A Deep Learning Approach to Depression Detection Based on Convolutional Neural Networks and Transfer Learning

by

Kaniz Fatima Sarmi
17102040
Shaikh Mahmudur Rahman
17101338
Nusrat Jahan Sultana
17101331
Khandaker MD. Asef Anzoom Shanto
17101248

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
October 2021

Declaration

It is hereby declared that

1. We are the only authors of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other university or any Institution. This paper is fully our own work while completing a 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:

Kaniz Fatima Sarmi
17201040

Khandaker MD. Asef Anzoom Shanto
17101248

Shaikh Mahmudur Rahman
17101338

Nusrat Jahan Sultana
17101331

Approval

The thesis titled "Comparative Analysis for Depression Detection Using Deep Learning" submitted by

1. Kaniz Fatima Sarmi (17201040)
2. Shaikh Mahmudur Rahman (17101338)
3. Nusrat Jahan Sultana (17101331)
4. Khandaker MD. Asef Anzoom Shanto (17101248)

of Summer 2021 has been accepted as satisfactory in partial fulfilment of the requirement for the degree of B.Sc. in Computer Science and Engineering on October 02, 2021.

Examining Committee:

Supervisor:
(Member)

Zavid Parvez
Digitally signed by Zavid Parvez
DN: cn=Zavid Parvez, o=Brac University, ou=CSE, email=zavid.parvez@bracu.ac.bd, c=US
Date: 2020.10.05 18:46:40 +06'00'

Dr. Mohammad Zavid Parvez
Assistant Professor
Department of Computer Science and Engineering
Brac University

Co- Supervisor:
(Member)

Saiful

Md. Saiful Islam
Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

____

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

Head of Department:
(Chair)

____

Sadia Hamid Kazi
Associate Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Ethics Statement

Our primary data were collected from a renowned databank website. In addition, we the members, hereby and sincerely declare that this thesis has been done based on the findings of our extensive research. All the materials, which have been used are properly noted and cited in this report. This research work, neither in full nor in any part, has never been submitted by any other person to another university or any institution for the award of any degree or any other purpose.

Abstract

Depression and mental health issues (stress, nervousness, panic attacks, anxiety attacks etc.) are nowadays a major issue in the whole world. It is a common cause of mental illness that has been linked to an increased risk of dying young. Especially in our country, mental health is an issue which most of the families do not want to give as much attention as it is supposed to get and because of that so many people who are suffering from Major Depressive Disorder (MDD) are often helpless. Currently there are numerous ways to detect depression by various methods. For example: emotion recognition, social media records, analyzing daily routine with the help of machine learning and many more. This paper aims to detect depression by implementing various deep learning/ transfer learning models (for example: VGG16, Xception, ResNet152, MobileNetV2 etc.) using EEG brain signals to discover the model that provides the highest level of accuracy for our data type. In addition, we want to analyze why the particular model performs better and what might be the cases to make a model perform better to propose a model so that this method of modeling can be used in most cases for detecting depression and model improvement. Furthermore, we have made a custom model which gives the most accuracy (99.75%). We are successful at bringing the highest accuracy among the existing models which were implemented by us. For this reason, we are analyzing the EEG brain signal data of several healthy and MDD patients. We believe that this research will aid in the development of innovative strategies for building models and early identification of depression in our daily lives.

Keywords: Electroencephalogram (EEG), Major Depressive Disorder (MDD), Brain Signal Analysis, Transfer Learning Models, VGG16, ResNet152, Xception, MobileNetV2, Machine Learning

Dedication

This thesis is dedicated to our loving parents and our department's respectable faculties, who have encouraged and supported us during the entire thesis and motivated us to achieve excellence in every aspect.

Acknowledgement

Firstly, we are grateful to the Almighty Allah for whom we have been able to finish our thesis without any major interruptions during this pandemic situation.

Secondly, we wish to express our sincere thanks to our supervisor Md Zavid Parvez sir and co supervisor Md. Saiful Islam sir of the Department of Computer Science and Engineering at BRAC University for their kind support and advice in our work. Their valuable suggestions, ideas, support and effort have significantly helped us in our research work. They helped us whenever we needed help as well as assisted us with our work in researching.

Thirdly, we want to take a moment to thank our supportive friends who have been there to give us mental support in the hard times.

Finally last but not the least, we would like to express our gratitude towards our parents who made sure we read in one of the best institutions and without their continued prayers, encouragement, and support without which it would not have been possible for us to reach this point.

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

ANN Artificial Neural Network

CNN Convolutional Neural Network

ConvNet Convolutional Neural Network

DenseNet Densely Connected Convolutional Networks

DFT Discrete Fourier Transform

DT Decision Tree

EEG Electroencephalogram

FFT Fast Fourier transform

MDD Major Depressive Disorder

MLP Multilayer Perceptrons

ReLU Rectified Linear Unit

ResNet Residual Neural Network

RNN Recurrent Neural Network

SVM Support Vector Machine

VGG Visual Geometry Group

Chapter 1

Introduction

A continuous feeling of sadness and loss of interest creating mood disorder is known as depression. Various physical and emotional problems are caused due to this major depressive disorder [1]. Anxiety is a feeling created by tension, unusual and nervous thoughts and which leads to various physical changes. This problem is very severe at present in the whole world. If we take a look at the situation of our country then 95% people are suffering from mental illness (Depression Anxiety) at present based on the survey done by NIMH 2018-19 [2]. Generally, depression anxiety affects every activity of life. It is not a minor issue that heals within a very short time rather it is a very lengthy process. A study shows that the world's second leading cause of disability is depression. Sadly, 10% of individuals obtained effective remedies [3]. Sometimes the treatments become ineffective for the patients suffering from major depression. Anxiety and panic attacks that last for a long time can lead your brain to release stress chemicals on a regular basis. Symptoms such as headaches and dizziness may become more frequent as a result. A 2017 study estimates that 792 million people lived with a mental health disorder. This is slightly more than one in ten people globally (10.7%)[4].

Depression is a critical public health concern because of its prevalence, as well as the pain, deterioration, complication, and productive cost it causes. Although depression and its adverse effects are recognized widely, there are many drawbacks related to this topic. Currently many deep learning methods are available which can detect depression depending on different types of inputs like EEG graphs, personal history, social media history etc. However, different types of data require different types of models. In the case of image data classification, like the dataset used on this paper, there are also many ways to detect depression. Nevertheless, there is no general guideline or process that can help us to create a custom model or a modified model from an existing model. An idea like this which might be able to create a custom model of a modified model will hopefully come handy to many future researchers who are going to work on this topic.

1.1 Background

Depression is a very frequent mental illness. The indications of depression are basically a persistent sense of sadness or melancholy, a sense of hopelessness and helplessness, having a low sense of self-worth and many others. Moreover, it can even occur for no specific reason. These events can cause obstruction in a person's everyday life.

A major portion of global disease load is occupied by depression. The aftermath of depression is very tragic as it can cause a person to dysfunction in their living. The most saddening part is the effects of depression remain for a long duration of time.

Various treatments are available as a cure for moderate depression. However, these facilities are not available in underdeveloped and developing countries. It is really tough to get proper treatment of depression in these countries. Therefore, it is very necessary to detect depression at the shortest possible time. Otherwise, it becomes very difficult for the patient to start the necessary treatments and cope up with them. Until now, various researches have been done related to this but in this paper, we are going to do an important study based on the works done earlier in this topic.

Firstly, we have preprocessed the required dataset and trained them. Later on, by implementing numerous CNN and Deep Learning techniques we have found the optimum performance of each of the models. We are going to do a study based on the existing models to see what is the actual reason that is giving the respective accuracies. This will not only help the people to work easily with this topic in future and understand the relative studies of the models to detect depression but also will give an effective result in treatment of the depressed patient.

However, we have faced many difficulties while working on this topic. Data preprocessing was a major issue. Since there was a lack of dataset, we would not get the optimum result without training them. Fine Tuning was not that much of an issue but while comparing the layers in the models we faced a bit of difficulty. Finally, our main goal was accomplished.

Since we are showing a detailed analysis, we are hoping that this will be helpful for further research works.

1.2 Problem Statement

Nowadays depression is a very common disorder, which is affecting almost 264 million around the world [5]. Depression and anxiety can lead to severe health condition and often to suicide. We decided to work on this topic for various reasons. Firstly, mental illnesses are a serious public health issue that account for 13% of the worldwide burden of disease as assessed by disability adjusted life years [6]. Since the number of depressed people is increasing day-by-day and so we wanted to bring some effective measures in the present system of detecting. Recently due to social media, people are becoming more depressed. We tend to share our happy moments in social media but the people who are deprived of that happiness become sad and depressed by seeing that. Team Facebook themselves admit that using it will increase the risk problems of mental health more [7]. We know that almost all teenagers in our country use Facebook or many other social media. So, the chances of being depressed are increasing day by day. Most people are unaware about mental health issues. In our country, 92.3% of the people suffering from mental disorders do not seek treatment [8]. So, it is important to find the best effective model to detect depression so that the patient can undergo treatment as early as possible.

According to a report of PMC, only 16% of patients came straight to seek treatment for mental health difficulties, according to the findings [9]. According to a community-based rural survey, 3.6 percent of people suffer from psychiatric diseases, and 2.9 percent suffer from both psychiatric and physical disorders, with depression and anxiety being the most common [10]. Between 2003 and 2005, the first national study on mental health found that 16.1% of elders suffered from severe mental problems such as depression, anxiety, with females accounting for a higher percentage than males [11]. According to a recent study based on data from the national population health survey, young adults (aged 12 to 24) have the highest rates of first onset depression (1.4 percent - 9.1 percent of the population), while persons 65 and older had the lowest rates (1.3 percent -1.8 percent) [12]. Since our country lacks in giving chances to effective treatments for depression and anxiety especially to the people of rural areas so we are facing the necessity to enlighten this fact. We have used CNN (convolutional neural network) and some transfer learning models in our work for analyzing the required datasets.

Nevertheless, the point is, there are many available deep learning models available, each with respective layers and parameters. Further, the accuracy of each model is different. Therefore, there is no effective way to determine which model to choose in case of some research. It will take a lot of time to figure out the perfect model for them to work with. It is not easy to study all of the available models to make a decision of which model they want to work with.

Thus, this study is attempting to address an idea. We will run some models with our dataset to compare which model has the highest accuracy of detecting depression. Then we will analyze the model layer by layer to find out what might be the reason why this model is working better than the others might. Then we will toggle the layers to see the effects of each change or which combination is giving the most effective performance and will try to propose a model of our own. Therefore, when someone needs to make some changes to an existing model or trying to make a new model what approaches might benefit them the most while doing that. We hope that this will increase the effectiveness of this type of research and create more awareness in this global issue.

1.3 Motivation

Depression is a significant mental condition that affects hundreds of millions of people worldwide. At present more than 264 million people, suffer worldwide because of depression [13]. Depressed people have to struggle a lot to lead a normal life but eventually many of them cannot. Until now, there are various detection methods available for different levels of depression. However, depression treatment and support services are not so developed and lacking in the developing and undeveloped countries. 80% of people have a lack of access to cure [14]. So, it is a dire need at present to detect depression as early as possible. After considering all these facts, we have implemented a number of models that can detect depression and give some optimized results. In addition, we present a comparison study based on the models we have built so that we get to know which one is working better and for what reason. Finally, we will try to present a model. Based on our research early detection of depression will be easier.

1.4 Research Objective

In Bangladesh, people do not take mental health problems seriously and think of it as a hoax that is leading generations to an unhappy life. In the city, people nowadays are more aware of mental health than before but in rural areas, most of the people do not consider it as anything important. The significant contributions of this thesis are stated as follows:

- As mentioned earlier, there are many deep learning models available to perform depression detection. We have proposed a way and a model to create customized or modified models for better accuracy.
- Our hypothesis shows how adding or subtracting layers to our modified model changes the results. This will help the researchers in the future to determine better models and compare their results.
- Though there are many existing models, ours one will give the best results in this particular dataset.
- Since we have implemented a new model, comparative analysis will be better if someone wants to work based on our datasets.
- Optimized performance is shown by our model so in the majority of cases, modeling can be used to detect depression and enhance models.

We have organized the remaining part of our report in the following way. In Chapter 2 we have discussed the literature reviews and the existing works about depression detection. In Chapter 3 we described about the implemented models and then Chapter 4 shows the implementing of datasets. In Chapter 5 we have described about our workflow and Chapter 6 describes the challenges we have faced. In Chpater-7 we showed the comparative analysis of the models we implemented and our proposed customized model has been implemented. Chapter 7 gives the conclusion of our thesis.

Chapter 2

Litarature Review

Previously there has been many works done regarding depression analysis through EEG signals and CNN, SVM, ALM, DFT, FFT- these classification algorithms are widely used in most EEG studies.

Halim, Z. et al [15] collected their dataset from 86 automotive drivers mostly young adults. Their main mission was to classify driving-induced stress patterns using machine learning-based approach and EEG signals were used as the physiological signals. They applied these raw EEG signals to the Convolutional Neural Network (CNN) model and result they got fed to Long Short-Term Memory (LSTM) as inputs. They performed three methods Neural Network (NN), SVM and Random Forest (RF) to break down the patterns of EEG on the basis of the self-assessment of the emotional state of the tested person when the drove in various situations. Tested over 50 automotive drivers, SVM performed better along with 97.95% $\pm$ 2.65% accuracy as well as 88.83% sensitivity and 94.92% specificity.

Acharya, U. R. et al [16] used both linear as well as nonlinear methods for their project. In linear methods, different types of results came for the depression patients. They performed a feature ranking to define the importance of the identified features between normal and depression groups. The authors used the algorithms of SVM, K-Nearest Neighbor (KNN) algorithms and Decision Tree (DT) and compared between group of normal and depressed persons. They finally find accuracy of 88 and 94% in the linear and nonlinear methods, also claimed 99.5% accuracy in discrimination between normal and depression signals.

Bobde, S. P. et al [17] collected EEG data of 80 subjects and out of these, 76 subjects were classified. They collected their dataset from PhysioNet and PhysioBank where every dataset accommodate one minute EEG data. They used DFT, FFT on the occipito-parietal wave which was produced from occipital as well as parietal parts of their brain. In their project, 37.5% subjects showed the signs of MDD whereas 80% subjects showed a small amount of depression and last 20% of the subjects did not show statistical EEG parameters that would indicate depression. In addition, they could not classify 2.5% of the subjects because of some noises in EEG signal.

Juliana, M. J. M. et al [18] choose four classification algorithms-KNN, LDA (Linear Discriminant Analysis), QDA (Quadratic Discriminant Analysis) and D tree

(Complex Decision Tree) and applied on 24 sets of data from different databases. Then they applied FFT on EEG to identify the frequency of anxiety using MATLAB as the tool. Among the algorithms, LDA prediction percent provided was 55.4144% and area under curve showed 0.5184. KNN prediction percent provided was 55.1880% and area under curve showed 0.5420. QDA prediction percent provided was 47.0507% and area under curve showed 0.4963. D tree prediction percent provided was 53.4272% and area under curve showed 0.5042.

H. Cai, J. Han et al [19] used KNN, SVM, CT, and ANN (Artificial Neural networks)- these four algorithms on 250 participants where 92 depressed patients and 121 normal subjects total 213 finished successfully their experiment. All of these classifications and 10-fold cross-validation were performed using the MATLAB software version of R2014a. Among all selected algorithms, the performance of KNN was best and the accuracy rate was 79.27%. The outcomes likewise showed featuring "absolute power of theta wave" to be processed with a quicker data handling productivity.

W Mumtaz, SSA Ali et al [20] used some classification models such as SVM, logistic regression (LR), and naive Bayesian (NB) in their work. This model employed the relationship between EEG functions and study groups between MDD patients and healthy persons. The dataset was collected from 34 MDD patients with 18 women (age, mean = 40.33, standard deviation ± 12 861) and a group of 30 healthy persons with women = 9 of the same age (age, mean = 38 237, standard deviation ± 15, sixty-four). The results show that the odds are better than the chance. They found that the study results SVM performed accuracy of 98%, 99.9% sensitivity and 95% specificity. Whereas LR performed accuracy of 91.7%, 86.66% sensitivity and 96.6% specificity. Lastly NB performed accuracy of 93.6%, 100% sensitivity and specificity of 87.9. The authors finally concluded by saying that Synchronization likelihood (SL) features may be a promising diagnostic tool for depression.

Mousavian, M. et al [21] explored classification, augmentation and feature extraction to detect MDD in sMRI pictures. They extracted ten pieces with the best entropy value using visualization. By the use of machine learning SVM, CNN with two famous CNN architectures named VGG16 and Inceptionv3; the authors classified MDD for HC using sMRI pictures. The data was taken from the NKI-Enhanced that contained totally 289 objects where they included 3D sMRI scans of around of 54 subjects. They saw that the raw functions, together with replication and SVMRBF, successfully classified the sMRI image with 0.89 AUC and 0.96 accuracy. The outcome showed that the SVMRBF and SVMLINEAR was thoughtful to unbalanced data as well as SVMRBF was more accurate but take longer to learn than SVMLINEAR.

Uyulan, Caglar et al [22] constructed an electroencephalogram based MDD diagnostic model using an advanced computational neurobiology approach combined using a deep convolutional neural model. The electroencephalogram was experimented by simulating three dissimilar deep CNN constructs like ResNet50, MobileNet and Inceptionv3 to separate MDD subjects from normal persons. EEG dataset used in the study were collected from 46 control persons and 46 depressed patients, approved by a local medical research committee. The result the authors found that

MobileNet architecture provides classification accuracy of 89.33% and 92.66%. In terms of bandwidth, delta band outperforms with a prediction accuracy of 90.22% and area under curve showed 0.9 for RestNet50 architecture.

Rodrigues Makiuchi, M. et al [23] offered a multimodal fusion of speech and language presentation to detect depression in their work. The authors trained their model to derive subject scores on the Patient Health Questionnaire (PHQ) and used deep spectral characteristics extracted from a previously trained VGG16 network as well as Gated Convolutional Neural Network (GCNN) accompanied by LSTM for speech method. They obtained CCC scores of 0.497 and 0.608 in EDAIC. In addition, they combined the two methods using feature fusion, applying the final representation of each model from a separate method to a fully connected layer to evaluate. By the use of multimodal approach, The CCC value of the development set is 0.696, and the CCC value of the EDAIC corpus is 0.403, which shows an absolute improvement of 0.283 points compared with the initial value.

D Ahmad, R Goecke et al [24] investigated the effectiveness of various CNN models when viewing only face and upper body images by assessing the severity of depression on a regressive scale. To analyze the result, two different datasets the Black Dog Institute and the 2013 Audio/Visual Emotion Challenge (AVEC) were used in here. The first dataset was obtained from Black Dog Institute of Sydney, Australia that was a non-profit organization dedicated to diagnosing, treating and preventing depression as well as various mental illnesses. Initially, 60 subjects were selected for this study from 80 participants to balance gender and age, 30 healthy people from the control group and 30 clinical diagnoses. Audio Visual Challenge 2013 submitted the second dataset from participants from Germany only. Here, it contained 150 videos of participants with varying degrees of depression, initially divided into 50 training videos, 50 more videos for authentication and 50 videos for testing. The author found in the results that the model performance difference between face and torso is very small, because the model performance is very similar on different architectures, but changes with the import of different models.

Chapter 3

CNN & Deep Learning Models

EEG or Electroencephalography is a common tool to study brain function. It is used to detect problems in the brain. It generates time-series data. By comparing this output data with ideal EEG signals can detect diseases or problems. Recently neural networks have played an important role in technology development. Deep learning is one of its implementations. Deep learning has progressed significantly in recent years. In comparison to a traditional neural network, deep learning comprises a far more sophisticated layer. Convolutional neural network (CNN) is a type of deep learning method that has recently gained popularity in deep neural networks. It uses input images and teaches the meaning of different objects (learning weights and distortions), assigning pictures and being able to distinguish one from the other. It is a set of convolutional and pooling layers that allow you to delete and accumulate the most important aspects of an image to create feature maps [25]. CNN gathers and transforms data through a sequence of hidden layers, each of which contains a large number of neurons, each of which is fully connected to all neurons in the upper layer. Neurons in a single layer work invisibly and do not share any connections. The CNN structure uses the fact that the data contains images to make training smarter. It consists of three types of layers that are convolutional layer, cluster layer and fully connected layer. By using top-down and convolutional sampling techniques, CNN converts the first input layer by layer to use this specific technique to obtain course estimates for classification and repetition purposes [26].

In this research paper, we compared and analyzed depression detection with different transfer learning models. The basic CNN diagram is given in this figure:

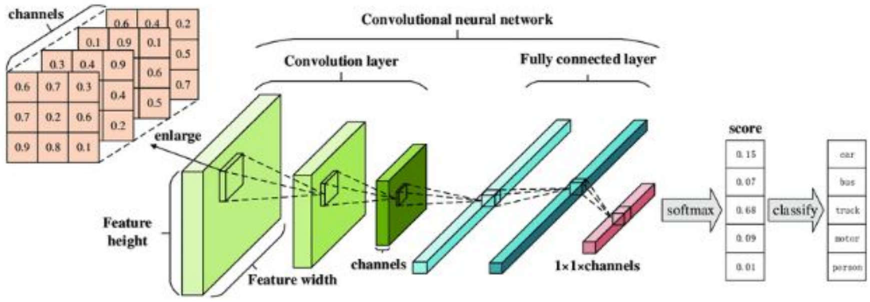

Figure 3.1: Basic CNN Architecture 01

3.1 Convolutional Layers

Convolutional layers measure the scalar product of their weights and the position of the input volume to determine the performance of neurons connected to adjacent input points. Activation occurs in last layer [27]. Convolution is a special type of simple operation used to extract when a set of numbers called parts are connected through a set of inputs called tensors. The size is small and the space is distributed throughout the depth of the entrance. When the data reaches the convolutional layer, they convolve each channel along the spatial dimension of the input to create a two-dimensional activation scheme [28]. Kernel convolution no longer applies only to CNN; it is an important part of many other computer vision calculations. This is a configuration; we use a small digital network, transmit it to the image and adjust it according to the channel value. Use m and n to check the row and column lists of subsequent networks, respectively which is shown in the bellow equation (3.1):

$$G[m,n] = (f * h)[m,n] = \sum_j \sum_k h[j,k] f[m-j, n-k] \tag{3.1}$$

After placing our channel on the selected pixel, we copied each bit estimate into its respective image value. Finally, we put everything together and placed the outcome, including the structure, in the proper location within the yield [29].

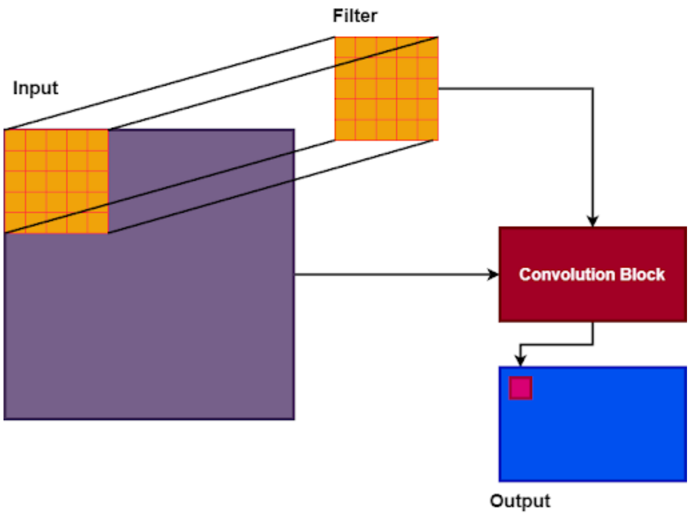

Figure 3.2: Basic Internal Architecture of CNN

3.1.1 Pooling layer

Pooling is a type of down sampling that eliminates the problem of boosting layers. The Pooling Layer performs numerous times on each profundity cut of the knowledge and resizes it spatially using the Maximum operation. The optimal configuration is a pooling layer of two 2x2 value-added channels with two down samples, where every depth, width, and stature are chopped in half, eliminating 75% of the decrees [30]. The depth measurement stays the same.

3.1.2 Fully connected layer

A feed forwarding neural network is used in the fully connected layer. These layers make up the network's final pair of layers. The convolutional neural arrangement is that type of arrangement where there are convolutional layers as well as pooling layers, which eliminates the fully connected top layer of the top line, which is equivalent to the neuron arrangement in the conventional neural configuration [31]. After extracting the highlights from the convolution layer and calculating the pooling layers, they are assigned to the final yield of the final fully bonded layer [32]. The number of yield nodes on the fully connected end layer is usually the same as the number of clusters. As shown in Figure 3.3 [33], there is a nonlinear work behind each fully connected layer, such as ReLU.

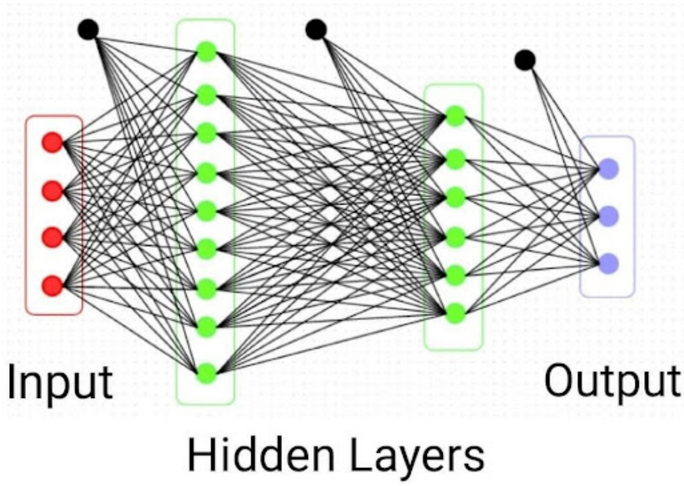

Figure 3.3: Fully Connected Layer

3.2 Activation Function

The activation function's major functions are to assess the hidden layer's asymmetric weight and then determine the neuron's activation. Determine how to convert the sum of input weights into layer neurons in the output. The network model's gradient and learning rate. The activation function can be broken down into three categories: binary, linear, and non-linear activation functions. We'll use a linear activation function in this case. Here are some of the first activations:

3.2.1 Sigmoid Function

When applying basic neural networks and logistic regression, this special function is mainly used [34]. However, the sigmoid function is avoided in more complex and advanced neural models. The equation of the sigmoid function is given in (3.2):

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

Here, it exists between [35] (0,1). Therefore, when the probability of predicting is between 0 and 1, the sigmoid chart is used. It has a differentiable S-curve. This is a monotonic function. The biggest disadvantage of this feature is data loss, which increases as the network grows deeper.

3.2.2 Hyperbolic Tangent (TanH)

The value range of the TanH function is from 1 to 1. A S-shaped sigmoid function is another name for this. The only difference is that the range is mapped so that negative points are on the negative axis and zero points are extremely near to zero.

$$y = 2\sigma(2x) - 1 \tag{3.3}$$

TanH is mainly used when assigning classes. This function has a central zero point[36], but there is another issue called Vanishing Gradient issue.

3.2.3 Rectified Linear Unit (ReLU)

ReLU is a piecewise linear function that outputs directly from an input. It has a range (0,). In the realm of deep learning, it is the most well-known activation function and can be summarized by the following equation (3.4):

$$y = max(0, x) \tag{3.4}$$

The core advantage of using ReLU [37] is that it fixes the vanishing gradient problem, compared with TanH, it has low activation (50%) and no backpropagation error. The disadvantage of ReLU is that this function has no zero center and cannot be differentiated to 0. Another drawback is the dying ReLU problem, which occurs when half of the ReLU's output is dormant and used for non-zero center actions.

3.2.4 Leaky ReLU

The leaky ReLU appeared in the realm of neural networks to overcome the dying ReLU problem. To get around the non-zero input problem, it defines a very small linear numerical element. The equation (3.5) is:

$$y = max(0.01 * x, x) \tag{3.5}$$

In case of any negative input, it returns the x value increased 0.01 in order that negative value conjointly can be achieved.

3.2.5 SoftMax

This function returns the probability distribution's output. As the entire summation becomes 1, it displays the input [38]. The equation (3.6) here is:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{k} e^{z_k}}; \quad for\ j = 1, 2, 3, ... \ k \tag{3.6}$$

It is used to multi-classify the logistic regression model used in the output layer of and it is used to classify the input into multiple groups of those networks.

3.2.6 Fine Tuning

Fine tuning means making minor changes to a process in order to get the desired result or performance. This method entails using a smaller dataset to train a previously trained network. It is a method that includes a multi-step process. In our case we not only had to upgrade the CNN architecture but also retrain it to learn new object classes in order to fine-tune it. The fully linked layers at the end of the neural network construction are referred to as classified layers, whereas the previously pre-trained layers have a lower learning rate. This is how we adapt features from the existing models to the new model and the dataset. We used it to make some changes in the existing models so that we can increase the accuracy.

3.3 Loss Function

Loss functions are significant in every statistical model because they establish an objective against which the model's performance is measured, and the parameters learned by the model are set by minimizing a given loss function [39] Loss functions outline what constitutes a good forecast and what does not. In other words, the loss function you choose determines how good your estimator will be. This article will look at loss functions, their importance in validating predictions, and the different types of loss functions that are utilized.

3.3.1 Cross Entropy Loss

The corresponding method for achieving a coding model that provides probability output of 0 to 1 is called cross-entropy loss [40]. The cross-entropy loss increases when the projected probability departs from the true value. In equation (3.7), the cross-entropy loss is described -

$$CE = -\sum_{i}^{c} t_{i\ log(s_i)} \tag{3.7}$$

Here, ti and si indicate fundamental truth as well as estimate considering i class in C in each CNN. The SoftMax loss, also known as the categorical cross entropy loss, is a SoftMax combination and cross entropy Loss shown in equation (3.8):

$$f(s)_i = \frac{e^{s_i}}{\sum_j^c + e^{s_i}} + CE = -\sum_{i}^{C} t_{i\ log(f(s_i))} \tag{3.8}$$

The sigmoid activation function and the lost entropy of the cross combine to form binary cross-entropy loss, also known as sigmoid loss which acts as sovereign for each element of the vector. In equation (3.9), we can see its mathematical depiction:

$$CE = -\sum_{i=1}^{C'=2} t_i log(f(s_i)) = -t_1 log(f(s_1)) - (1-t_1) log(1-f(s_1)) \tag{3.9}$$

The area of information theory known as cross-entropy [41] is concerned with getting entropy and quantifying the difference between probability distributions.

3.4 Optimization Algorithm

The optimization algorithm is utilized for both training as well as updating the cost function in the case of deep learning. This algorithm's general equation (3.10) is as follows:

$$J(W,b) = \frac{1}{m}\sum_{i=1}^{m} L(y'^i, y^i) \tag{3.10}$$

We employed Adam optimizer in our model, which is discussed below:

3.4.1 AdaM

Its full name is Adaptive Momentum. Because of its outstanding performance, AdaM is a well-known algorithm. Adam combines the RMSprop propulsion as a result, it's a very powerful as well as efficient algorithm. The following equations (3.11 and 3.12) define the optimizer:

$$\alpha_t = \alpha . \frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t} \tag{3.11}$$

$$\Theta_t \leftarrow \Theta_{t-1} - \alpha_t . m_{t/(\sqrt{v_t}+\hat{\epsilon})} \tag{3.12}$$

When we run into gradient descent problems, we use the AdaM optimizer to solve this problem as it gives us the advantage of keeping an exponentially decreasing average from previous similar gradients.

3.5 Existing Deep Learning Models

There are numerous deep learning models available on the market. Classic Neural Networks (Multilayer Perceptrons), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and other neural networks are only a few examples [42]. They are frequently utilized due to their capacity to learn high-level properties from data progressively. The need for domain expertise and the extraction of hard-core features is reduced as a result [43]. We have used Convolutional Neural Network (CNN) because parameter sharing and dimensionality reduction are two aspects of CNN. The number of parameters in CNN is lowered because of parameter sharing, and therefore the calculations are reduced [44].

3.5.1 CNN

Convolutional Neural Networks (CNN) is considered the most advanced architecture for image classification. When we talk about Convolutional Neural Networks (CNNs), we commonly refer to two-dimensional convolutional neural networks (CNNs) that are used to categorize images, but there are also one-dimensional and three-dimensional convolutional neural networks.[45].

It is a convolutional neural network that was first used in the Lenet5 architecture. Conv2D is primarily used with image data. Because the core moves across two dimensions of the data, this is referred to as 2D CNN. Here, the kernel moves in two directions. Inputs and outputs are 3 dimensional 2D CNN data [46].

The fundamental point of utilizing CNN is because the kernel can extract spatial information from your data, which other networks cannot do. It is particularly good at categorizing photos and other data with spatial properties.

3.6 Transfer Learning Models

Transfer learning is a technique for training a neural network model on a problem that is similar to the one at hand. The layers of the learned model are then used to train a new model on the problem of interest. Transfer learning cuts down on the time it takes to train a neural network model while also lowering generalization error. It is generally used to modify pre-trained models. There are numerous high-performing image recognition models accessible for download and use as the foundation for picture recognition and related computer vision applications which can be VGG16, VGG19, ResNet50, InceptionV3. Because of their effectiveness, these models are commonly utilized for transfer learning.

3.6.1 DenseNet169

Dense Convolutional Network, commonly known as DenseNet, is a feed-forward network that pre-connects each layer to all other layers. DenseNet has a number of undeniable benefits, including the elimination of the vanishing gradient problem, improved feature propagation, feature reuse, and a significant reduction in the number of parameters [47]. Densenet169 is one of DenseNet's group of image classification models. Densenet169 is approximately 57 MB in size. Densenet169 is a typical output of an object classifier for 1000 different classifiers that fit the ImageNet database [48]. Here is the figure the DenseNet 169 model architectures for ImageNet.

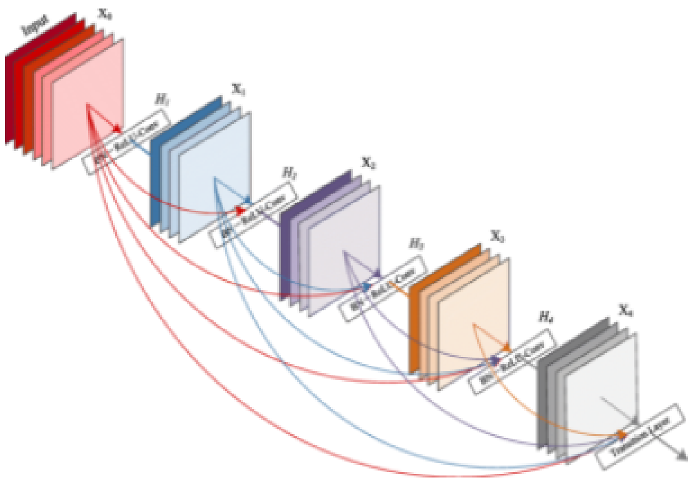

Figure 3.4: Layered Architecture of DenseNet169

Based on the ImageNet validation dataset, this model has a top 1 accuracy of 76.106

percent and a top 5 accuracy of 93.106 percent. It has 14,307,880 parameters and the topology depth of this network is 169 [49]

3.6.2 DenseNet201

DenseNet 201 is another image classification model from the DenseNet family. It is a 201-layer deep convolutional neural network [50]. The key difference between DenseNet 169 and this model is its size. The DenseNet 201 is larger, at over 80MB, than the DenseNet 169, which is around 57MB [51]. The typical object classifier output for 1000 separate classifiers that correspond to the ImageNet database is the model output for DenseNet 201.

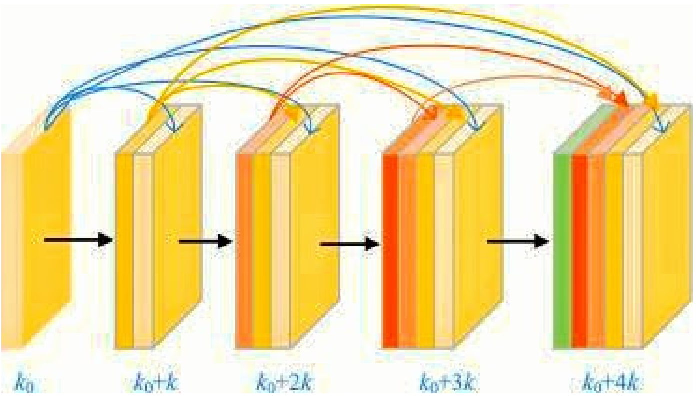

Figure 3.5: Layered Architecture of DenseNet201

Because of the ImageNet validation dataset, this model achieves top 1 accuracy of 76.886% and top 5 accuracy of 93.556%. It has 20,242,944 parameters and the topology depth of this network is 201 [52].

3.6.3 InceptionV3

Because the inception is easily configurable, adjustments in the number of channels at various levels are possible without affecting the quality of the completely constructed network [53] the main sprinter for the ILSVRC 2015 is Initiation v3 [54]. There are 42 layers in Inception v3. It is noteworthy that each channel is verified. It has a size of 92 MB, 23,851,784 parameters, and 159 depth, and ImageNet validation dataset shows that it has 77.9% accuracy in the top 1 and 93.7% in the top 5 [55]. Its architecture is shown in Figure 3.7.

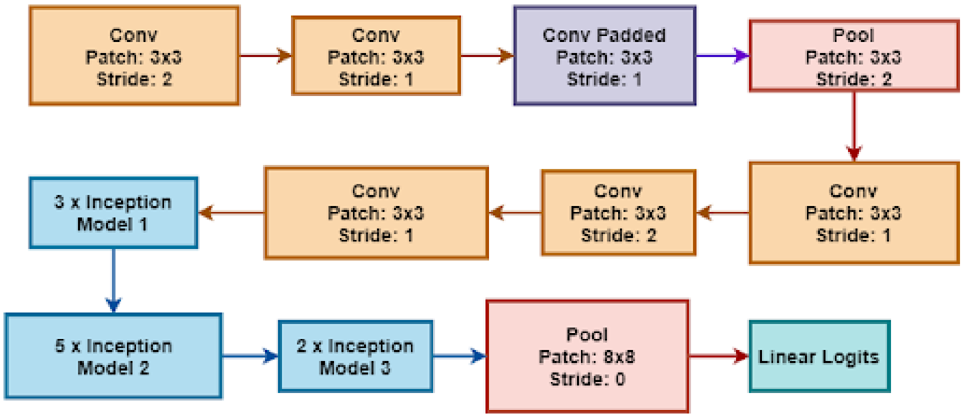

Figure 3.6: InceptionV3 Architecture

3.6.4 InceptionResNetV2

The Inception-Resnet v2 model is a convolutional neural network that was trained on over a million images from the ImageNet database. It has 164 layers and can categorize photos into 1000 different items. So, this network has learned multifunctional views for many types of pictures. The input picture size of this model is 299x299 [56]. Figure 3.8 shows the basic networking architecture of Inception-Resnet v2.

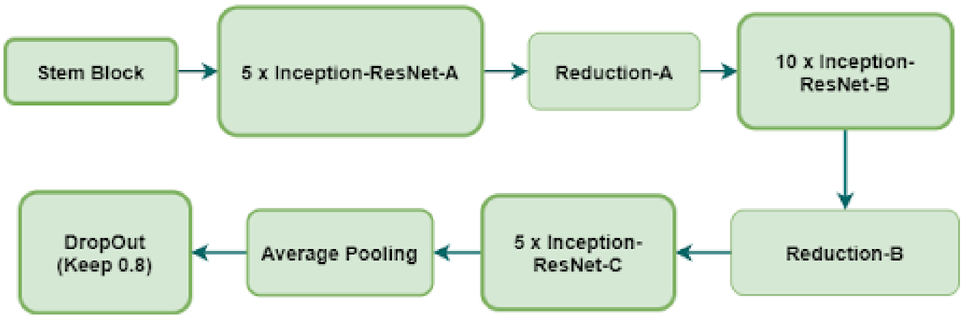

Figure 3.7: Basic architecture of InceptionResnetV2.

On basis of the ImageNet validation dataset, this network has a top 1 accuracy of 80.3% and a top 5 accuracy of 95.3%. It has 55,873,736 parameters and the topology depth of this network is 572 [57]

3.6.5 MobileNetV2

MobileNetV2 is a convolutional neural network architecture designed for mobile devices. It is built on an inverse residual structure, in which residual linkages between layers of bottlenecks are discovered. Collectively, MobileNetV2 architecture contains a total of 32 convolutional layers with 32 filters, followed by the remaining 19 bottleneck layers [58]. It is 53 layers deep and has learned multifunctional representations for many types of images with an input size of 224-by-224 [59]. Architecture of MobileNetV2 is given in the figure 3.9 below:

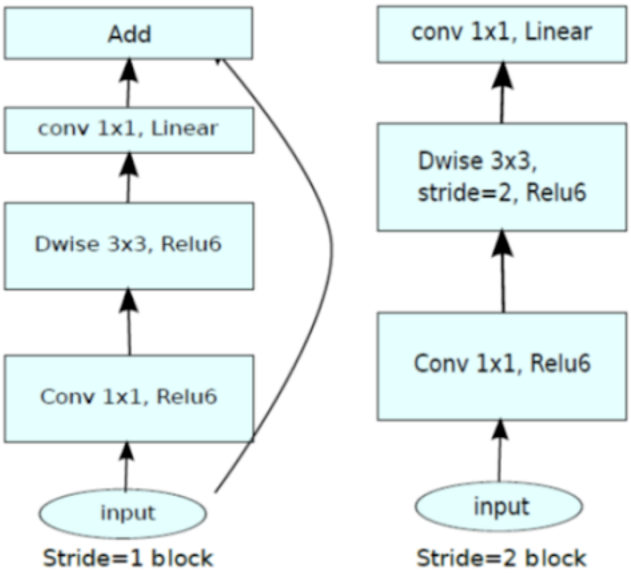

Figure 3.8: MobileNetV2 Overall Architecture

On the basis of the ImageNet validation dataset, this architecture achieves top 1 accuracy of 71.3 percent and top 5 accuracy of 90.1 percent and typically, the primary network has a computational cost of 3,538,984 parameters and 88 topological depths. [57] On the basis of the ImageNet validation dataset, this architecture achieves top 1 accuracy of 71.3 percent and top 5 accuracy of 90.1 percent and typically, the primary network has a computational cost of 3,538,984 parameters and 88 topological depths. [60]

3.6.6 ResNet152

Residual networks (ResNET) are a group of deep neural networks that have similar topologies but differ in depth [59]. The key benefit of this device is that it improves classification accuracy without adding to the model's complexity. By evaluating the residual representation functions rather than the signal representation directly, ResNet may create an extremely deep network with up to 152 layers. ImageNet is a collection of more than 15 million high-resolution images [61]. Here is the basic architecture of Resnet152 (in figure 3.10).

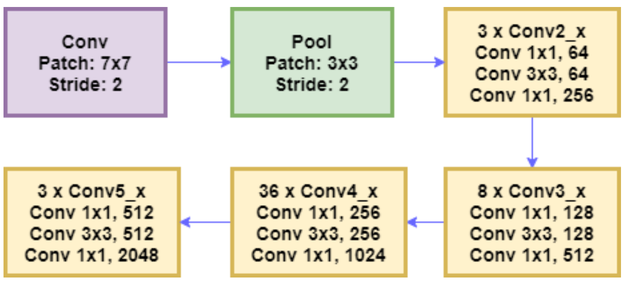

Figure 3.9: The Basic Architecture of ResNet152

Based on the ImageNet validation dataset, this model has a top 1 accuracy of 76.6% and a top 5 accuracy of 93.1%. It has 60,419,944 parameters and the size is 232 MB. [62]

3.6.7 VGG16

VGG16 is a convolutional neural network mastermind. This obtains a top-5 check precision of 92.7 percent in ImageNet, which is a dataset with over fourteen million images and a thousand classes [63]. It refines AlexNet by exchanging a significant number of calculated channels (11 and 5) within the top two convolutional layers, respectively, with different 33bit estimated channels in a very steady progression.

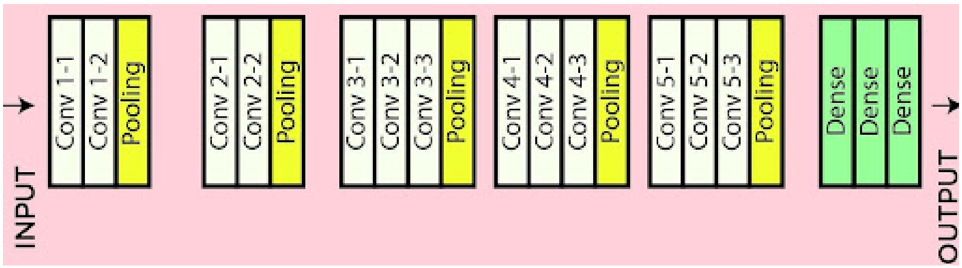

Figure 3.10: VGG16 Basic Architecture

The figure 3.11 shows that the cov1 layer is of settled gauge 224X224 RGB image [64]. It occupies 11 convolution channels, five max-pooling layers, three totally connected layers and takes after a variety of convolutional layers This network performs 71.3% accuracy in top 1, 90.1% accuracy in top 5 on the basis of the ImageNet validation dataset. It has 138,357,544 parameters and the topology depth of this network is 23 [65]

3.6.8 VGG19

VGG19 is a VGG variation of 19 layers containing 16 convolutional layers and 3 fully connected layers with 5 MaxPool layers and 1 SoftMax layer and 16 convolutional layers [66]. Compared to VGG16, VGG19 is slightly better, but it requires more memory. There are 16 layers in total with 5 blocks and each block has a maximum pooling layer [67].

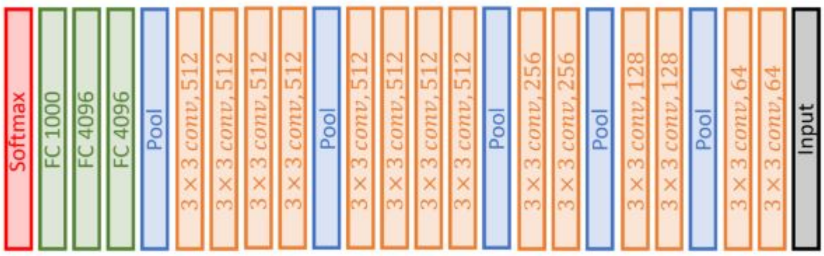

Figure 3.11: Basic VGG19 Architecture

Here in figure 3.12, first two convolutional layers are 3x3 filters where first two layers use 64 filters with a 224×224×64 RGB volume. The pooling layer goes from 224x224x64 down to 112×112×64 and dimension is 56x56x128. Finally, a 7x7x512 fully connected FC layer of 4096 units with softmax output of 1000 classes [68]. This network performs 71.3% accuracy in top 1, 90% accuracy in top 5 on the basis of the ImageNet validation dataset. It has 143,667,240 parameters and the topology depth of this network is 26 [69]

3.6.9 Xception

Xception is a deep convolutional neural network developed by Google researchers. Launchers are an intermediate step between conventional convolution and depth-division convolution in a convolutional neural network, according to Google. They proposed a deep convolutional neural network architecture inspired by Inception as the result of this observation [70]. It is a 71-layer deep convolutional neural network. The network has an input image size of 299 x 299 [71]. Overall architecture of Xception is given in figure 3.13 below:

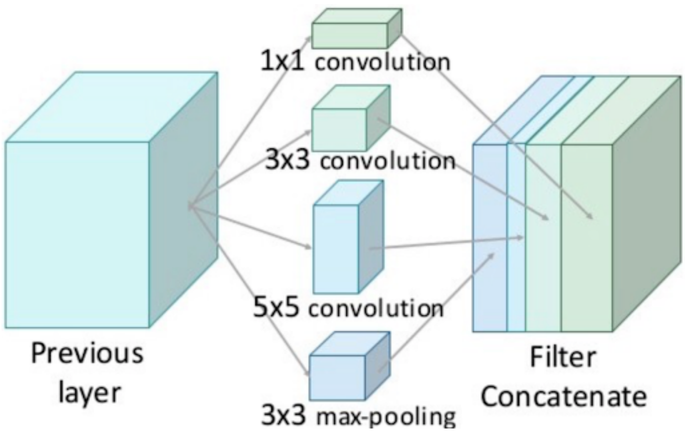

Figure 3.12: Overall Architecture of Xception

Here, we can see that SephableConvs is considered an Inception Module and is hosted throughout the deep learning architecture [72]. It has the size of 88 MB. This architecture performs in top 1 accuracy is 79%, top 5 accuracy is 94.5% on basis of the ImageNet validation dataset and typically, the primary network has a computational cost of 22,910,480 parameters and 126 topological depths [73]

Chapter 4

Research Methodology

Every scientific researcher has to go through a proper method to finish the task in time and properly. This includes from the very beginning like collecting data to the very end of the result.

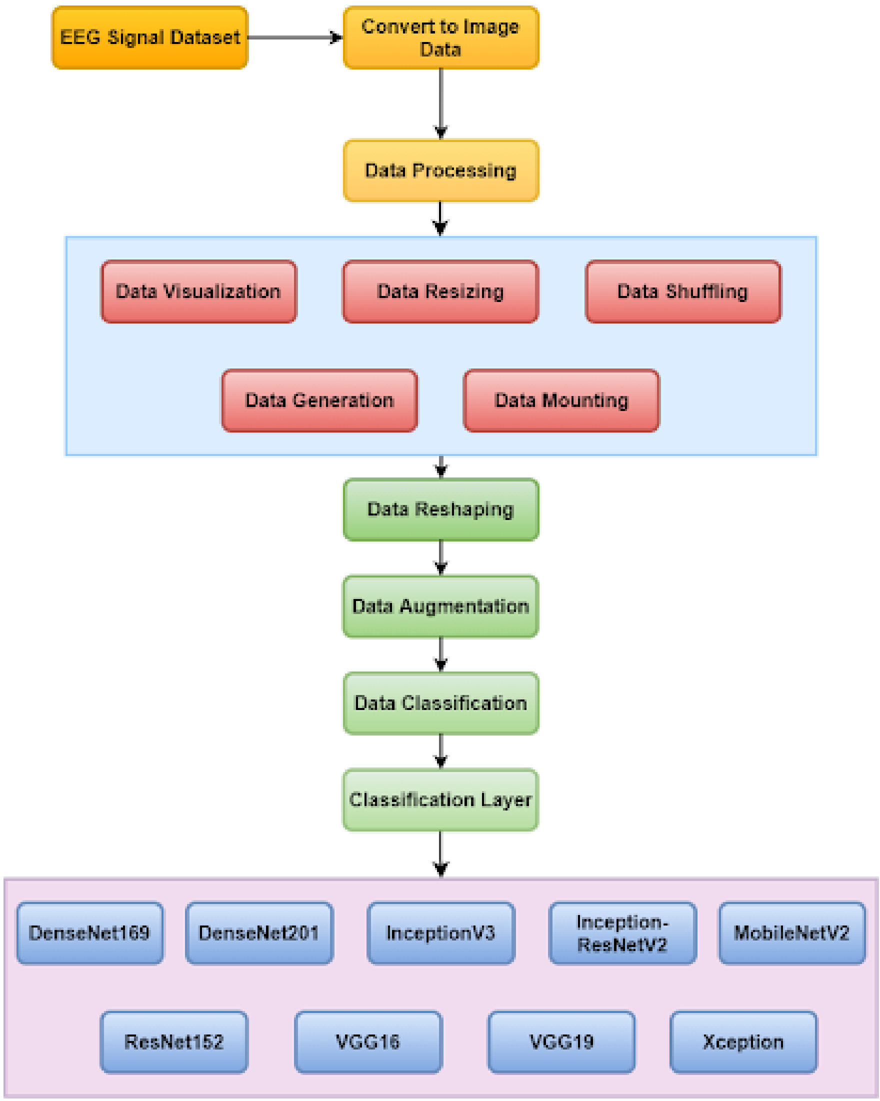

Figure 4.1: Research Methodology

All of the steps are related to each other. Data collection, data input, data processing and data classification are the four components of our developed system model. Figure 4.1 depicts the machine's overall architecture and all of its components.

4.1 Data Collection

This dataset [74] contains 193 EEG brain signals from two different individuals. Signals are classified into one of two categories: "MDD," which refers for "Major Depressive Disorder," and "Healthy," which stands for "Healthy Controls." Furthermore, each of the categories has its own set of subcategories. The letters "EO" stand for "Eyes Open," "EC" for "Eyes Closed," and "Task" for "P300 data," which stands for "event-related potential (ERP) endogenous." Moreover, we obtained this data through "Figshare," an open-access online repository where researchers can save and share their research outputs, such as figures, datasets, photos, and videos. This dataset has received over 4000 downloads and over 3500 views. This datasets EEG signals have some properties like:

Channels: 20/22 (Brain regions where the signals were taken from)
Sampling Frequency: 256.0 Hz
Data Shape: Value varies in a wide range
Here is an overview of our dataset:

	MDD = Major Depressive Disorder	**H = Healthy Controls**
EO	32	30
EC	30	28
Task	33	28

Table 4.1: Dataset table for number of sample patients

4.2 Data Input

The first step of our task is we need to import our dataset to analyze its type. For that we used the Jupiter Anaconda text editor. Our dataset's data type was "EDF" format which is known as "European Data Format". The EDF file format is a standard file format for exchanging and storing medical time series. We analyzed some data and came to know that there are 22 overall channels in our brain where researchers collect the EEG signal [75]. Here is a visual version (Figure 4.2) or those areas in our brain:

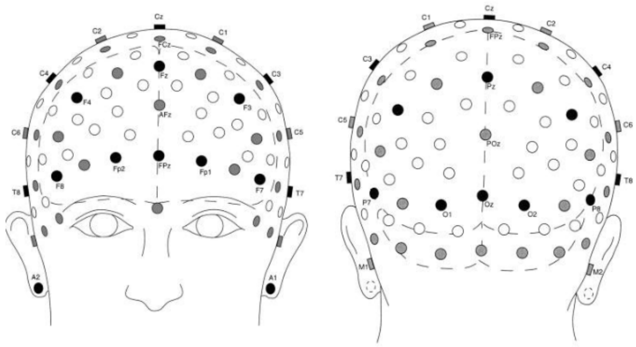

Figure 4.2: EEG Signal Locations

In some cases, we noticed the number of channels available on our data is 20. Hereby, in a few cases a couple of channels were left out. Furthermore, we learned a few more facts about our data. For example, the total duration of each data which varies from five minutes to ten minutes and sampling frequency.

4.3 Data Pre-Processing

This stage is concerned with formatting the input data in a way that we can analyze it more deeply and make it fitting for our work.

After importing our dataset, we need to label each of our data. EEG signals are generally displayed as a graph where "Brain Channel" is represented as "Rows" and "Time" is represented as "Columns". We run every single data one by one and from there we get two corresponding graphs for each data we have. We converted each graph as an image and saved it for our implementation phase. Each image is labeled as the respective unique subject code. For example, the loaded EEG graphs for data "MDD EO 17" means the graphs of this EEG data are saved as two images which are labeled as "MDD EO 17 (1)" and "MDD EO 17 (2)". The program will iterate through all data saved in different folders and do the same thing for all the data we have. This below figure (Figure 4.3) is an example:

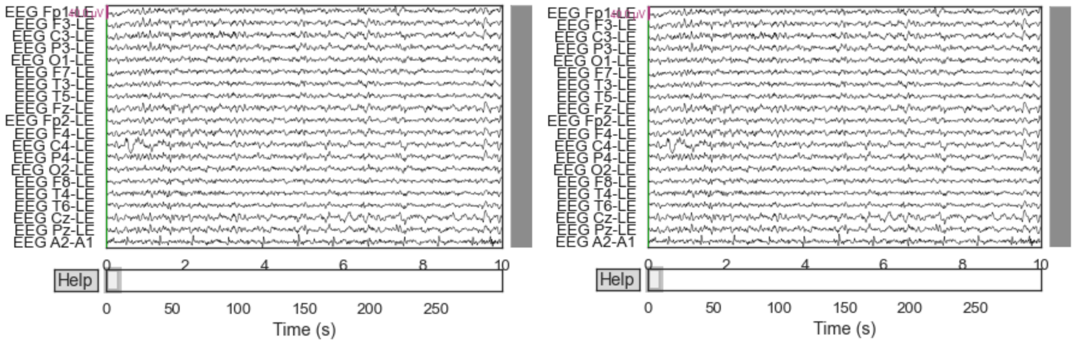

Figure 4.3: EEG Signal Graph

4.4 Data Processing

The term "processing data" refers to any adjustments made to raw data before it is processed by the profound learning module. Training a convolutional neural network on raw images, for example, is likely to result in poor classification. Before being passed to the following ResNet152, ResNet152V2, VGG16, VGG19, InceptionResNetV2, InceptionV3, Xception, DenseNet169, DenseNet201, MobileNetV2, 2D CNN (with and without Augmentation) module, this study ran through a series of seven pre-processing processes.

4.4.1 Data Mounting

This stage works as a virtual machine, comparable to a USB drive, on the Windows operating system to install a Google Drive account (DGA) for searching and accessing the drive at the Google Co-laboratory. Our Figshare dataset was imported into a directory and then extracted (referred to as a master) on Google Drive. It also made use of the Python/glob module, which allows for data processing such as framing, reading and writing between data in-memories and format structures, as well as accessing data in Co-Lab. We can also use the Python/Glop package to retrieve the Figshare dataset.

4.4.2 Data Resizing

This phase is required to reduce redundancies in input data that primarily increase the computation time of the same network without causing major changes. This is done during pre-processing with the help of Python/Keras. We were able to finish with an image size of 451x291 because of using several image sizes, with a feasible compute complexity while maintaining image readability.

4.4.3 Data Shuffling

The training data set's data samples are re-divided in this step to ensure that each sample of the data generates an unbiased transformation in the model. Preprocessing library can be achieved using Python/Keras.2.

4.4.4 Data Visualization

By showing several sample sizes of the training dataset using 2D models connected with the new pixel dimensions, this stage is required to sample and analyze the input data in order to ensure the accessibility of the input images. This can be done with the Python/TensorFlow and Python/NumPy libraries.

4.4.5 Data Generation

This step is used to generate batches of tensor image data that are augmented in real time. The data would be repeated (in batches) for both test and training. For sample normalized pictures and encoded labels, batch normalization is merged with picture plotting in this step.

4.5 Data Reshaping

This phase can be used to suit the input structure for our pre-processed data set image with different width and height as our main goal is to compare results with different scenarios. We also manipulated the layers of input ResNet152, ResNet152V2, VGG16, VGG19, InceptionResNetV2, InceptionV3, Xception, DenseNet169, DenseNet201, MobileNetV2. This may be done with the Python/Keras libraries.

4.6 Data Augmentation

In data analysis, data augmentation refers to procedures that add slightly changed copies of current data or newly created synthetic data from existing data to expand the amount of data available. When training a machine learning model, it functions as a regularizer and helps reduce overfitting. Oversampling in data analysis is directly related to it [76]. Firstly, we did not think data augmentation would be needed for our work. Besides, when we started collecting dataset, we noticed that there are not too many resources about this topic. In addition, this kind of data is usually confidential. That being the case, we did not manage to gather that much data. Therefore, we thought if we apply data augmentation to both increase the amount of data and hope for a better accuracy rate. We eliminated the junk values (parameters, letters) and kept only the signal part for this part. Another way of data augmentation is we can rotate the pictures.

4.7 Data Classification

Classification of data is an important technique in which datasets may be separated into groups for judgments, pattern identification and other purposes. A classification layer employs the completely implemented layer that estimates the loss of entropy for problems of multi-class classification for each other. It is possible to use Python/keras.layers, Python/keras.models, and Python/Keras/optimizers

Here we basically classified our dataset into two types of sets. Firstly, for the 2D CNN model we created two classes named “Healthy” and “MDD”. Moreover, for other models of convolutional layers, we further classified our dataset into “Train” and “Test” for each of the two mentioned classes “Healthy” and “MDD”.

Chapter 5

Model Implementation

The implementation model consists of a set of components as well as the implementation subsystems that house them. Both deliverable components, such as executables, and components from which deliverables are created, such as source code files, are considered components [77].

5.1 Workflow Overview

In order to obtain the best possible outcome, it is important to choose the correct pattern of work. As our initial goal is to compare among different models to comprehend how the implemented models are giving results and which model(s) might be the best for this experiment. Following is an overview of our Workflow. The details are as follows:

- Dataset is collected from "Figshare," an open-access online repository. The type of dataset was EDF from which we generated the EEG graphs. These graphs are used in the models.
- Dataset is cleaned and processed. This includes labeling, augmenting and feature scaling.
- The 2D CNN model was developed with and without data augmentation for detection.
- The train test split is 80:20, with the test set being held out for review in the final model.
- More deep learning models such as: ResNet, InceptionV3, VGG16 and some others have been implemented using Keras Application. Keras Applications are deep learning models that come with weights that have already been trained. Prediction, feature extraction, and fine-tuning are all possible with these models [78].
- Comparing the results of the models to analyze which model performs better.
- Analyzing the best model and customizing its layers to present a more effective model.

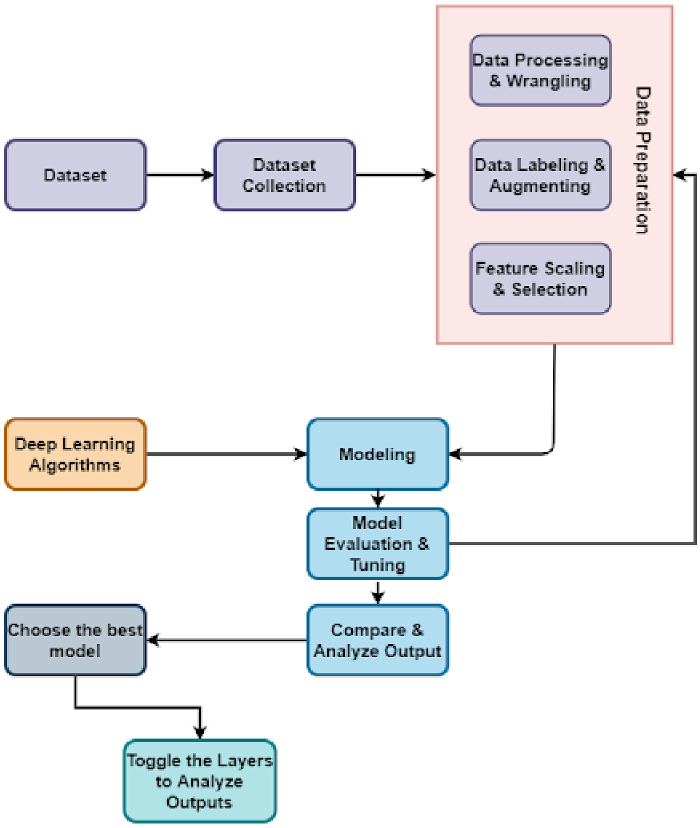

Figure 5.1: Workflow Diagram

5.2 Training Set

Dividing a dataset into training and testing is critical for model evaluation and helps to better understand model properties. The train dataset is used to fit the machine-learning model. We used 80% of our data from both "Healthy" and "MDD" for training the model.

5.3 Test Set

The test dataset, on the other hand, is used to evaluate the machine-learning model. We can find out how effective/efficient our model is. We can make change(s) to our model if we need to by analyzing the test result. We used 20% of our data from both "Healthy" and "MDD" for training the model.

5.4 Validation Set

A validation set is a set of data used to train artificial intelligence (AI) in order to find and develop the best model to solve a certain problem. Validation sets are also known as development sets. A supervised artificial intelligence (AI) learns from a set of training data. We also used our test set as a validation set in a few circumstances.

5.5 Train-Test Split

On the basic 2D CNN model, we applied a train-test split to perform the model. We divided the dataset into Train and Test using the Scikit-learn module. Although the typical ratio is 30% test set and 70% train set, our algorithms have adopted an 80/20 ratio for greater performance. As a result, we have kept the standard and chosen healthy (X) and a target (Y = "MDD") before splitting the dataset into Train Test splits. The training dataset will create an unknown model or a trained model, and the test dataset will assess its accuracy.

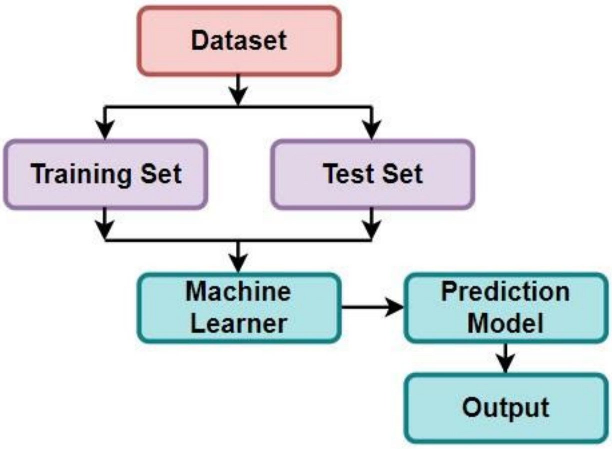

Figure 5.2: Train-Test Split

Chapter 6

Challenges

In our thesis, we have decided to implement a number of models. For that, we had to go through many challenges, which we could somehow overcome at the end. Since we had the idea to work on so many models so the optimum performance was very important in our case. For this we needed to emphasize on the datasets mainly which was one of the challenges we faced.

6.1 Data Dependency

It is a drawback of CNN that it needs large training data to give the desired result. The more the data set the more we will get optimum performance. Moreover, if enough data is not given in the model then the output will be biased on the training set and we will not get the correct result according to our trained model. So in deep learning, to achieve maximum optimum performance, enough quality data set is needed. Otherwise, we will have to face the issue of overfitting.

6.2 Overfitting

It happens when any model fits really well in a training set. The model thus has a hard time generalizing to new examples that were not in the training set [79]. It occurs when the dataset is insufficient in relation to the network's depth. On the split training set, the network becomes biased and is unable to reliably recognize the testing set. To address the problem, various strategies such as data augmentation, early halting, and parameter regularization can be applied. The dataset in our situation was small during the first phase of the model's training. So we faced this problem of overfitting. Later on, we used data augmentation to increase the number of datasets and solved this issue.

6.3 Excessive Training Time

We had no access to a high-performance machine with a GPU. For this reason, we used Google Collaboratory to run our code and implement the respective models. We faced various network issues and complications in this case. Still in most cases per model, we needed 20-25 minutes which is really time consuming.

6.4 Data Leakage

When constructing predictive models with machine learning, data leaking is a major issue. When information from outside the training dataset is used to generate the model, this is known as data leakage [80]. This occurs due to not properly splitting the dataset for the training and testing cases. When same data is split across the training cases and testing cases, this can also happen. This problem masks the actual performance, which deteriorates as it becomes accustomed to fresh input.

6.5 Gradient Vanishing

When the derivatives that were supposed to be multiplied by the exact amount of hidden layer are lower than expected; when back propagating, the gradients will steadily decrease and eventually vanish. Sometimes the loss function becomes so small or also zero that the model is unable to distinguish between previous and updated weights.

6.6 How did we Overcome?

Since depression is a very severe issue in today's world, we have decided to work on this and have the assurance of availability of enough datasets. Our first and foremost challenge was to gather an appropriate dataset. Then somehow, we managed to collect data from Kaggle but that too was not enough for our thesis. After that, we split our dataset to a train set, a test set, and to get better results we had to do the data augmentation. Otherwise, we would have to face the problem of overfitting.

Another challenge was to train the datasets because we had such machines that lacked GPU so it took so long to train the datasets. Moreover, our work is based on a comparative analysis. In some cases we had to find the necessary articles or papers based on which we are going to do the respective analysis of the models.

Overall, we had overcome these challenges within the end of our research and finally implemented our comparison.

Chapter 7

Comparative Results and Proposed Model

Two or more things or concepts are investigated and compared in a comparative study. Investigations, comparisons, and contrasts of subjects or concepts are all part of comparative studies. Comparative research illustrates how two subjects are similar or different [81]. Specifically, in our research, we used comparative analysis to know which model works better on our dataset. Later, that analysis is used while making our proposed model.

7.1 Results of Supervised Models

First, we applied a very basic CNN model to our dataset. Initially we applied 2D CNN. Later to improve the accuracy level, we applied data augmentation. Our basic testing CNN model has only 7-8 (we toggled a bit for experiment) layers as we just wanted to understand the CNN neural network.

This is the result we got from our model:

2D CNN	Value Loss	Accuracy (Highest)
Without Augmentation	0.6082	78%
With Augmentation	0.2741	89%

Table 7.1: Basic CNN model result

Then, we applied our dataset to a few well-known existing deep learning models using Keras Application. Keras Applications are deep learning models that come with weights that have already been trained. Prediction, feature extraction, and fine-tuning are all possible with these models [82]. Weights of each model are automatically downloaded when we initialize the models. Here we applied the following deep learning models using application:

DenseNet169	DenseNet201	InceptionV3
InceptionResNetV2	MobileNetV2	ResNet152
VGG16	VGG19	Xception

Table 7.2: Used Deep Learning Models

Firstly, our collected dataset was EEG data which was in EDF format. We converted it to image data for our convenience. In our first attempt (Table 7.3) we used the actual image input size (Size = 451x291) to see how it performs.

Deep Learning Models	Value Loss	Accuracy (Highest)
DenseNet169	1.0012	89%
DenseNet201	0.8590	92%
InceptionV3	1.9105	90%
InceptionResNetV2	0.9907	85%
MobileNetV2	2.1014	90%
ResNet152	0.7027	74%
VGG16	**0.9133**	**95%**
VGG19	0.5101	91%
Xception	1.0973	86%

Table 7.3: Deep Learning Model result with Input size = 451x291

As we can see, the accuracy level is a bit lower than usual. The highest accuracy we got is 95% from the model VGG16.

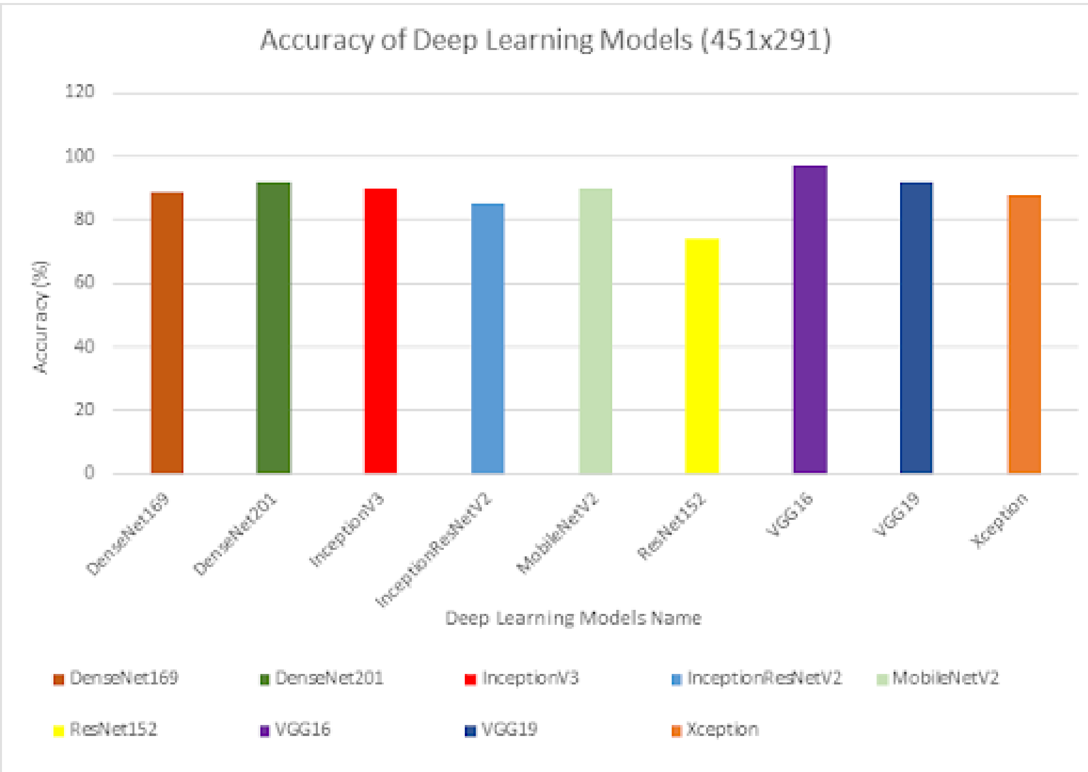

Figure 7.1: Accuracy of Deep Learning Models (451x291)

Thus, we decided to apply a different Input size. This time we took input size 224x224. The reason for choosing this input size is, from the previous table we can see VGG16 was giving the best accuracy, which is 94.4% among all models. Therefore, we targeted this particular model and found that the base input size of the VGG16 model is 224x224. This is why we decided to apply this input size to see how it performs. The following table (Table 7.4) shows the outputs we got for each respective model:

Deep Learning Models	Value Loss	Accuracy (Highest)
DenseNet169	0.9921	91%
DenseNet201	0.8471	93%
InceptionV3	1.6879	91%
InceptionResNetV2	0.9383	89%
MobileNetV2	1.6837	92%
ResNet152	0.5131	78%
VGG16	**0.0804**	**97%**
VGG19	0.2148	92%
Xception	1.0458	88%

Table 7.4: Deep Learning Model result with Input size = 224x224

As we can see, the accuracy level has increased. The highest accuracy we got is 97% from the model VGG16.

7.2 Analysis of Comparative Results

Comparing results with previous research and papers is a good way to show how our research has improved or aggravated the task. Wajid Mumtaz originally posted the dataset we used on “Figshare”. He and his fellow researchers also used this dataset to conduct a few researches.

Firstly, we will compare our results with the research papers, which share the same dataset and goal. Then we will compare the results of the models that we experimented and so far, not used with this dataset to see the output.

Finally, we will implement the VGG16 model, which gives us the highest accuracy. Then we will try to customize the model by adding or removing convolutional layers, adding or removing fully connected layers, increasing or decreasing the dropout and learning ratio to find out which scenario suits our case the best and gives the best accuracy. Moreover, we will try to suggest an optimized model considering all the scenarios.

7.2.1 Previous Researches on this Dataset

Previously there were few researches conducted based on this dataset. Now we will try to compare their results based on our models.

In the paper, "A deep learning framework for automatic diagnosis of unipolar depression", Wajid Mumtaz and Abdul Qayyum worked with different portions of the dataset to detect depression [83]. In one attempt, they took 15 depressive and 15 healthy cases and the accuracy they achieved is 93%. This deep learning model has 13 layers CNN. They also used the 1D CNN model here.

Other papers used different deep learning models to detect depression or for some other purposes. We are going to implement CNN models and measure how far we can get. We are going to use a 2D CNN model.

7.2.2 Deep learning Models Comparison

If we try to go back to our "Table 7.5", we will see the results we got from our deep learning models' accuracy performance on our task. Each model's accuracy varies as their number of layers, parameters are different.

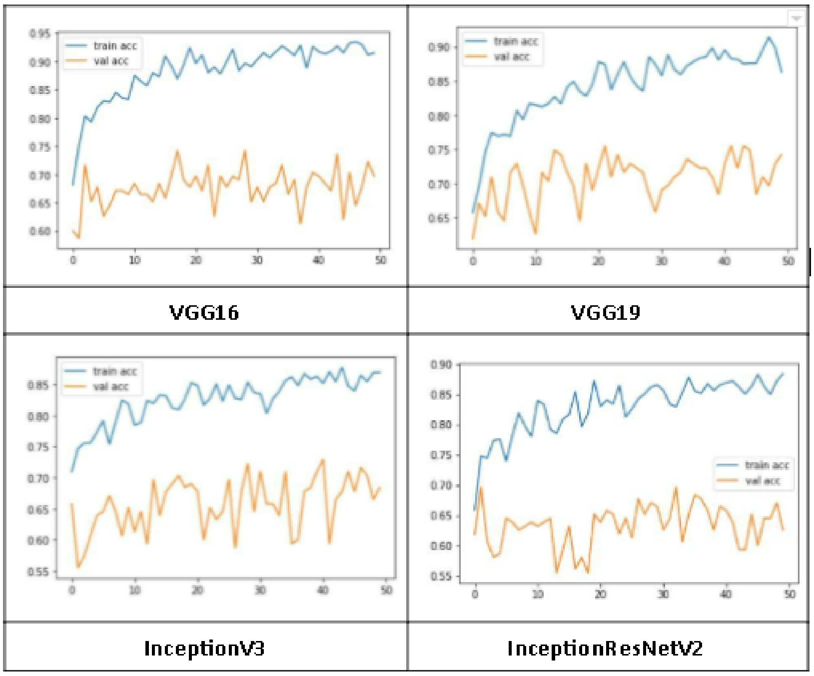

Figure 7.2: Accuracy Graphs of Deep Learning Models (1)

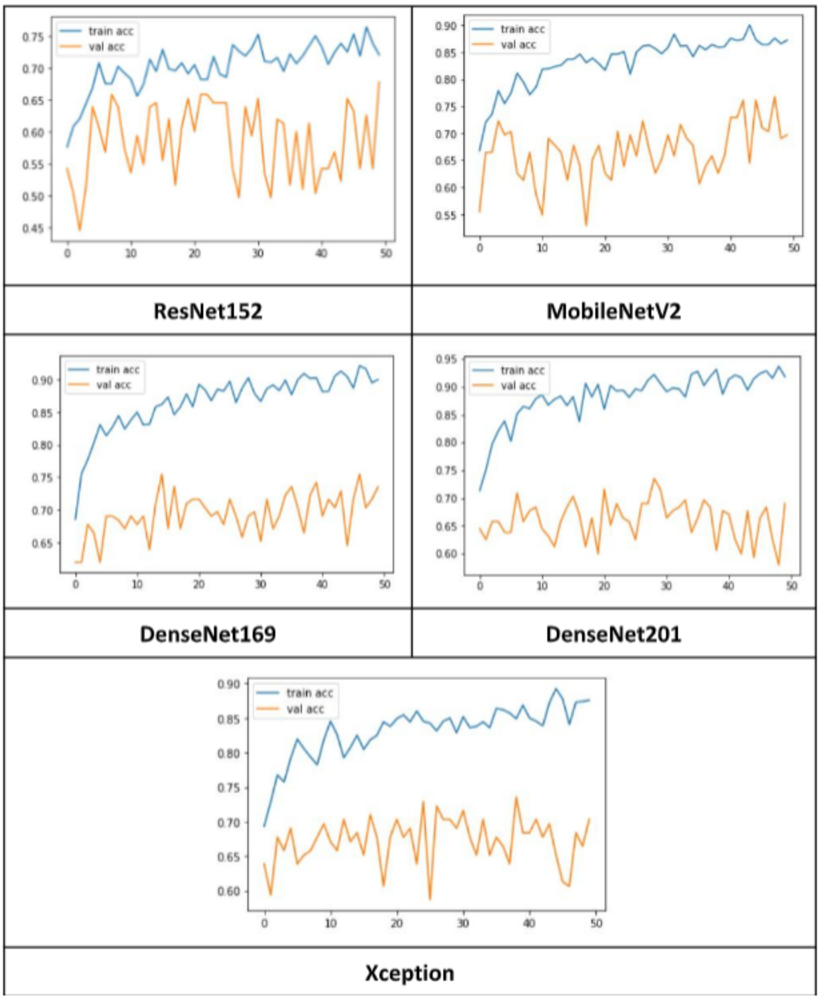

Figure 7.3: Accuracy Graphs of Deep Learning Models (2)

Highest accuracy we got from model VGG16 which gave us 97% accuracy with value loss 0.804. It has 23 layers with a number of 138,357,544 parameters.

The second highest accuracy we got is from the DenseNet201 model, which gave us 93% accuracy with value loss 0.8471. It has 23 layers with a number of 143,667,240 parameters.

On the other hand, the lowest accuracy we got from the ResNet152 model. The accuracy we got from ResNet152 was 78%. It has up to 152 layers with 60,419,944 parameters.

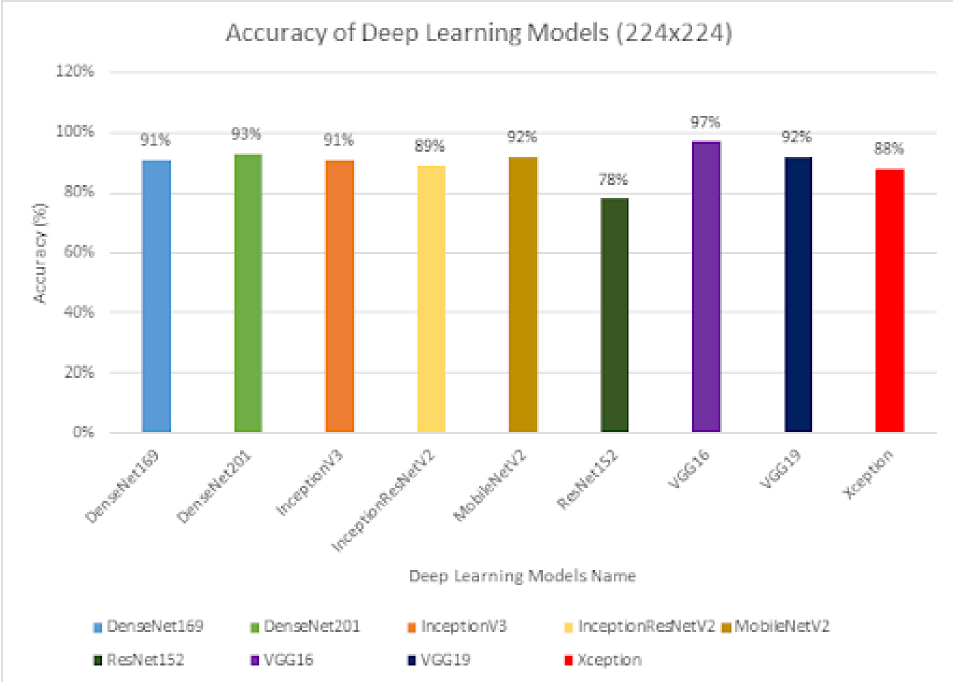

Figure 7.4: Accuracy of Deep Learning Models (224x224)

From these observations, despite having the lowest depth in terms of layers (23 Layers) it gave the highest accuracy. If we compare it with our other models with respect to layers, we can see the model with highest layers is InceptionResnetV2 with 512 layers. Nevertheless, the accuracy we got from this model was 89%. Which is a bit low with respect to VGG16.

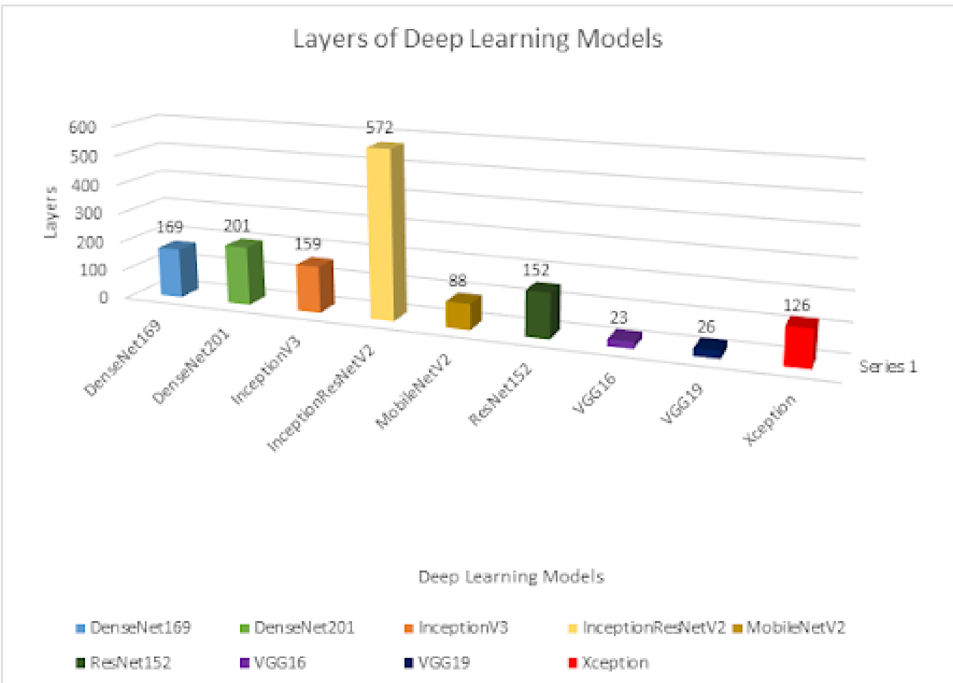

Figure 7.5: Number of Layers and Accuracy on Deep Learning Models

Therefore, the models' accuracy does not depend on any specific variable. We always need to tweak things and change value to monitor when its accuracy is increasing or decreasing. After so many attempts of trial and error, a model can be perfect.

7.3 Proposed Customized Model

We tried to propose a new model for this dataset to detect depression. We started by making use of the VGG16 model systematically. As it gave us the highest accuracy but when we ran our dataset to our custom VGG16 model, its accuracy was not up to the mark. It is because in the available built-in model, the model is optimized. Besides, it is difficult to make an optimized model step by step. That was why we did not get the accuracy we got in the VGG16 model.

Then at first, we tried to add more layers to the problem. However, the accuracy declined. We tweaked a few parameters and added some built-in methods for example dropout, batch normalization, optimizer-learning algorithm: AdaM to see how it reacts.

While tweaking the parameters, we observed a few patterns. They are if we increase the number of convolutional layers the accuracy usually decreases. Nonetheless, if we remove a few layers the accuracy increases. Additionally, in case of dropout, the more the dropout rate the lower the accuracy and vice versa.

Other methods like batch normalization and optimizer algorithms do not have that kind of consistency in case of increasing or decreasing. In case of convolutional layers and dropout rate, it always stayed in the flow.

One thing we need to mention here is, the pattern we are seeing in case of layers and dropouts, specifically based on our dataset. There is no warranty that the models will always behave like this. It may vary from model to model and dataset to dataset.

Our assumption is, as our dataset is very small if we compare it to other datasets, is the reason behind this.

Finally, after a lot of test runs, we found a model giving the best accuracy among all models. A deep convolutional neural network that seems to work really well. We used convolutional layers, pooling layers, fully connected layers with batch normalization, dropouts.

This is the architecture of our proposed model:

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 253, 253, 16)	448
max_pooling2d (MaxPooling2D)	(None, 126, 126, 16)	0
batch_normalization (BatchNo	(None, 126, 126, 16)	64
dropout (Dropout)	(None, 126, 126, 16)	0
conv2d_1 (Conv2D)	(None, 124, 124, 32)	4640
max_pooling2d_1 (MaxPooling2	(None, 62, 62, 32)	0
batch_normalization_1 (Batch	(None, 62, 62, 32)	128
dropout_1 (Dropout)	(None, 62, 62, 32)	0
conv2d_2 (Conv2D)	(None, 60, 60, 48)	13872
max_pooling2d_2 (MaxPooling2	(None, 30, 30, 48)	0
batch_normalization_2 (Batch	(None, 30, 30, 48)	192
dropout_2 (Dropout)	(None, 30, 30, 48)	0
dropout_3 (Dropout)	(None, 30, 30, 48)	0
flatten (Flatten)	(None, 43200)	0
dense (Dense)	(None, 64)	2764864
flatten_1 (Flatten)	(None, 64)	0
dense_1 (Dense)	(None, 2)	130

Total params: 907,346
Trainable params: 907,154
Non-trainable params: 192

Figure 7.6: Architecture of Proposed Model

The highest accuracy we got from this model is 99.75% with a value loss 0.0122.

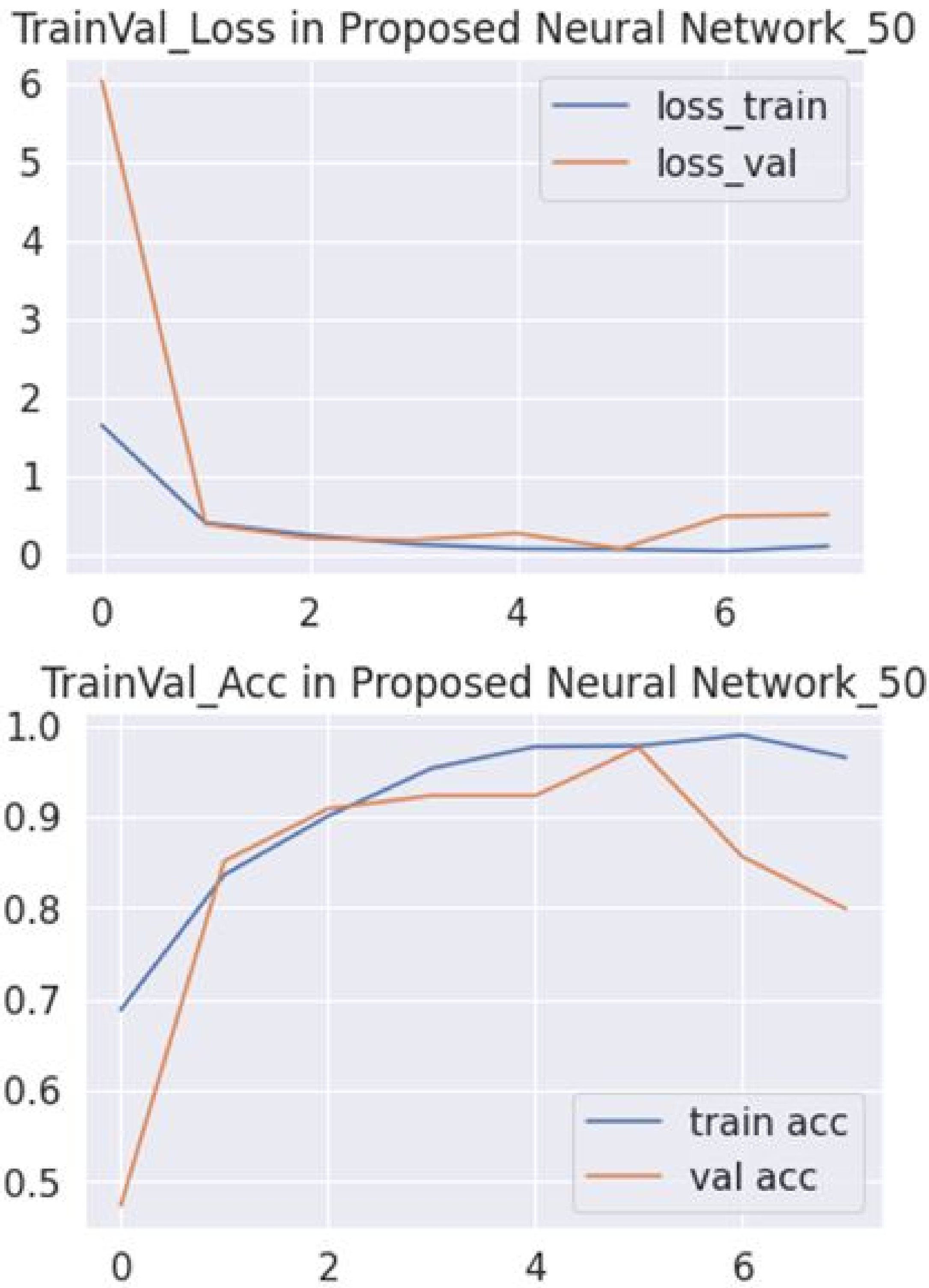

Figure 7.7: Graph of Train Value Loss Accuracy

In our research, our proposed deep convolutional neural network model has got 99.75% test accuracy with value loss of 0.0122 for detecting depression based on our dataset. While on the available models and on the previous research based on this dataset wasn't able to achieve this level of accuracy. Also, our test accuracy is more significant if we compare it to the other previously used models in past research.

Epoch	Training Loss	Training Accuracy(%)
1	2.4516	64.94
2	0.4708	76.29
3	0.3663	85.01
4	0.1820	92.63
5	0.0546	97.88
6	0.0319	98.90
7	0.0220	99.32
8	0.0118	99.75

Table 7.5: Training Accuracy and Training Loss of Proposed Model

Proposed Convolutional Neural Network Model	
Testing Accuracy (Highest)	Testing Loss
99.75%	0.0122

Table 7.6: Testing Accuracy and Testing Loss of Proposed CNN Model

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Depression has a high likelihood of morbidity and mortality when left untreated. The majority of people who have depression do not feel depressed; instead, they experience anhedonia or other vague, unexplained symptoms. The main problem with depressed people is that they are mostly afraid or do not want to talk about it with others and take necessary treatments.

However, there are many treatments available these days but those are not applicable for long-term issues. On a national level, steps must be taken to eradicate this disease. Nevertheless, for starting treatment, detecting it at the shortest possible time is very important. In our thesis we have implemented various models to detect depression and found the best one among them based on the accuracy each of them give. Moreover, we are doing comparative analysis between the respective models we have implemented based on the works that were done earlier with these models. Our goal is to find which one works better and the reason behind this. As a result it will become helpful for the people by choosing the most accurate and efficient model, who wants to research on this topic in future. In addition, it can be helpful for early detection of depression for the MDD patients. In conclusion, if the MDD patient starts taking medication from the beginning phase then there are high chances to be cured from this severe disease.

8.2 Future Work

In our thesis, we have found the best model out of the existing ones to detect depression as early as possible compared to others. However, we have some more to work on this project. Our future goal is to find a probable treatment for the MDD patients.

Our idea will be to develop a chip for the brain of an MDD patient. So, when a depressed patient will be going through a severe condition especially when he will have suicidal thoughts, panic attack or something upsetting at that time the chip is going to collect a distress signal from the patient and thus it will understand the pattern of the signal and also the patient's condition. Thus, the chip will alter this signal and send another signal to the neuron of the patient that will make him happy or content for that particular time and it will remove the suicidal thought from his mind.

We are hopeful about this work in future since it can be applicable in many cases to cure people.

Bibliography

[1] "Depression (major depressive disorder)." https://www.mayoclinic.org/diseases-conditions/depression/symptoms-causes/syc-20356007. Accessed: 2021-9-25.

[2] https://nimhbd.com/. Accessed: 2021-9-25.

[3] https://adaa.org/learn-from-us/from-the-experts/blog-posts/consumer/depression-and-daily-life. Accessed: 2021-9-25.

[4] S. Dattani, H. Ritchie, and M. Roser, "Mental health," *Our World in Data*, 2021.

[5] "Depression." https://www.who.int/westernpacific/health-topics/depression. Accessed: 2021-9-25.

[6] M. D. Hossain, H. U. Ahmed, W. A. Chowdhury, L. W. Niessen, and D. S. Alam, "Mental disorders in bangladesh: a systematic review," *BMC Psychiatry*, vol. 14, no. 1, p. 216, 2014.

[7] S. Levin, "Facebook admits it poses mental health risk – but says using site more can help," *Guardian*, Dec. 2017.

[8] "16.8% bangladeshi adults suffer from mental health issues." https://www.dhakatribune.com/health/2019/11/07/16-8-bangladeshi-adults-suffer-from-mental-health-issues, Nov. 2019. Accessed: 2021-9-25.

[9] N. A. Giasuddin, N. F. Chowdhury, N. Hashimoto, D. Fujisawa, and S. Waheed, "Pathways to psychiatric care in bangladesh," *Soc. Psychiatry Psychiatr. Epidemiol.*, vol. 47, no. 1, pp. 129–136, 2012.

[10] A. K. Chowdhury, M. N. Alam, and S. M. Ali, "Dasherkandi project studies. demography, morbidity and mortality in a rural community of bangladesh," *Bangladesh Med. Res. Counc. Bull.*, vol. 7, no. 1, pp. 22–39, 1981.

[11] *Prevalence, medical care, awareness, and attitude towards mental illness in Bangladesh.* National Institute of Mental Health & Hospital.

[12] M. A. Mohit, M. M. Maruf, H. Ahmed, and M. T. Alam, "Depression and physical illnesses: An update," *Bangladesh Med. J.*, vol. 40, no. 1, pp. 53–58, 1970.

[13] "Depression." https://www.who.int/health-topics/depression. Accessed: 2021-9-24.

[14] https://www.who.int/teams/mental-health-and-substance-use/about. Accessed: 2021-9-24.

[15] Z. Halim and M. Rehan, "On identification of driving-induced stress using electroencephalogram signals: A framework based on wearable safety-critical scheme and machine learning," *Inf. Fusion*, vol. 53, pp. 66–79, 2020.

[16] U. R. Acharya, V. K. Sudarshan, H. Adeli, J. Santhosh, J. E. W. Koh, and A. Adeli, "Computer-aided diagnosis of depression using EEG signals," *Eur. Neurol.*, vol. 73, no. 5-6, pp. 329–336, 2015.

[17] S. P. Bobde, S. T. Mantri, D. D. Patil, and V. Wadhai, "Cognitive depression detection methodology using EEG signal analysis," in *Intelligent Computing and Information and Communication*, pp. 557–566, Singapore: Springer Singapore, 2018.

[18] M. J. Mary Juliana and N. Rangaswamy, "A neoteric approach towards anxiety quantification by EEG signal analysis," in *2018 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, pp. 51–59, IEEE, 2018.

[19] H. Cai, J. Han, Y. Chen, X. Sha, Z. Wang, B. Hu, J. Yang, L. Feng, Z. Ding, Y. Chen, and J. Gutknecht, "A pervasive approach to EEG-based depression detection," *Complexity*, vol. 2018, pp. 1–13, 2018.

[20] W. Mumtaz, S. S. A. Ali, M. A. M. Yasin, and A. S. Malik, "A machine learning framework involving EEG-based functional connectivity to diagnose major depressive disorder (MDD)," *Med. Biol. Eng. Comput.*, vol. 56, no. 2, pp. 233–246, 2018.

[21] M. Mousavian, J. Chen, and S. Greening, "Depression detection using feature extraction and deep learning from sMRI images," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1731–1736, IEEE, 2019.

[22] C. Uyulan, T. T. Ergüzel, H. Unubol, M. Cebi, G. H. Sayar, M. Nezhad Asad, and N. Tarhan, "Major depressive disorder classification based on different convolutional neural network models: Deep learning approach," *Clin. EEG Neurosci.*, vol. 52, no. 1, pp. 38–51, 2021.

[23] M. Rodrigues Makiuchi, T. Warnita, K. Uto, and K. Shinoda, "Multimodal fusion of BERT-CNN and gated CNN representations for depression detection," in *Proceedings of the 9th International on Audio/Visual Emotion Challenge and Workshop - AVEC '19*, (New York, New York, USA), ACM Press, 2019.

[24] D. Ahmad, R. Goecke, and J. Ireland, "CNN depression severity level estimation from upper body vs. face-only images," in *Pattern Recognition. ICPR International Workshops and Challenges*, pp. 744–758, Cham: Springer International Publishing, 2021.

[25] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015.

[26] Arc, "Convolutional neural network - towards data science." https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05, Dec. 2018. Accessed: 2021-9-25.

[27] P. Skalski, "Gentle dive into math behind convolutional neural networks." https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9, Apr. 2019. Accessed: 2021-9-25.

[28] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, IEEE, 2017.

[29] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Stroudsburg, PA, USA), Association for Computational Linguistics, 2014.

[30] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, 2017.

[31] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *ArXiv e-prints*, 11 2015.

[32] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, 2017.

[33] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, vol. 1, 04 2014.

[34] H. S, "Activation Functions : Sigmoid, ReLU, Leaky ReLU and Softmax basics for Neural Networks and Deep Learning," 01 2020.

[35] S. Sharma, "Activation Functions in Neural Networks - Towards Data Science," 12 2019.

[36] K. Srinivasan, A. K. Cherukuri, D. Vincent P M, A. Garg, and B.-Y. Chen, "An efficient implementation of artificial neural networks with k-fold cross-validation for process optimization," *Journal of Internet Technology*, vol. 20, pp. 1213–1225, 06 2019.

[37] H. S, "Activation Functions : Sigmoid, ReLU, Leaky ReLU and Softmax basics for Neural Networks and Deep Learning," 01 2020.

[38] H. S, "Activation Functions : Sigmoid, ReLU, Leaky ReLU and Softmax basics for Neural Networks and Deep Learning," 01 2020.

[39] "Understanding loss functions in machine learning." https://www.section.io/engineering-education/understanding-loss-functions-in-machine-learning/. Accessed: 2021-9-25.

[40] "Understanding categorical Cross-Entropy loss, binary Cross-Entropy loss, softmax loss, logistic loss, focal loss and all those confusing names." https://gombru.github.io/2018/05/23/cross_entropy_loss/. Accessed: 2021-9-25.

[41] J. Brownlee, "A gentle introduction to cross-entropy for machine learning," *Machine learning mastery*, vol. 20, 2019.

[42] R. Gupta, "6 deep learning models — when should you use them? - towards data science." https://towardsdatascience.com/6-deep-learning-models-10d20afec175, Oct. 2019. Accessed: 2021-9-25.

[43] S. Mahapatra, "Why deep learning over traditional machine learning?." https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063, Mar. 2018. Accessed: 2021-9-25.

[44] Sudhir_ and Praveen_, "Why convolutional neural network is better? - intellipaat community." https://intellipaat.com/community/46830/why-convolutional-neural-network-is-better. Accessed: 2021-9-25.

[45] S. Mahapatra, "A simple 2D CNN for MNIST digit recognition." https://towardsdatascience.com/a-simple-2d-cnn-for-mnist-digit-recognition-a998dbc1e79a, May 2018. Accessed: 2021-9-25.

[46] S. Verma, "Understanding 1D and 3D convolution neural network." https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610, Sept. 2019. Accessed: 2021-9-25.

[47] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017.

[48] P. Ruiz, "Understanding and visualizing DenseNets - towards data science." https://towardsdatascience.com/understanding-and-visualizing-densenets-7f688092391a, Oct. 2018. Accessed: 2021-9-25.

[49] "densenet-169 - OpenVINO™ toolkit." https://docs.openvinotoolkit.org/latest/omz_models_model_densenet_169.html. Accessed: 2021-9-25.

[50] "DenseNet-201 convolutional neural network - MATLAB densenet201." https://www.mathworks.com/help/deeplearning/ref/densenet201.html. Accessed: 2021-9-25.

[51] "densenet-201 - OpenVINO™ toolkit." https://docs.openvinotoolkit.org/2021.3/omz_models_model_densenet_201.html. Accessed: 2021-9-25.

[52] Keras Team, "Keras applications." https://keras.io/api/applications/?fbclid=IwAR3qN9fXW-lEpLHH-8jrXDROGq4lh49-L_QCMCmeKvZN8wNXh0Ckslf7CM0. Accessed: 2021-9-25.

[53] L. D. Nguyen, D. Lin, Z. Lin, and J. Cao, "Deep cnns for microscopic image classification by exploiting transfer learning and feature concatenation," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, IEEE, 2018.

[54] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[55] Keras Team, "Keras applications." https://keras.io/api/applications/?fbclid=IwAR3qN9fXW-lEpLHH-8jrXDROGq4lh49-L_QCMCmeKvZN8wNXh0Ckslf7CM0. Accessed: 2021-9-25.

[56] "Pretrained Inception-ResNet-v2 convolutional neural network - MATLAB inceptionresnetv2." https://www.mathworks.com/help/deeplearning/ref/inceptionresnetv2.html. Accessed: 2021-9-25.

[57] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," 2016.

[58] "MobileNetV2." https://paperswithcode.com/method/mobilenetv2. Accessed: 2021-9-25.

[59] "MobileNet-v2 convolutional neural network - MATLAB mobilenetv2." https://www.mathworks.com/help/deeplearning/ref/mobilenetv2.html. Accessed: 2021-9-25.

[60] S.-H. Tsang, "Review: MobileNetV2 — light weight model (image classification)." https://towardsdatascience.com/review-mobilenetv2-light-weight-model-image-classification-8febb490e61c, May 2019. Accessed: 2021-9-25.

[61] S.-H. Tsang, "Review: ResNet — winner of ILSVRC 2015 (image classification, localization, detection)." https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015-image-classification-localization-detection-e39402bfa5d8, Sept. 2018. Accessed: 2021-9-25.

[62] Keras Team, "Keras applications." https://keras.io/api/applications/?fbclid=IwAR3qN9fXW-lEpLHH-8jrXDROGq4lh49-L_QCMCmeKvZN8wNXh0Ckslf7CM0. Accessed: 2021-9-25.

[63] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-vgg16 cnn model for big data places image recognition," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 169–175, IEEE, 2018.

[64] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[65] Keras Team, "Keras applications." https://keras.io/api/applications/?fbclid=IwAR3qN9fXW-lEpLHH-8jrXDROGq4lh49-L_QCMCmeKvZN8wNXh0Ckslf7CM0. Accessed: 2021-9-25.

[66] M. Shu, *Deep learning for image classification on very small datasets using transfer learning*. PhD thesis, Iowa State University, 2019.

[67] A. Kaushik, "Understanding the VGG19 architecture." https://iq.opengenus.org/vgg19-architecture/, Feb. 2020. Accessed: 2021-9-25.

[68] "#013 CNN VGG 16 and VGG 19 - master data science." http://datahacker.rs/deep-learning-vgg-16-vs-vgg-19/, Nov. 2018. Accessed: 2021-9-25.

[69] Keras Team, "Keras applications." https://keras.io/api/applications/?fbclid=IwAR3qN9fXW-lEpLHH-8jrXDROGq4lh49-L_QCMCmeKvZN8wNXh0Ckslf7CM0. Accessed: 2021-9-25.

[70] "XCeption model and depthwise separable convolutions." https://maelfabien.github.io/deeplearning/xception/, Mar. 2019. Accessed: 2021-9-25.

[71] "Xception convolutional neural network - MATLAB xception." https://www.mathworks.com/help/deeplearning/ref/xception.html. Accessed: 2021-9-25.

[72] S.-H. Tsang, "Review: Xception — with depthwise separable convolution, better than inception-v3 (image classification)." https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dc Sept. 2018. Accessed: 2021-9-25.

[73] Keras Team, "Keras applications." https://keras.io/api/applications/?fbclid=IwAR3qN9fXW-lEpLHH-8jrXDROGq4lh49-L_QCMCmeKvZN8wNXh0Ckslf7CM0. Accessed: 2021-9-25.

[74] W. Mumtaz, "MDD patients and healthy controls EEG data (new)," Nov. 2016.

[75] "EEG." https://www.medicine.mcgill.ca/physio/vlab/biomed_signals/eeg_n.htm. Accessed: 2021-9-25.

[76] Wikipedia contributors, "Data augmentation." https://en.wikipedia.org/w/index.php?title=Data_augmentation&oldid=1045265391, Sept. 2021. Accessed: 2021-9-25.

[77] "Artifact: Implementation model." https://sceweb.uhcl.edu/helm/RationalUnifiedProcess/process/artifact/ar_impmd.htm. Accessed: 2021-9-25.

[78] Keras Team, "Keras applications." https://keras.io/api/applications/. Accessed: 2021-9-25.

[79] "Difference between ANN, CNN and RNN." https://www.geeksforgeeks.org/difference-between-ann-cnn-and-rnn/, June 2020. Accessed: 2021-9-25.

[80] J. Brownlee, "Data leakage in machine learning." https://machinelearningmastery.com/data-leakage-machine-learning/, Aug. 2016. Accessed: 2021-9-25.

[81] S. A. H. Bukhari, “What is comparative study,” *SSRN Electron. J.*, 2011.

[82] Keras Team, “Keras applications.” https://keras.io/api/applications/. Accessed: 2021-9-25.

[83] A. P. White, A. M. Weljie, D. Apel, P. Zhang, R. Shaykhutdinov, H. J. Vogel, and M. G. Surette, “Metabolites with statistical difference in the wild-type s. typhimurium and csgd mutant colonies,” July 2010.