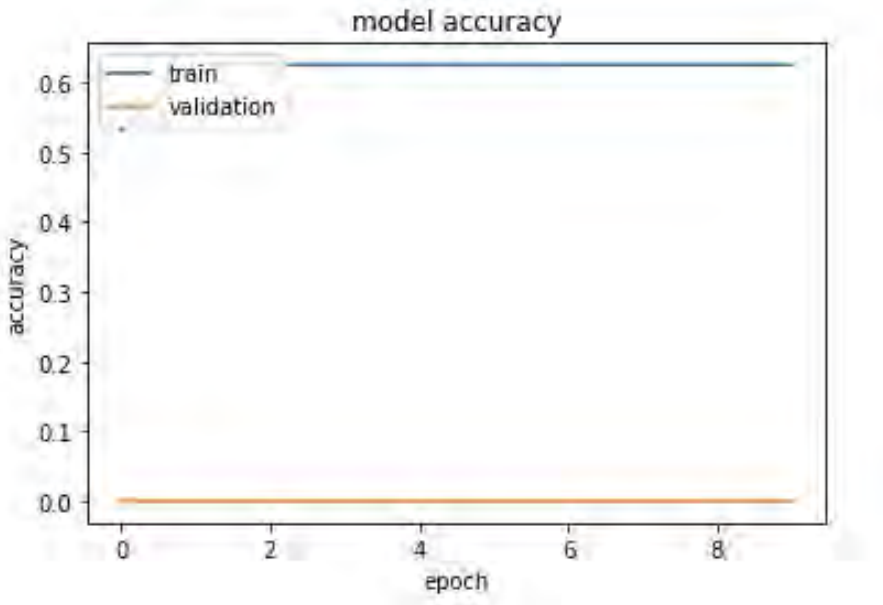

Figure 4.29: Model Accuracy of VGG16 and VGG19.

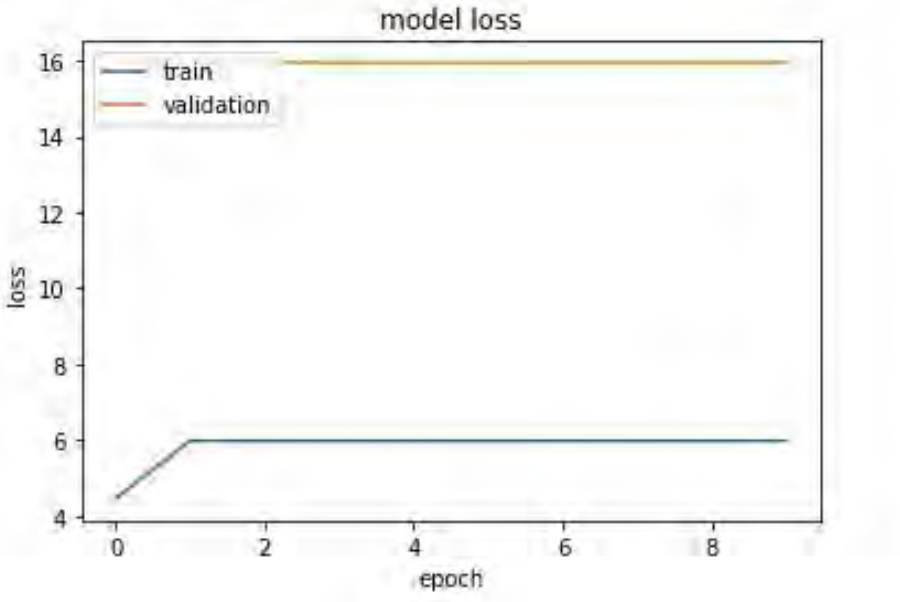

Figure 4.30: Model Loss of VGG16 and VGG19.

4.7 Comparison between different CNN Models

From the comparison table: 4.7 and Figure no: 4.31, 4.32 and 4.33, we can see that our self-designed model, InceptionV3 , Xception and MobileNetV2 have over all performed well on our data. On the other hand VGG16 and VGG19 have performed very poor on our model. With the same accuracy and slightly different result in precision, recall, f1 score and area under ROC curve our self-designed model and inceptionV3 both have shown best performance on our data.

Table 4.7: Comparison Table.

Model Name	Precision		ReCall		f1		Area Under ROC Curve	Accuracy
	Demented	Non-Demented	Demented	Non-Demented	Demented	Non-Demented		
Self Designed	100%	84%	81%	100%	90%	91%	99.21%	90.62%
Inception V3	93%	88%	88%	94%	90%	91%	96.48%	90.62%
Xception	100%	76%	69%	100%	81%	86%	96.48%	84.37%
MobileNetV2	92%	75%	69%	94%	79%	83%	85.15%	81.24%
VGG16 & VGG19	0%	50%	0%	100%	0%	67%	50%	50%

As mentioned in table: 4.7 and figure no: 4.31, 4.32 and 4.33 we have shown and plotted accuracy, precision, recall, F1 score and area under ROC curve to discuss the performance of our different CNN architectures. As we have already seen the difference in performance between different CNN models now let us discuss why our self-designed model and inception V3 have the best performance than other CNN models and why overall our self-designed model, InceptionV3, Xception, MobileNetV2 have shown a good performance but VGG on the other hand has a very poor performance. In clinical and bio medical decision making using deep neural network size of Dataset is a crucial factor. If sample size is not that big it is heard for deep neural networks to give a better performance. In our dataset as we have mentioned earlier we have fifty six demented patients and fifty six non demented patients. If we consider the size of the dataset it is not very large. Now if we analyze

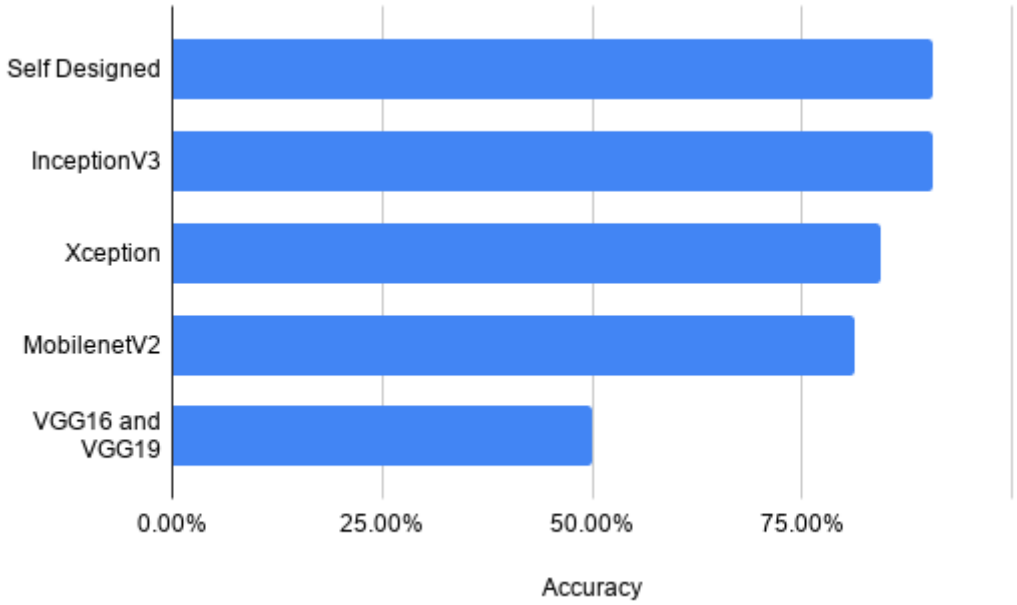

Figure 4.31: Accuracy VS Model Name.

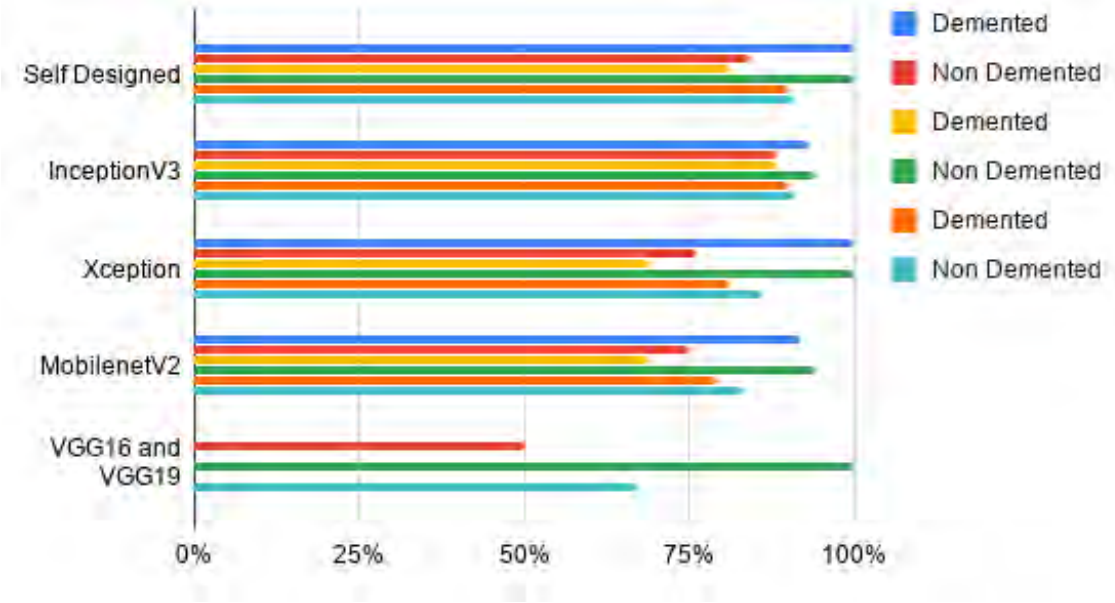

Figure 4.32: Precision,RaCall,f1 Score for Demented and Non-Demented.

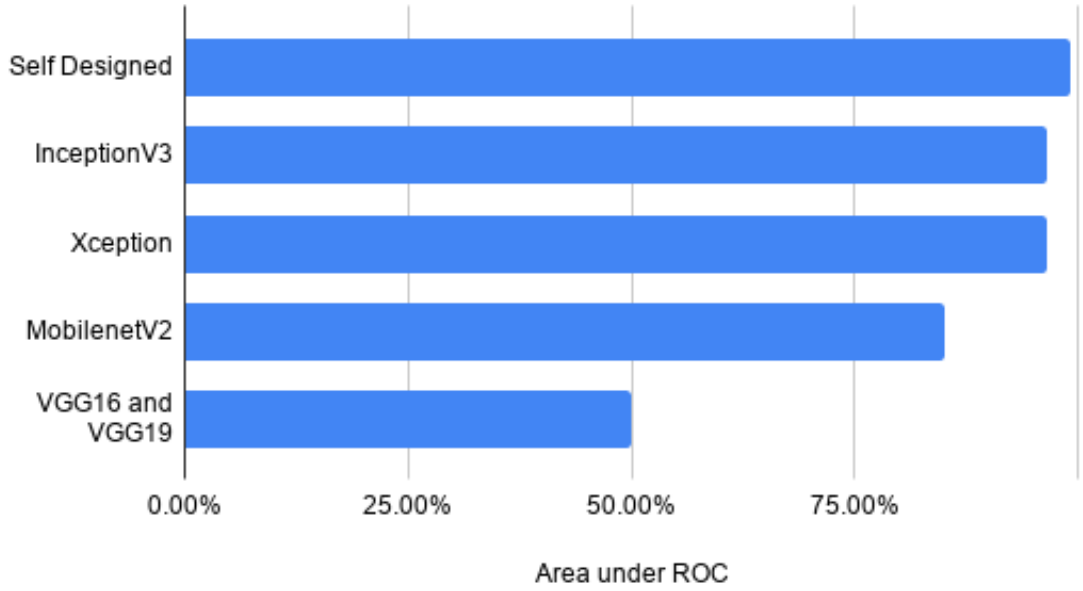

Figure 4.33: Area under ROC curve VS Model Name.

the CNN models we used we will see that in inception V3 number of total parameters are 24 M , in Xception number of total parameters are 23 M , MobileNetV2 total numbers of parameters are 3.4 M and In VGG16 total numbers of parameters are 138 M.

We have a better result with Inception than the other predefined defined models . The reason why we are getting a better result with InceptionV3 is among the other pre defined models of CNN InceptionV3 has a reputation for better performance accuracy which is always greater than 78.1% on image based datasets. InceptionV3 was 1st runner up in ILSVRC 2015 which was based on ImageNet dataset which is of high resolution images. Moreover, InceptionV3 has a lower error rate than other pre-defined models in image based datasets.

We can easily determine that VGG is a very deep network than the others. As we have a smaller sample size with a very deep network like VGG with lots of parameters it can be a reason we are not providing enough value for all those parameters . This is our fist guess why VGG is giving a poor performance. Also, in some other image based datasets like ImageNet, VGG has shown bigger error rate. Certainly there are other factors which will be more legit in this situation to explain the poor performance of VGG which we are still working and researching to find out.

Chapter 5

Conclusion

The main purpose of this research work was to apply different CNN models and find out which model performs best on our given data. After the whole work process was done we saw that model like InceptionV3 gives a very good performance on our data on the other hand VGG16 and VGG19 has a very poor performance. In order to complete this research work we have done vast study on machine learning, deep learning and different CNN architectures. We have come to know about a lot of things. In this research we have done binary classification. It is possible to work on this data using multi class classification with better computing set up.

Future Plan

In this research we have classified Alzheimer vs healthy patients using different CNN architectures. In this we have worked on binary classification which is demented vs non-demented. Our future plan is to detect early Alzheimer using different CNN architectures. Also we have a plan to make this as a multi class problem where we will classify mild, strong severe these types of different demented patients and non-demented patients separately. Also the models VGG16 and VGG19 have given less accuracy. We have a plan to work on these models with different parameters to determine if the accuracy and performance of this model can be increased.

Bibliography

[1] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies (oasis): Cross-sectional mri data in young, middle aged, nondemented, and demented older adults", *Journal of cognitive neuroscience*, vol. 19, no. 9, pp. 1498–1507, 2007.

[2] S. Klöppel, C. M. Stonnington, C. Chu, B. Draganski, R. I. Scahill, J. D. Rohrer, N. C. Fox, C. R. Jack Jr, J. Ashburner, and R. S. Frackowiak, "Automatic classification of mr scans in alzheimer's disease", *Brain*, vol. 131, no. 3, pp. 681–689, 2008.

[3] P. Vemuri, J. L. Gunter, M. L. Senjem, J. L. Whitwell, K. Kantarci, D. S. Knopman, B. F. Boeve, R. C. Petersen, and C. R. Jack Jr, "Alzheimer's disease diagnosis in individual subjects using structural mr images: Validation studies", *Neuroimage*, vol. 39, no. 3, pp. 1186–1197, 2008.

[4] C. Plant, S. J. Teipel, A. Oswald, C. Böhm, T. Meindl, J. Mourao-Miranda, A. W. Bokde, H. Hampel, and M. Ewers, "Automated detection of brain atrophy patterns based on mri for the prediction of alzheimer's disease", *Neuroimage*, vol. 50, no. 1, pp. 162–174, 2010.

[5] K. R. Gray, "Machine learning for image-based classification of alzheimer's disease", PhD thesis, Imperial College London, 2012.

[6] A. Gupta, M. Ayhan, and A. Maida, "Natural image bases to represent neuroimaging data", in *International conference on machine learning*, 2013, pp. 987–994.

[7] F. Liu and C. Shen, "Learning deep convolutional features for mri based alzheimer's disease classification", *arXiv preprint arXiv:1404.3366*, 2014.

[8] E. Arvesen, "Automatic classification of alzheimer's disease from structural mri", Master's thesis, 2015.

[9] T. Liu, S. Fang, Y. Zhao, P. Wang, and J. Zhang, "Implementation of training convolutional neural networks", *arXiv preprint arXiv:1506.01195*, 2015.

[10] A. Payan and G. Montana, "Predicting alzheimer's disease: A neuroimaging study with 3d convolutional neural networks", *arXiv preprint arXiv:1502.02506*, 2015.

[11] E. Hosseini-Asl, R. Keynton, and A. El-Baz, "Alzheimer's disease diagnostics by adaptation of 3d convolutional network", in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2016, pp. 126–130.

[12] S. Sarraf and G. Tofighi, "Classification of alzheimer's disease using fmri data and deep learning convolutional neural networks", *arXiv preprint arXiv:1603.08631*, 2016.

[13] D. Cheng and M. Liu, "Combining convolutional and recurrent neural networks for alzheimer's disease diagnosis using pet images", in *2017 IEEE International Conference on Imaging Systems and Techniques (IST)*, IEEE, 2017, pp. 1–5.

[14] *Introduction to convolution neural network*, Aug. 2017. [Online]. Available: https://www.geeksforgeeks.org/introduction-convolution-neural-network/.

[15] J. Islam and Y. Zhang, "An ensemble of deep convolutional neural networks for alzheimer's disease detection and classification", *arXiv preprint arXiv:1712.01675*, 2017.

[16] A. Rosebrock, *Imagenet: Vggnet, resnet, inception, and xception with keras*, Mar. 2017. [Online]. Available: https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/.

[17] S. SHARMA, *Activation functions in neural networks*, Sep. 2017. [Online]. Available: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[18] M. Hollemans, *Mobilenet version 2*, Apr. 2018. [Online]. Available: https://machinethink.net/blog/mobilenet-v2/.

[19] V. Nigam, *Understanding neural networks. from neuron to rnn, cnn, and deep learning*, Sep. 2018. [Online]. Available: https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90.

[20] Ozgen and B. Ozgen, *Dark side of neural networks explained*, Dec. 2018. [Online]. Available: https://blog.aimultiple.com/how-neural-networks-work/.

[21] Prabhu, *Understanding of convolutional neural network (cnn) - deep learning*, Mar. 2018. [Online]. Available: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148.

[22] B. Raj, *A simple guide to the versions of the inception network*, May 2018. [Online]. Available: https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202.

[23] S. Saha, *A comprehensive guide to convolutional neural networks-the eli5 way*, Dec. 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[24] M. Sandler and A. Howard, *Mobilenetv2: The next generation of on-device computer vision networks*, Apr. 2018. [Online]. Available: https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html.

[25] S.-H. Tsang, *Review: Googlenet (inception v1)- winner of ilsvrc 2014 (image classification)*, Aug. 2018. [Online]. Available: https://medium.com/coinmonks/paper-review-of-googlenet-inception-v1-winner-of-ilsvlc-2014-image-classification-c2b3565a64e7.

[26] ——, *Review: Inception-v3 - 1st runner up (image classification) in ilsvrc 2015*, Sep. 2018. [Online]. Available: https://medium.com/@sh.tsang/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c.

[27] ——, *Review: Mobilenetv1-depthwise separable convolution (light weight model)*, Oct. 2018. [Online]. Available: https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69.

[28] ——, *Review: Xception - with depthwise separable convolution, better than inception-v3 (image...* Sep. 2018. [Online]. Available: https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568.

[29] Y. Uchida, *Why mobilenet and its variants (e.g. shufflenet) are fast*, Apr. 2018. [Online]. Available: https://medium.com/@yu4u/why-mobilenet-and-its-variants-e-g-shufflenet-are-fast-1c7048b9618d.

[30] *Vgg16 - convolutional network for classification and detection*, Nov. 2018. [Online]. Available: https://neurohive.io/en/popular-networks/vgg16/.

[31] Chris Nicholson, *Evaluation metrics for machine learning - accuracy, precision, recall, and f1 defined*, 2019. [Online]. Available: https://pathmind.com/wiki/accuracy-precision-recall-f1.

[32] S.-H. Tsang, *Review: Mobilenetv2-light weight model (image classification)*, May 2019. [Online]. Available: https://towardsdatascience.com/review-mobilenetv2-light-weight-model-image-classification-8febb490e61c.

[33] J. Wei, *Vgg neural networks: The next step after alexnet*, Jul. 2019. [Online]. Available: https://towardsdatascience.com/vgg-neural-networks-the-next-step-after-alexnet-3f91fa9ffe2c.

[34] *Xception model and depthwise separable convolutions*, Mar. 2019. [Online]. Available: https://maelfabien.github.io/deeplearning/xception/#.

[35] V. Zhou, *Cnns, part 1: An introduction to convolutional neural networks*, Aug. 2019. [Online]. Available: https://victorzhou.com/blog/intro-to-cnns-part-1/.

[36] [Online]. Available: http://cs231n.github.io/convolutional-networks/.

[37] *A beginner's guide to neural networks and deep learning.* [Online]. Available: https://skymind.ai/wiki/neural-network.

[38] E. Amulya, S. Varma, and V. Paul, "Classification of brain mr images using texture feature extraction",

[39] *Oasis-1: Cross-sectional mri data in young, middle aged, nondemented and demented older adults.* [Online]. Available: https://www.oasis-brains.org/#data.

[40] T. Tape. [Online]. Available: http://gim.unmc.edu/dxtests/ROC3.htm?fbclid=IwAR2yjqfD78XS2SsvkcGn6LPYE7cuAtsAkxFxZbMKJCV43L1tlTYEfdXb8dM.