Detection of Diabetic Retinopathy Using Machine Learning

By

Koushik Chikkegowda

Student Number: 10515678

Dissertation submitted in part fulfilment of the requirement for the degree of master of science (M.Sc) in Data Analytics at Dublin Business School in conjunction with Quality and Qualifications Ireland (QQI).

Supervisor: Dr. Shahram Azizi Sazi

August 2020

DECLARATION

I, Koushik Chikkegowda, declare that my research for the degree of Master of Science in Data Analytics is my own personal work and that it has never been presented in any institution or college to serve the purpose for any Degree or Diploma. Furthermore, I took reasonable care to ensure the originality of my work to the best of my knowledge and that it did not in any manner breach any copyright laws. I hereby confirm that my work is entirely compliant with the Dublin Business School's academic honesty policy.

Signed: Koushik Chikkegowda
Date: 25/08/2020

ACKNOWLEDGMENTS

I would like want to thank my guide Dr. Shahram Azizi Sazi, who gave knowledge and skill that extraordinarily aided the research. As my instructor and coach, he has shown me beyond what I would ever give him the credit for here. He helped me in seeing profound adapting better. He has appeared, by model, of what a decent computer scientist researcher ought to be.

DIABETIC RETINOPATHY USING TRANSFER LEARNING

DEFINITIONS AND ACRONYMS

DR-Diabetic retinopathy -is a standout amongst the most widely recognized complexities of diabetes. It can create in individuals with both type 1 and type 2 diabetes, and as a rule influences the two eyes. Individuals with retinopathy have harmed veins in their retina

Artificial Intelligence -In computer science, artificial intelligence, sometimes called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans and other animals.

Deep Learning -Deep learning is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.

Machine Learning-Machine learning is an application of Artificial Intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

CNN -In deep learning, a convolution neural network is a class of deep neural networks, most normally connected to breaking down visual symbolism. CNNs utilize a variety of multilayer perceptron's intended to require negligible preprocessing

R-CNN -It shows the region of interest and it focuses on certain point

Transfer learning–It's a Machine Leaning technique where a trained model for any other data is repurposed for second task as to be retrained on another data with previous weights.

Training – A stage during compile-time where the CNN model identifies the dataset and classifies them according to their identification labels (if any).

Keras - Keras is deep learning framework or language which is built on theanos and tensorflow

ABSTRACT

Diabetic retinopathy (DR) is a disease that effect human eye with people having diabetes which make harm to the human eye that can further leads to blindness. It's a common eye disease. Early detection of DR and Proper treatment is necessary to prevent visual loss and further eye damage. There are different treatment methods, but they are much time taking and include many eye tests with manual treatment Procedure. Purpose of this Project is to introduce an automated learning model which shows less time taking results with maximum accuracy using CNN algorithms to classify eye images infected by sickness (DR). Detection of Diabetic retinopathy using Gaussian filtered images is the main objective of this paper. Five different classifications will be performed using transfer learning techniques of Deep convolutional neural network (DCNN). As early detection of Diabetic Retinopathy (DR) in healthcare domain can be helpful to prevent visual loss of human eye with early treatment/diagnosis. This report will also increase awareness of Diabetic retinopathy and automated learning techniques among people with diabetes, IT sector and healthcare sector.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Diabetic Retinopathy is the human eye disease with people having diabetes which makes harm to eyes and may even cause visual deficiency. Detection of Diabetic Retinopathy in early stages is essential to keep away from blindness.[1] There's been many treatment for this disease but it's a lot time taking and may even include many eye tests i.e.: Photocoagulation, Vasectomy. The target of our project is to give accurate and less time taking results by applying convolutional neural networks algorithms to classify different eyes infected by the illness.

In which Retinopathy (eye disease) is one of them. Diabetic retinopathy is one of most driving reasons behind blindness, there's been no treatment of diabetic retinopathy yet except for laser treatment which can be utilized to prevent sight loss in case if retina has not been harmed absolutely .so by the time human identify the results and the delayed results lead to delayed treatment. So there should be a model that can recognize the retinopathy with accurate outcomes[2].

Diabetic retinopathy is one of the main reasons of blindness. It's most common disease of eye. Almost 85% of patients who have suffered from this disease for more than 10 years could be affected by this. More than 10% of the patients with diabetic have suffered vision problems(3). Early detection of Diabetic retinopathy could prove helpful to reduce chances of vision loss. If they are properly treated

1.1 BACKGROUND AND MOTIVATION

As per survey among in Pakistan out of 16507 patient 95% of patient had diabetes and among them 15% patient were diagnosed with Diabetic retinopathy[4]. In Europe almost 60 million people are patient of diabetes and are mostly prone to the DR[5].In US 10.2 million people over the age of 40 have diabetes and among them 40% of people are on vision threatening disease DR[6].

Diabetic retinopathy is a standout amongst the most widely recognized Complication of diabetes. It can occur in individuals with both types 1 and type 2 diabetes, and affect both eyes. Individuals with retinopathy have damaged veins in their retina.

With the time when there is more than enough sugar in your blood there might be chance that there will be blockage of tiny blood vessels near retina , which cuts off the blood supply to eyes which causes harm of veins , which are at the back end of eye (retina), it can cause visual impairment at first it causes nothing or maybe minimal partial vision issues but in the end it can even cause complete blindness The condition can occur in any individual who has type 1 or type 2 diabetes[7].

TYPES

There are two types of DR.

1. **Non-Proliferative diabetic retinopathy**
    - This is early stage in which tiny blood vessels leak, making the retina swell.
    - Mild non-proliferative (This stages caused minimal swelling near or in the eye)
    - Moderate non-proliferative (This stage causes some of the vessel to be blocked)
    - Severe non-proliferative (This stage causes more vessels to be blocked resulting in the inadequate flow of blood to eye)

2. **Proliferative diabetic retinopathy**

This is the advanced stage in which retina starts growing new blood vessels. it might block all vision

- Proliferative diabetic retinopathy (The final stage cause some other abnormal blood vessels to grow near your retina , which can cause blood leakage)[8]

The symptoms of DR are different abnormal spots in sight, blurred sight, color loss vision, darkness in vision, vision loss[9].

DR is best diagnosed with the manual eye test, in which the examiner put drops in patient's eye which causes the eye pupil to broaden to examine better inside view of your eye. In which the examiner can see abnormalities in eye.[10]

Digital photography of retina is also another method for screening, in which photo of retina is take through camera and then later on observed by Doctor[11].

If only first two stages of DR is diagnosed, then there is no need for proper treatment only diabetes should be seen more properly. If only there is one of last two stages, then there is proper need for treatment and even can cause surgery depending upon condition.

Mainly it may be because of the diabetes which plays major role in having DR. It is cause by over time, too much sugar in blood. It connects you to your genes of DNA i.e. family genetic disease. It leads to blockage of the modest veins that feed the retina Accordingly eye endeavors to develop fresh recruit's vessels, yet these fresh recruit's vessels don't grow appropriately and can cause loss of blood vessels and leakage into the retina known as maculopathy and cause partial loss or blurry vision. Sometimes this abnormal vessel grows on the retina and blocks the vision result in complete loss of sight[12].

1.2 PROBLEM STATEMENT

In last decade, literature related to our research domain has been conducted using handcrafted features and using manually treatment. As technology is evolving, most of the researchers are now focusing on deep architecture-based solutions due to its success in other domains of computer science. In contact to this, deep learning architectures has its own limitations and don't show the better results in domain specific problems. A transfer learning approach has been approached in this contest.

1.3 RESEARCH QUESTIONS

RQ: Investigate how can Diabetic retinopathy is identified/Detected Using Deep Learning Techniques e.g. DCNN and Transfer learning?

RQ: To what extent proposed approach will perform for the prediction of diabetic retinopathy in human eye?

RQ: How this report will perform in research sector for Diabetic retinopathy prediction and to what extent it can be predicted?

1.4 RESEARCH OBJECTIVES

To address the research gaps and problems identified in problem statement, following objectives of this study are formulated.

Obj(a): A deep critical review of existing approaches discussed in Literature review of Diabetic retinopathy.

Obj(b): investigate use of transfer learning techniques for the prediction of Diabetic retinopathy using pre-trained transfer learning models based on proposed data set.

Obj(c): Comparison of the performance of Deep Learning Based model for the prediction of diabetic retinopathy of different level.

1.5 SCOPE

Our point is to show signs of improvement exact results and to propose a superior methodology than the past strategies for betterment of HealthCare domain with new Machine Learning techniques.

1.6 METHODOLOGY

Data is collected from kaggle and python visualization library were used to visualize our data. We use convolutional neural network and supervised learning architecture CNN is used to detect the image. We have used supervised learning in our project to train our model.

1.7 SIGNIFICANCE/ POTENTIAL APPLICATIONS

It's great if it's identified early in light of the fact that it very well may be analyzed in beginning times than its better to get treatment early and to fix it early. This can be executed in various clinics and hospitals with the goal that it might be valuable other than HealthCare domain it is going to helpful in IT sector for learning and analyzing the performance of the algorithms like CNN.

2. LITERATURE REVIEW ON DIABETIC RETINOPATHY (2004-2019)

The current detection of diabetic retinopathy is through dilated eye exam in which doctor place same eye drops into patient eyes which helps him to exams any damaged blood vessels and any other factors leading to cause of DR

Detection from blood vessels is one of the difficult steps in classification of different level of diabetic retinopathy. Another difficult step is feature extraction because it contains many branches of vessels so it is difficult to identify from Micro aneurysms and hemorrhages.

2.1 LITERATURE REVIEW ON DIABETIC RETINOPATHY USING DEEP LEARNING TECHNIQUES.

Rajiv Raman, Sangeetha Srinivasan, Sunny Virmani, Sobha Sivaprasad, Chetan Rao & Ramachandran Rajalakshmi in 2019 (8) used artificial intelligence techniques such as Convolutional neural network (Deep learning Method) for the detection of diabetic retinopathy. Diabetes is a common disease and millions of people are need to be screened. Deep learning offers great advantage in the screening of DR, improve identification from images and help in prevention of vision loss.

In 2017 Alexander Rakhlin Neuromation OU Tallinn, 10111 Estonia (9)used Deep learning Method for the identification of Diabetic retinopathy they used Deep Convolutional neural network which results revolutionized in the field of vision including medical imaging. For training their model they have used publicly available data from kaggle and they have achieved 99% accuracy with 71 % of sensitivity.

In [18] Varun Gulshan et al. apply deep learning to create an algorithm for automated detection of diabetic retinopathy and diabetic macular edema in retinal fundus photographs. They use the data set of EYEPACS.

According to [19], authors proposed deep learning approach for localizing discriminative features in DR. They Evaluate on Diaret DB1 Lesion Detection. This approach can be used to categorized into non-proliferative and proliferative diabetic retinopathy.

In [24] Rishab Gargeya] et al. apply deep feature learning for automated identification of Diabetic Retinopathy using deep learning, developed and evaluated a data-driven deep learning algorithm as a novel diagnostic tool for automated DR detection. The algorithm processed color fundus images and classified them as healthy (no retinopathy) or having DR, identifying relevant cases for medical referral.

Misgina Tsighe Hagos and Shri Kant (10) used transfer learning for an already trained DNN and it can reduce the cost of training. For this purpose, they have Inception modules for Diabetic Retinopathy detection. They have use the smaller version of the kaggle data and testedthemodel accuracy on unseen data.

In [20] Harry Prat et al. research about stages of DR, they use CNN model and this paper shown the five class problem for national screening of DR can be approached using CNN model. They use the data set of Kaggle.

Wan et al. [16] implemented transfer learning and hyper parameter tuning on the pre-trained models AlexNet, VggNet-s, VggNet-16, VggNet-19, GoogleNet and ResNet using the Kaggle dataset and compared their performances. The highest accuracy score was that of VggNet-s model, which reached 95.68% when training with hyper-parameter tuning [16]. In [17] Tomi Kouppi et al. use DIARETDB1 for detection of DR and evaluation using automatic methods of screening and monitoring of DR using Baseline method (used for distinguishing different findings) and performance measured using ROC (Receiving Operating curve) and WER (Weighted error rate) methods.

According to [21], authors used the small data set for transfer learning based detection of DR. Deep convolutional networks have been achieving high performance results on the Image Net Large Scale Visual Recognition Competition (ILSVRC) image classification challenge. This technique could be used in other deep learning based medical image classification problems facing the challenge of labeled training data insufficiency.

In [16]Nour Eldeen M. Khalifa .et.al uses Deep Transfer Learning models for the diabetic retinopathy prediction the models were trained on APTOS 2019 Data set. Many augmentation techniques were used for Models robustness according to literature survey the article is first to use APTOS data they perform many transfer learning models on this dataset such as AlexNet, ResNet18, SqueezeNet, GoogleNet, VGG16 and VGG19. Many evaluation techniques were used to analyze the best Model, Alex Model achieved 97% accuracy which is highest testing accuracy among the all proposed models.

Carson Lam, MD .et.al [17] in their research they uses Fundus images for the detection of diabetic retinopathy in human eye. They have proposed a deep convolutional neural network model for their proposed research and they have achieved 95% accuracy for their model. Other than their proposed model they have applied different transfer learning models on proposed dataset such as GoogleNet, AlexNet and ImageNet and their respective accuracies are 74.5%, 68.8% and 57.2% on test dataset respectively.

In [18] Misgina Tsighe Hagos .et.al have used annotated data as its one of difficult classification problem As deep convolutional neural networks DCNN have gain so much success in imageNet data classification they have shown very good performance in classification problems. They have applied different Transfer learning techniques for the competition but InceptionV3 shows maximum performance on training and test data for the detection of diabetic retinopathy the have applied different preprocessing techniques to extract features then data is trained on proposed model. The have used a small dataset of diabetic retinopathy from kaggle for this research to show InceptionV3 performance.

Sarfaraz Masood et.al. [19] used data provided by eyepacs which contains fundus images of different levels. As Deep convolutional neural networks gain much success in ImageNet they have used transfer learning technique using a pertained model for retraining it from start for another data. A Deep convolutional neural net classifier engineered from InceptionV3 Model that is trained on imageNet for classification of diabetic retinopathy of scale 0 – 5.

2.2 LITERATURE REVIEW ON DIABETIC RETINOPATHY USING MACHINE LEARNING TECHNIQUES.

Another way of detecting DR is examining through ophthalmoscopy DR was identified by ophthalmoscopy in 16% of 442 right eyes that had both ophthalmoscopy and any gradable photography, just as by photography in 8%. Image processing was additionally used to identify DR based on highlights, for example, veins, radiates, hemorrhages, and small scale aneurysms. Normal cases were found to be 90% accurate and NPDR were 87.5% accurate. SVM provide 95% and Bayesian provided 90% accuracy as average accuracy were 90%. Machine learning techniques were used in detection of DR, two classifiers were used to classify the DR. SVM provide 86% accuracy and KNN provided 55%.

In 2018 Maisha Maliha, Ahmed Tareque and Sourav Saha Roy (6)used machine learning algorithms for diabetic retinopathy detection. They use different techniques for feature extraction form images. They perform different machine learning algorithms for diabetic retinopathy detection, they used decision making algorithms for example Support vector mechanism, K-Nearest neighbor and Random forest.

In 2013 R. Priya1 and P. Aruna2 (7) used three models to diagnose diabetic retinopathy which are Bayesian Classification, Probabilistic Neural net and support vector mechanism and they compared

their performance. They use preprocessing techniques to extract features as it's easy to detect from features and after that they feed those feature to classification model. 350 fundus images were used. 250 were used for training and 100 for testing purpose.

Normal cases were found to be 90% accurate and NPDR were 87.5% accurate[22]. SVM provide 95% and Bayesian provided 90% accuracy as average accuracy were 90% . Machine learning techniques were used in detection of DR, two classifier were used to classify the DR. SVM provide 86% accuracy and KNN provided 55% [23].

R. Priya et.al in [21]they have proposed three different approaches to diagnose diabetic retinopathy such as Probabilistic neural networks PNN, Bayesian Classification and Support vector mechanism SVM. At the end proposed models performance is compared. As how much disease spread in the eye can be predicted from the retinal images by extracting feature using feature extraction techniques. They have used 350 fundus images with 100 images of training set and 250images for testing set. Result and comparison among the approaches shows that PNN has accuracy of 89.7%, Bayesian has 94.4% accuracy and SVM has the highest accuracy among all which is 97.6. they have tested the model performance on another dataset with 150 images provide by DiretDB data set and the performance of models is that PNN shows 87.6% accuracy, Bayesian shows 90.76 and SVM shows the highest accuracy of 95.38%. As it is clear from the performance comparison that SVM outperforms than all other models for both the datasets selected for this research report.

2.3LITERATURE REVIEW ON DIABETIC RETINOPATHY USING SEVERAL FEATURE TECHNIQUES

During the recent years, there have been many studies on automatic diagnosis of diabetic retinopathy using several features and techniques. In [16], authors proposed a method for automated detection and classification of vascular abnormalities in Diabetic Retinopathy using scale and orientation selective Gabor filter banks using Kaggle dataset with accuracy of 87%.

In 2019 Prabhjot Kaur, Somsirsa Chatterjee, Dilbag Singh (5)used classification algorithm, neural network to detect the diabetic portion in the image by applying proper segmentation and Preprocessing techniques. The model is implemented in MATLAB and results are analyzed in the paper.

In [17], authors presented a method to classify diabetic retinopathy subjects from changes in visual evoked potential spectral components. Detection of normal and other two main classes of diabetic retinopathy has been presented. In [13], this approach can be used to categorized into non-proliferative and proliferative diabetic retinopathy. According to [18], authors examine the reasons why a number of research groups have worked on developing methodology and computer software for automated image detection of retinal pathology.

In [24] Varun Gulshan et al. use kaggle data set (3500 images) and Messidor1(1200 images) for detection of DR and evaluation using automatic methods of screening and monitoring of DR using Baseline method (used for distinguishing different findings) and performance measured using ROC (Receiving Operating curve) and WER (Weighted error rate) methods.

According to [25], apply Kenneth algorithms for automated detection and classification of vascular abnormalities in DR. They use Canny's methods, used gradient operator to detect edges. They distinguish Mild NPDR from severe NPDR using Global image and Gabor filter bank.

In [26] RishabGargeya et al. apply feature learning for automated identification of DR, they developed and evaluated a data-driven deep learning algorithm as a novel diagnostic tool for automated identification of DR. The algorithm processed color fundus images and classified them as no DR or having DR.

2.4 LITERATURE REVIEW ON DIABETIC RETINOPATHY USING IMAGE PROCESSING TECHNIQUES

Past studies using various high bias, low variance digital image processing techniques have performed well at identifying one specific feature used in the detection of subtle disease such as the use of top-hat algorithm for micro aneurysm detection [14]. However, a variety of other features besides micro aneurysms are efficacious for disease detection. A convolutional neural network (CNN) was trained to classify a dataset of 128,175 fundus images into 2 classes, where the first class contains images with severity levels 0 and 1, and the second class contains levels 2, 3 and 4. In [15] Authors got sensitivity of 97.5% and specificity of 93.4% on the EyePACS-1 dataset which consists of 9963 images. That CNN scored a sensitivity of 96.1% and a specificity of 93.9% on the Messidor-2 dataset; and the sensitivity and specificity were 90.3% and 98.1% on the EyePACS-1, while 87% and 98.5% was scored on the Messidor-2, consecutively [15].

According to [23], apply Kenneth algorithms for automated detection and classification of vascular abnormalities in DR. They use Canny's methods, used gradient operator to detect edges. They distinguish Mild NPDR from severe NPDR using Global image and Gabor filter bank.

In [22] Varun Gulshan et al. use kaggle data set (3500 images) and Messidor1(1200 images) for detection of DR and evaluation using automatic methods of screening and monitoring of DR using Baseline method (used for distinguishing different findings) and performance measured using ROC (Receiving Operating curve) and WER (Weighted error rate) methods.

Wong Li Yun et.al in [25] they analyzed almost 124 retinal images of different stages and in results four categories of diabetic retinopathy are classified in human eye. They have used three layered architecture of feed forward neural network for the classification of these four level diabetic retinopathy. Before training the model on image samples/ training data, features are extracted from the raw images of diabetic eye using image processing techniques and the processed images are feuded to the classifier for classification. They have achieved 90-100% sensitivity and specificity.

In [26]Neera Singh et.al have deeply discussed the causes of diabetic retinopathy and its effects on eye and how can they be predicted by applying Image processing algorithms so it can be treated as early as possible. They have analyzed effected images of eyes and explained the certain features of retina's physiology that should be differentiated from abnormality pathology. Same as this blood vessels and fovea region of the retina should be extracted before diagnosing. They have applied grading retinal algorithm on the images which is international criteria for disease assessment. So the

effected human can be referred to a specialist as soon as possible and get receives a proper treatment accordingly.

Oliver Faust et.al [27] they have discussed different image processing algorithms for extracting features and automated diabetic retinopathy stage detection they have used digital image processing tool for screening that provide high quality and permanent record of retinal image. That can be further use for progression monitoring and accordingly treatment.

2.5 GAP ANALYSIS

- The existing approach of detecting DR is dependent of doctors. The basic method of dilated eye exams required attention and good eye vision of examiner. The bad eye vision or focus can lead to miss detection. They are certain tools available for helping examiner to look in the patient eyes such as ophthalmoscopy etc. but still the detection can rely from doctor to doctor based on their performance and experience. These approaches dependent on humans and there are several factors of human error which can lead to miss classification.
- Automatic detection techniques were used to eliminate human error and provide better detection. Digital Image Processing Technique were used for transformation of images to gather useful features. It applies different techniques like image augmentation, resizing, grey scaling, image reconstruction, smoothing, sharpening etc. This technique is useful but it requires a lot of feature extraction which can be difficult. It also requires to gather hand crafted features which can be time consuming.
- As we know that machine learning has recently caught focus in AI industry. As machine learning is useful in many domains its technique is also useful for detection or prediction. It naturally takes in and improve as a matter of fact without being unequivocally customized it learn through data and its technique is also used to detect the image whether it is effected or not but still this algorithm also required feature extraction on which classifier can classify and it's the main problem because we cannot decide which factor exactly lead to DR.

3. SCIENTIFIC METHODOLOGY APPROACH USED

3.1 PROPOSED METHODOLOGY

- As we have seen that there are certain approaches used to detect DR but mainly they were dependent on humans and were prone to human error. Our methodology of detection is to totally automate the detection by minimizing the human work. We have selected Deep Neural network approach to detect the DR. The main reason of choosing this approach is that for detection manually and other technique we have to extract hand crafted feature or have to use feature engineering. The main benefit of this Approach is that we don't have to worry about feature extraction because this technique requires no manual feature extraction. DNN algorithm is successful in providing the higher accuracy and it required minimal human labor. It requiresa huge amount of data through which its model learn and outperform no. of another technique by giving best performance It one of benefit is that this approach can work on raw data and don't require extra feature engineering and domain expert suggestions. Supervised learning is the part of DNN, which help models to learn from the previous

examples and records and can train the model efficiently. Efficiency of model also depends on the right weights and hyper parameters [11].

- The selection of right choice of data matters a lot. To extract a good training result, we need huge amount of data because more the data is more the accuracy is. The data gather from kaggle is sufficient for our DNN model. The DR images contain 5 classes. Our approach to classify the DR is the first we will classify binary class that whether image is affected or not because to detect the positive DR image is more important. It will allow us to first just detect whether the eye disaffected or not and this decision is more useful than detecting the stage if it predict affected image as not effected it can decrease the system efficiency. So first classification will be binary, after image is detected and the eye is affected we will move further where we will predict the among the stages of the DR.
- After gathering the data preprocessing is the first essential step. Data transformation is mostly used in pre-processing step because all the data is usually not normalizing and contain different representation and shapes and DNN require data should be in same shape and dimension to process the model efficiently. It transforms the data into same representation and format. Image usually gather are in high dimensional if it's reduce to low dimension because of higher computational cost. To enlarge the dataset data augmentation technique is used. Data augmentation contains Rotation, Flipping, Shearing, Rescaling transformations [56]. Gaussian transformation convert images from RGB format into one channel it. RGB channel contains 3 channel color representation which represent more features but by converting into Gaussianfiltered we convert images into low quality which removes the unnecessary features.
- This research uses conventional neural network architecture and transfer learning Models which is one of the best architecture of DNN. Most of the architecture fails to perform on raw data but conventional neural network has provided best performance on the raw data and extract useful features. The CNN receive data after preprocessing the data is split into 70:30 ratios (train: test). The model of CNN is defined in which no. of layers and filter is provided which process on images. Image is provided to architecture in which it extracts feature by using spatial filters and image dimension is reduced because it eliminates unnecessary feature and extract feature which are useful to classify the image this importance of each features is decided by weights in which we assign weights to our feature and use different technique until its optimized. Then weights backpropagation to reduce the loss of model. When all the feature is extracted the CNN classify the images on the basis that feature and provide accuracy of prediction.

3.2 DATA SET DESCRIPTION

Dataset consists of images eye retina scan images which are Gaussian filtered to identify diabetic retinopathy. APTOS 2019 Blindness Detection here is the data is freely available other than this dataset is also available at Kaggle as Diabetic Retinopathy 224 * 224 * 3 Gaussian filtered images. Images are resized into 224 * 224 shape to easily readily for different Machine Learning techniques. Images are saved in separate folders of each class according to stage of diabetic retinopathy. There are five folders of different stages of images.

- No_DR
- Mild
- Moderate
- Severe
- Proliferate_DR

Dataset consists of 3662 images.

TABLE 1: DATA SAMPLES OF EACH CLASS

NO DR	Mild	Moderate	Severe	Proliferate_DR
1805	370	999	193	295

Below is the sample image of dataset.

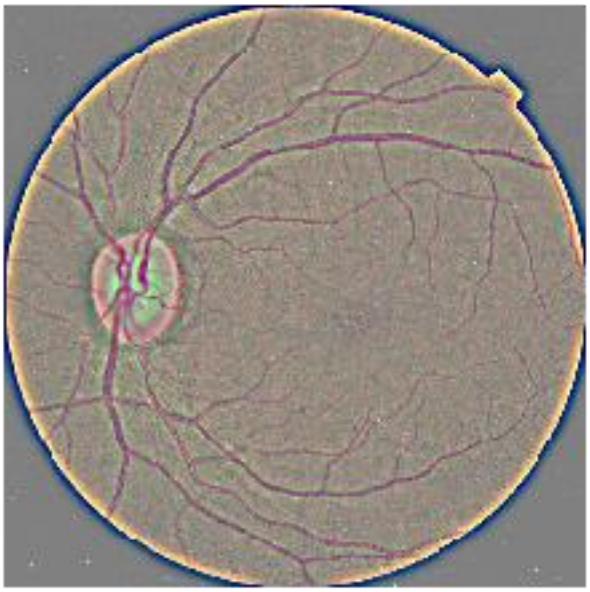

FIGURE 1: SAMPLE IMAGE FROM DATASET

Data Visualization by kaggle (Diabetic Retinopathy 224 * 224 * 3 Gaussian filtered).

id_code	diagnosis
Unique id of the images.	Level of diabetic retinopathy.
3662 unique values	
000c1434d8d7	2
001639a390f0	4
0024cdab0c1e	1
002c21358ce6	0
005b95c28852	0
0083ee8054ee	4
0097f532ac9f	0
00a8624548a9	2
00b74780d31d	2
00cb6555d108	1
00cc2b75cddd	0

FIGURE 2: DATA LEVELS WITH UNIQUE ID'S

3.3 WORKFLOW OF THE SYSTEM

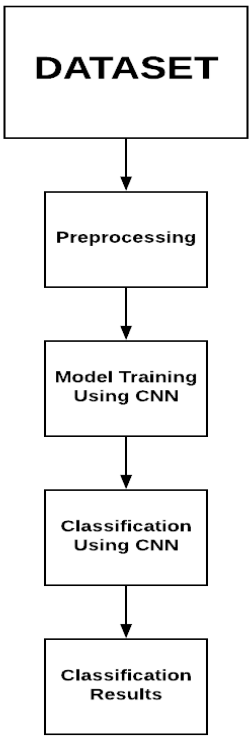

FIGURE 3: SYSTEM WORK FLOW DIAGRAM

3.4 ALGORITHMS/ARCHITECTURE

CNN is a type of neural network, it contains many hidden layer and each hidden layer contains several neurons and all neurons are connected to each other. Certain weights are given to model which are optimized by time and our model. All the hidden layers are connected with each other which form as full connected layer and is responsible for output. In our model first we provide our dataset to our model then use our conventional layer which contains set of neurons and weights which provides image features. The conv layer consists of filter size which decides how much features we want to extract. We reduced our data until it give us maximum sharp feature then we apple dense layer which is responsible for classify. As dimensions of our dataset were reduced to 128*128 and CNN model were implemented accordingly as mentioned in figure.

Layer (type)	Output Shape	Param #
conv2d_49 (Conv2D)	(None, 124, 124, 16)	1216
conv2d_50 (Conv2D)	(None, 120, 120, 32)	12832
max_pooling2d_18 (MaxPooling	(None, 40, 40, 32)	0
conv2d_51 (Conv2D)	(None, 36, 36, 64)	51264
max_pooling2d_19 (MaxPooling	(None, 12, 12, 64)	0
conv2d_52 (Conv2D)	(None, 10, 10, 128)	73856
flatten_13 (Flatten)	(None, 12800)	0
dropout_25 (Dropout)	(None, 12800)	0
dense_40 (Dense)	(None, 64)	819264
dropout_26 (Dropout)	(None, 64)	0
dense_41 (Dense)	(None, 64)	4160
dropout_27 (Dropout)	(None, 64)	0
dense_42 (Dense)	(None, 2)	130

Regional Conventional Neural Network (R-CNN)

Usually in other architecture we process on every feature of image and then decide it is important or not but in Rr-CNN we only process on the area of image which is of our interest. It can reduce the number of process that is applied on every feature of image. In output it provides bounding boxes and labels for each object in the image. It only focuses on the region of interest. In DR detection it can only focus on the region which specifies the main feature of image. The region of interest is resized and then CNN perform only on

the resized region then it is more refined by using SVM. R-CNN used pre-trained network like Alexnet,Vgnet etc. then last dense layers are added according to dataset classes.[12]

Transfer learning

Transfer learning is a model which is used for reusability so we do not have to write our model our train our weights from scratch. In this pre-trained models are used on our datasets so we do not have to train the weights again which can be help if our computational complexity is too high. We can use transfer learning when we know that dataset is similar to the previous model dataset which is already trained. But in pre-trained model we cannot change the layer or any specific parameter. We can also use transfer learning when we don't have enough data to train on our model. It can be also used to extract features from the image.Some of Transfer Learning techniques used for Prediction of Diabetic retinopathy for this report are discussed below:

- **VGG Models**

Vgg16 is a convolutional neural network architecture introduced by Oxford university. It is one of the best vision recognition architecture. Instead of large number parameters vgg architecture is developed having focused on 3*3 convolutional layers with one strides with same padding and 2*2 max pool. The whole architecture is based on convolutional layers and max pool layers at the end of the architecture there is two fully connected layer with softmax activation. By default, its weights are pre trained and can be used for retraining as starting point for another dataset.

- **Res Net**

Unlike other models (VGG, Alexnet) resNet is a form of exotic architecture which basically relies on Micro architecture modules know as network to network architecture. Firstly introduced in 2015 [28] by He et.al. it is extremely deep network along with convolutional layers and pool. ResNet50 is basically 50 deep layers' implementation in keras. ResNet is deeper than other models such as vgg16 and Vgg19 but the size of the model is less due to global average pooling.

- **Inception Model**

The main goal of inception is to extract feature by computing 1*1, 3*3 and 5*5 convolutions with same module network and the output of this layer is stacked along and feuded to the next layer. The original incarnation of this network is google Net but subsequent manifestation is known as Inception vN where N is number of the version introduced by google. The weights for Inception V3 are smaller than both VGG and ResNet, coming in at 96MB.

- **Xception Model**

Xception was proposed by as a matter of fact François Chollet himself, the creator and head maintainer of the Keras library. Xception is an expansion of the Inception structure which replaces the standard Inception modules with depthwise distinguishable convolutions. Xception sports the littlest weight serialization at just 91MB.

Max pooling:

Pooling layer is used in between hidden layer of Conventional neural network. It reduces the size of images without losing useful information and reduces computation in network. It also help in reducing over fitting. It depend on which layer you want to use it.It usually consists of filter size which tell how much you want to reduce features e.g. 2*2, 3*3. When matrix of image is gathered it takes maximum value of filters. Maxpool is not always same it is dependent on our model and our dataset. Excessive maxpooling will result in loss of useful features

Loss Functions:

Loss function is simply a method to check that our model is good enough or not loss function gives the distance between the ground truth value and the predicted value. There are many loss function, it depend on problem which we want to use some are as below

- **Mean squared error:** Mean squared error is the simplest error which just gives us the difference of squared distance between the actual value and the predicted value and the category with the high value shoots up because of the square.
- **Mean absolute error:** Mean absolute is just like the mean squared error but it doesn't take square but gives the simple difference between the ground truth values and the predicted value.
- **Hinge loss/ SVM multiclass loss:** this loss functions says that the sum of all wrong predicted values must be smaller than the predicted value which is true. It is a convex function and mostly use by the convex optimizers.
- **Cross entropy loss:** cross entropy loss function value increases when the probability of the wrong predicted labels goes high and it decrease when the score of predicted label is high Binary**- cross-entropy loss function:** this loss function takes the difference between the predicted probabilities between the two labels.
- **Categorical cross-entropy loss function:** Neural net have 1 neural for each of the possible labels which can be predicted by the model so its gives us the probabilities for each label and the neural with the highest probability will activate and pass its label

Optimizers

Optimizers are use in deep learning or in machine learning to update the weights of neural network weights are basically the learning of neural network so we need a function that learn rates fast and correct according to our training data so that we can use it later on new examples. Optimizers I use in my code are adam, adagrad, momentum,batch gradient descent, mini-batch gradient descent and sgd.

- **Adam:** Adam is updated version of stochastic gradient descent and has the qualities to RMSprop and adagrad. Adam is efficient in giving good results as compared to other

optimizers and takes less time to achieve goal. Adam work good for noisy data. Adam parameters

- Alpha : Alpha is the initial learning rate of the optimizer or the step size
- Beta1: Beta1 is the decaying rate of first moment estimation
- Beta2: Beta2 is the decaying rate of second moment estimation
- Epsilon: Epsilon are the small values used for getting rid of dividing by zero problems.

- **Batch Gradient descent:** Batch gradient descent is also known as the vanilla gradient descent it calculates the error of every example in the training set but after an epoch and then update the weights with respect to the error. One of the advantage of the batch gradient descent is it is computational effective and generate stable results. Buts it is not the best that a model achieves. It is also not suitable to work on large models or large dataset because it need to have all the training data to be load into the memory.

- **SGD:** Stochastic gradient descent work just like batch gradient descent but it updates the parameter after every training example not after an epoch so we can get better results faster. But fast change in updates are computationally expensive and because of rapid change in parameter it may go towards the noisy gradients.

- **Mini batch gradient descent:** Mini batch gradient descent have the power of both sgd and bgd because it makes mini batches in the training examples and update the parameters after every batch which helps Mbgd to maintain efficiency of BGD and help to get better results fast like SGD Helping material.

- **Momentum:** Momentum is a method that helps sgd to accelerate in the relevant direction as sgd.It's difficult for sgd to navigate good path or steep paths As steep ways which are closer to local minima so when sgd go towards steep paths it oscillates to much in the areas of local minima. So momentum is introduced to help sgd to follow relevant path by adding a factor R. Momentum helps sgd to get stable when there are so many local minima.

- **AdaGrad:** Adagrad is really useful when you have spare data because adagrad adapts low learning rate when there are frequent features in the data and on the other side adagrade adapts high learning rates when it comes to the infrequent data. Because of this robust property adagrade is used to train large neural nets

Risk Analysis

Proposed methodology is decided after going through many research work related to diabetic retinopathy but in case if the model shows less accuracy and efficiency we will use different techniques for feature extraction and use better methodology for this. For increasing the efficiency of the model we will train our model further on more training data to increase max accuracy and efficiency to predict Diabetic retinopathy.

4. IMPLEMENTATION, EVALUATION AND RESULTS OF AUDIO SPEECH EMOTIONS DETECTION MODELS

4.1 SYSTEM DESIGN

Performance

Performances depend upon the processing power of system and selection of suitable algorithm. On CPU processing Performance will be low because of computational power but by using GPU we can boost its Performance easily as GPU provide high computational power than CPU with efficient algorithm discussed in proposed methodology.

Robustness

It is not much robust because of architecture dependencies with system.

Interactively

Only specified and trained/expert users can access the system.

Flexibility

It can detect all kind of image whether it's on low quality or higher. It can also detect Image of different dimensions which provide flexibility to system.

Re-usability and portability

After training our model we can reuse our model to train on new data and it can be deployed on any server system and any operating system.

Security

The data of patients are private and its should not be accessible to anyone.

Functional Requirements

TABLE 2: FUNCTIONAL REQUIRMENTS

Requirement Name	Description
Early Detection	Early detection of diabetic retinopathy
Eyes Retina	Main function of eye in which disease produce
Diagnose	Diagnose the disease in early stage
Medication	Help Medical science to improve their equipment

Non - Functional Requirements

TABLE 3: NON FUNCTIONAL REQUIRMENTS

Requirement Name	Description
Early Detection	Early detection of diabetic retinopathy
Eyes Retina	Main function of eye in which disease produce
Diagnose	Diagnose the disease in early stage
Medication	Help Medical science to improve their equipment

As our project is research base so we have done different experiment with different algorithms on same datasets. And we will compare results of different algorithms against deep learning model and features.

Experimentation

Deep learning models have different hyper parameters that are tuned to get good outcomes. Whereas different libraries have been used and the environment used for this experimentation was cloud services of kaggle.

Data was preprocessed due to limited resources and also different architectures also been used in this experimentation kaggle services were used but also the experimentation was done on linux operating system.

Experimental Setup

We used the environment of kaggle. The experimentation was done on python 3.5, which was managed by kaggle keras library was used as front end whereas tensor flow was used as back end. the services that kaggle GPU has come up with is faster that 70% from CPU which is approximately equal to 960TI GeForce Nvidia Hyper parameters optimization techniques were used through randomized search, grid search and talos library.

The data was Gaussian filtered images and it was given to the model in patches because of less resources. The data was of 500+MB and the total no. of images were 3662. the dimensions for these images were 224*224*3

Pill library was used for image processing which reduces the dimensions of image. For good results tensor flow library was used which produces good outcomes.

Experimental Design/Details

Some of the images were taken and were visualized. the dimensions of the images were 3000*4000*3which were resized to the new dimensions i.e. 224*224*3, to get only the main features we want to use and to discard the other extra features.

Data frames of pandas were used to separate every category in different folders, to use it through fit generator which uses small patches at runtime for better RAM consumptions either way it can't be

use directly because the RAM resources were limited for this kind of large data fit generator converts images into numpy array and after that it feeds the model in this experimentation gradient descent based learning is used which is highly optimized and produces good outcomes.

4.2 IMPLEMENTATION,EVALUATION AND RESULTS OF BASIC MACHINE LEARNING TECHNIQUES IMPLEMENTATIONS

or this implementation a Machine Learning algorithm decision tree, Random Forest and Support vector mechanism SVM are implemented on proposed Data with 0.3 test data splitting and model is loaded using sklearn Machine learning library. After splitting the data into training and test data the training data is feuded to the Machine Learning Model and tested on test data.

4.2.1 DECISION TREE RESULT

Figure 4 below shows the performance of Decision tree on proposed data and the model achieved the accuracy of 64% on training.

```
Classification report for -
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best'):
              precision    recall  f1-score   support

           0       0.21      0.29      0.24        89
           1       0.56      0.56      0.56       314
           2       0.90      0.88      0.89       539
           3       0.21      0.19      0.20        88
           4       0.14      0.10      0.12        69

    accuracy                           0.64      1099
   macro avg       0.40      0.41      0.40      1099
weighted avg       0.64      0.64      0.64      1099
```

FIGURE 4: DESISION TREE PERFORMANCE

4.2.2 RANDOM FOREST RESULT

Figure 5 below shows the performance of Random forest on proposed data and the model achieved the accuracy of 72% on training.

```
Classification report for -
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=2, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10,
                       n_jobs=None, oob_score=False, random_state=0, verbose=0,
                       warm_start=False):
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        89
           1       0.55      0.85      0.67       314
           2       0.85      0.96      0.90       539
           3       0.00      0.00      0.00        88
           4       0.56      0.07      0.13        69

    accuracy                           0.72      1099
   macro avg       0.39      0.38      0.34      1099
weighted avg       0.61      0.72      0.64      1099
```

FIGURE 5: RANDOM FOREST MODEL PERFORMANCE

4.2.3 SVM RESULT

Figure 6 below shows the performance of Random forest on proposed data and the model achieved the accuracy of 71% on training.

```
Classification report for -
GridSearchCV(cv='warn', error_score='raise-deprecating',
             estimator=SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
                           decision_function_shape='ovr', degree=3,
                           gamma='auto_deprecated', kernel='rbf', max_iter=-1,
                           probability=False, random_state=None, shrinking=True,
                           tol=0.001, verbose=False),
             iid='warn', n_jobs=None,
             param_grid=[{'C': [1, 10, 100, 1000], 'kernel': ['linear']},
                         {'C': [1, 10, 100, 1000], 'gamma': [0.001, 0.0001],
                          'kernel': ['rbf']}],
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring=None, verbose=0):
              precision    recall  f1-score   support

           0       0.27      0.56      0.36        89
           1       0.64      0.66      0.65       314
           2       0.90      0.97      0.93       539
           3       0.50      0.01      0.02        88
           4       0.56      0.07      0.13        69

    accuracy                           0.71      1099
   macro avg       0.57      0.45      0.42      1099
weighted avg       0.72      0.71      0.68      1099
```

FIGURE 6: SVM MODEL PERFORMANCE

4.3 IMPLEMENTATION,EVALUATION AND RESULTS OF SIMPLE CNNSYSTEM IMPLEMENTATION

System Implementation

Develops tools and environment used

- Python 2.7/3.6

- Theano
- TensorFlow 1.5
- Keras
- Scikit-learn
- NVIDIA CUDA Toolkit 9.0
- NVIDIA cuDNN 7.0.5
- Anaconda Distribution
- Jupyter Notebook
- PIL

Evaluation and Results

There were total 20% black images in the data with no knowledge(outliers). To discard them from the data we built CNN model. Total 8 convolutional layers were used. 4 maxpool layers were used.3 Dense layers were used. Loss was binary cross entropy. Optimizer was of 'adam'. Training accuracy is 89% and Validation accuracy is 84%.

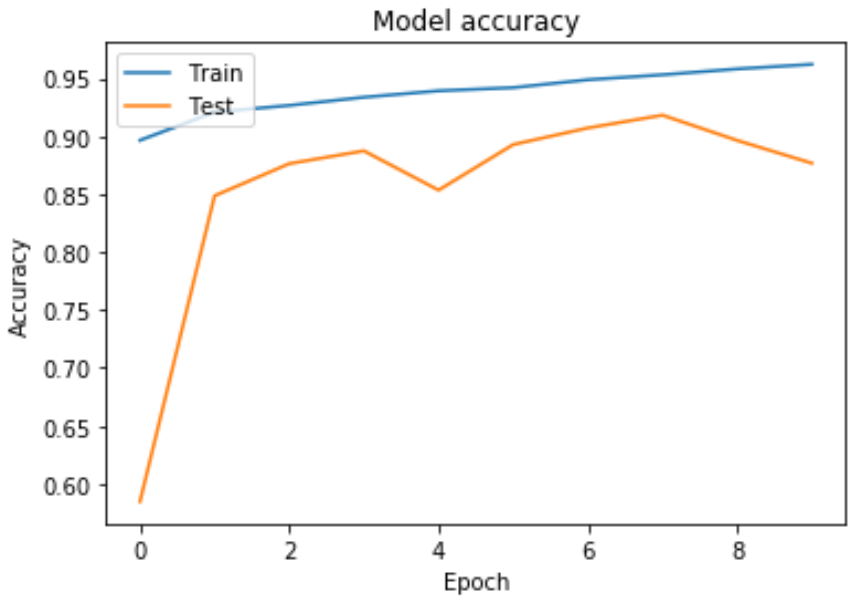

FIGURE 7: CNN MODEL ACCURACY PLOT

Binary Classification was done which differentiate between positives and negatives, then after that categorical Classification was done. Here in figure blue solid represents the training accuracy of the model on the other hand yellow solid line presents the testing accuracy of the model along with epochs on x-axis and accuracy on y-axis.

4.4 IMPLEMENTATION,EVALUATION AND RESULTS OF TRANSFER LEARNING USING VGG19

System Implementation

Develops tools and environment used

- Python 2.7/3.6
- Theano
- TensorFlow 1.5
- Keras
- Scikit-learn
- Anaconda Distribution
- Jupyter Notebook
- PIL

Evaluation and Results

For this experiments VGG19 Model is downloaded from Github using

vgg = VGG19(input_shape=[224,224,3], weights='imagenet', include_top=False)

and a Dense Layer is added to the model with softmax activation.

VGG16 Model summary.

```
dense_1 (Dense)              (None, 5)                 125445
=================================
Total params: 20,149,829
Trainable params: 125,445
Non-trainable params: 20,024,384
_________________________________
```

After Training the model on Data We have got 30% training Accuracy and 29% Validation Accuracy as the training accuracy is 30% which is very bad as model is not fit for selected dataset.

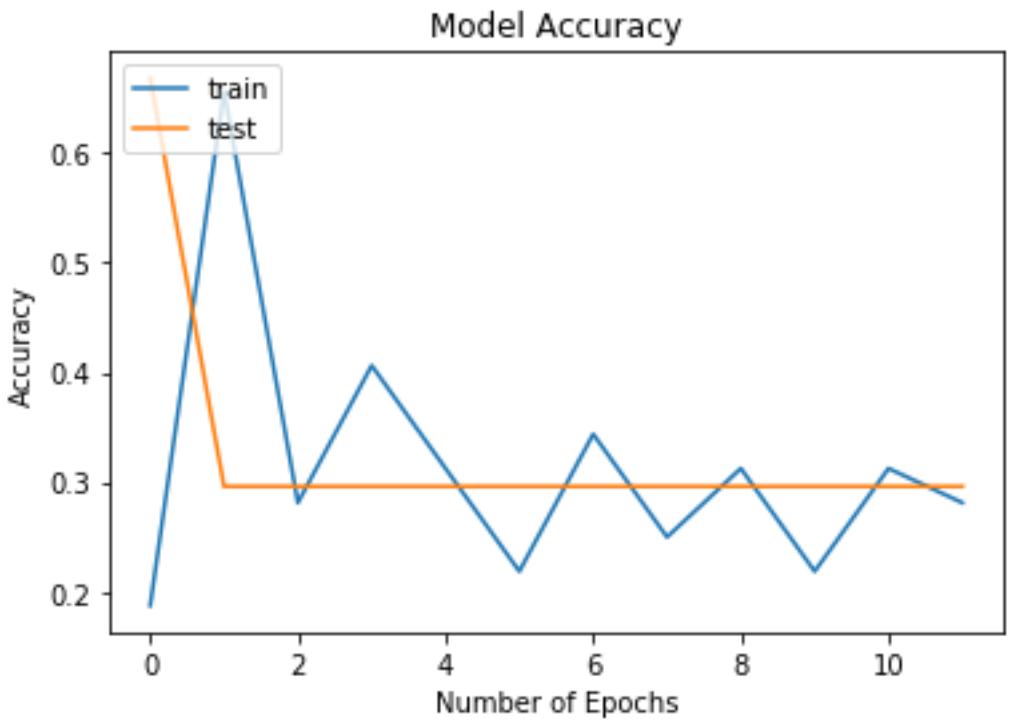

FIGURE 8: VGG19 MODEL ACCURACY PLOT

Categorical Classification is done with VGG19 Model and the accuracy is very bad and with 30% of validation accuracy model is consider to show less promising results as compare to other models. Here in figure blue solid represents the training accuracy of the model on the other hand yellow solid line presents the testing accuracy of the model along with epochs on x-axis and accuracy on y-axis.

Below confusion matrix shows the accuracy and loss of the classifiers on test data

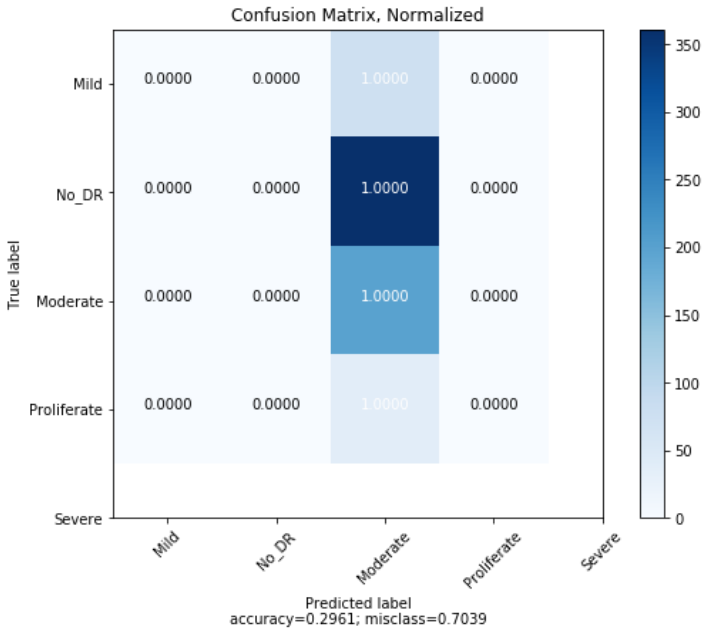

FIGURE 9: VGG19 MODEL CONFUSION MATRIX

4.5 IMPLEMENTATION,EVALUATION AND RESULTS OF TRANSFER LEARNING USING VGG16

System Implementation

Develops tools and environment used

- Python 2.7/3.6
- Theano
- TensorFlow 1.5
- Keras
- Scikit-learn
- Anaconda Distribution
- Jupyter Notebook
- PIL

Evaluation and Results

For this experiments VGG16 Model is downloaded from Github using

vgg = VGG16(input_shape=[224,224,3], weights='imagenet', include_top=False)

and a Dense Layer is added to the model with softmax activation.

VGG16 Model summary

```
Model: "model_1"
_________________________________
```

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 224, 224, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 25088)             0
_________________________________________________________________
dense_1 (Dense)              (None, 5)                 125445
=================================
Total params: 14,840,133
Trainable params: 125,445
Non-trainable params: 14,714,688
_________________________________
```

After Training the model on Data We have got 90% training Accuracy and 74% Validation Accuracy as the training accuracy is 90% which is good as model is not over trained.

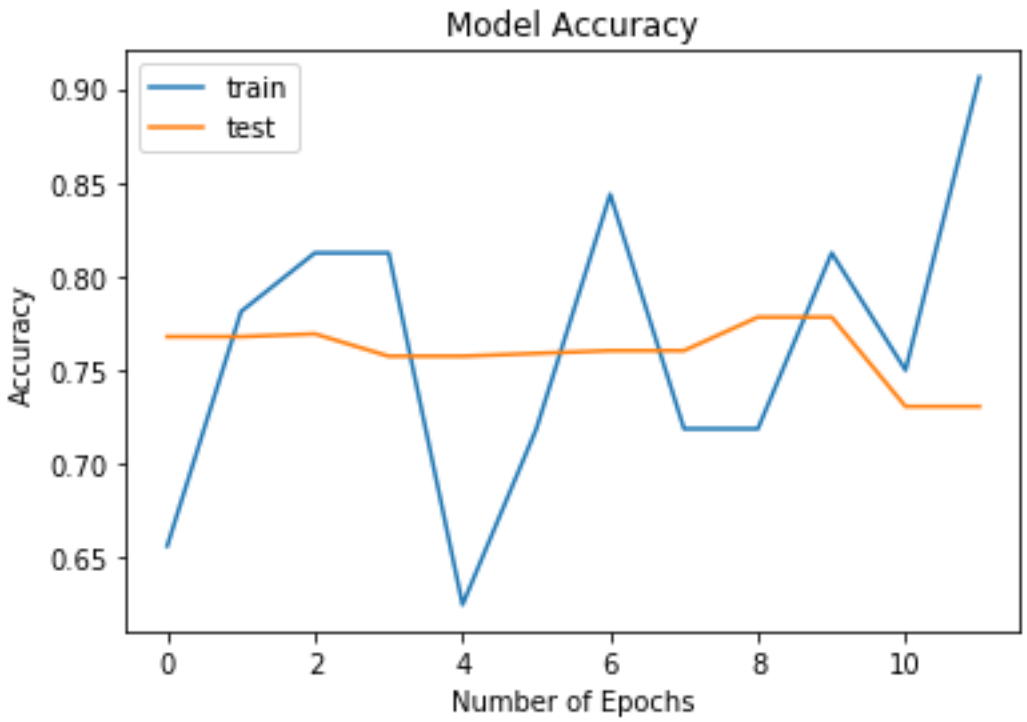

FIGURE 10: VGG16 MODEL ACCURACY PLOT

Categorical Classification is done with VGG16 Model and the accuracy is good and with 90% of validation accuracy model is consider to show more promising results as compare to other models.Here in figure blue solid represents the training accuracy of the model on the other hand yellow solid line presents the testing accuracy of the model along with epochs on x-axis and accuracy on y-axis.

Below confusion matrix shows the accuracy and loss of the classifiers on test data

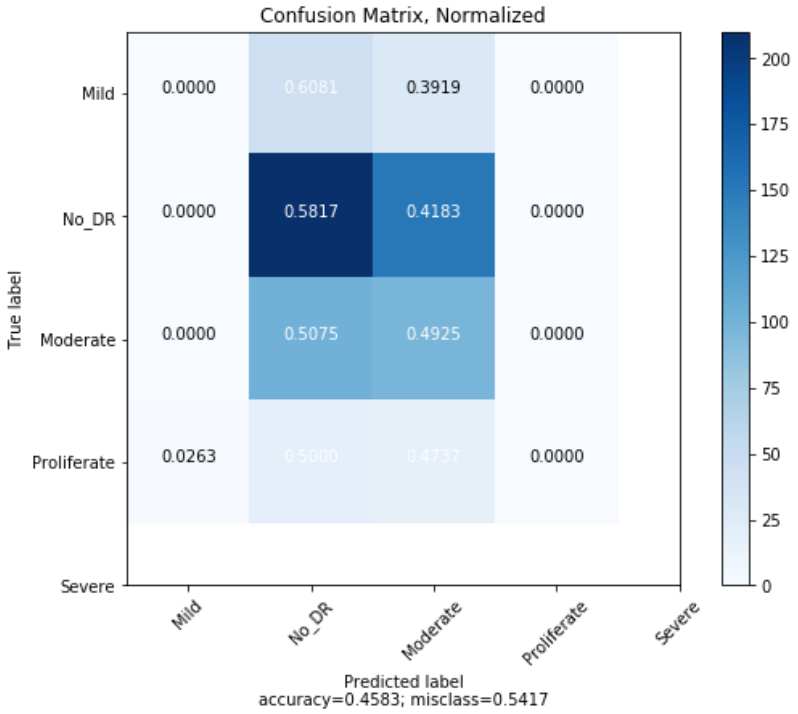

FIGURE 11: VGG16 MODEL CONFUSION MATRIX

4.6 IMPLEMENTATION,EVALUATION AND RESULTS OF TRANSFER LEARNING USING RESNET50

System Implementation

Develops tools and environment used

- Python 2.7/3.6
- Theano
- TensorFlow 1.5
- Keras
- Scikit-learn
- Anaconda Distribution
- Jupyter Notebook
- PIL

Evaluation and Results

For this experiments ResNet50 Model is downloaded from Github using

Resnet = ResNet50 (input_shape=[224,224,3], weights='imagenet', include_top=False)

and a Dense Layer is added to the model with softmax activation.

ResNet50 Model summay.

```
Model: "model_1"
dense_1 (Dense)                 (None, 5)             501765      flatten_1[0]
[0]
==================================
Total params: 24,089,477
Trainable params: 501,765
Non-trainable params: 23,587,712
__________________________________
```

After Training the model on Data We have got 78% training Accuracy and 61% Validation Accuracy as the training accuracy is 78% which is good as model is not over trained.

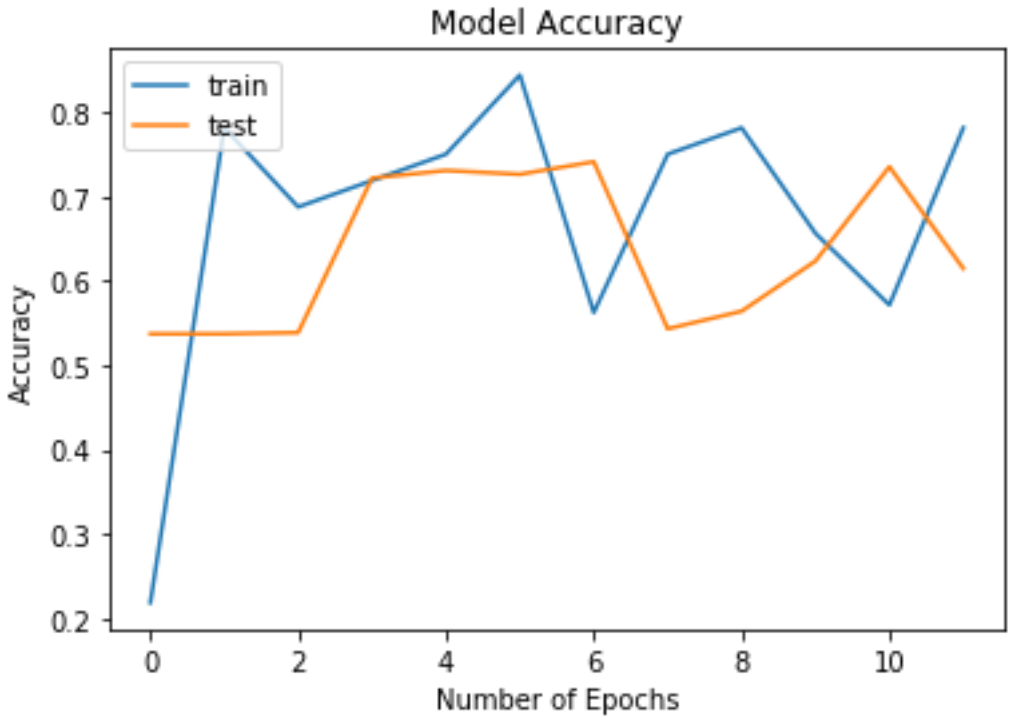

FIGURE 12: RESNET MODLE ACCURACY PLOT

Categorical Classification is done with ResNet50 Model and the accuracy is good and with 78% of validation accuracy model is consider to show more promising results. Here in figure blue solid represents the training accuracy of the model on the other hand yellow solid line presents the testing accuracy of the model along with epochs on x-axis and accuracy on y-axis. Below confusion matrix shows the accuracy and loss of the classifiers on test data

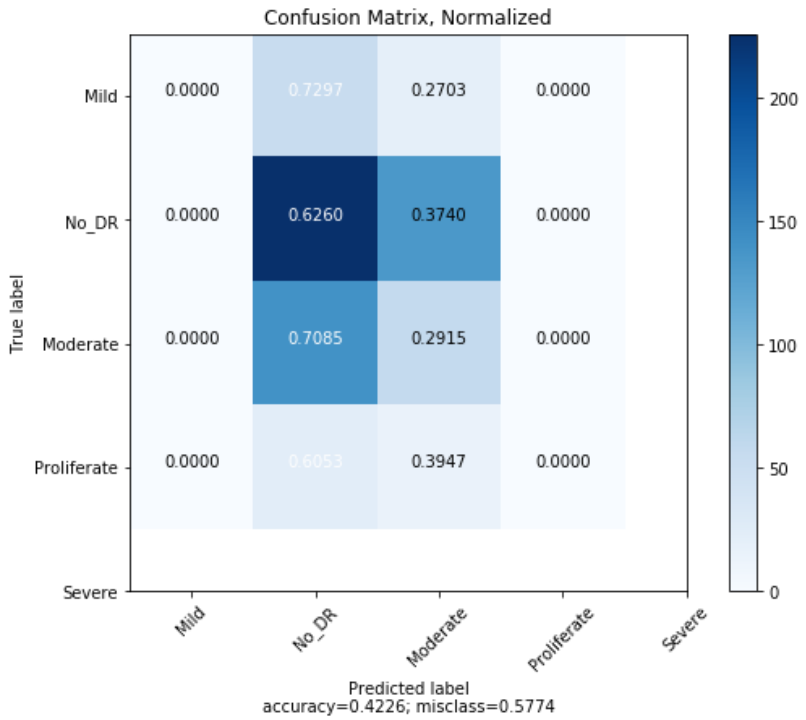

FIGURE 13: RESNET50 MODEL CONFUSION MATRIX

4.7 IMPLEMENTATION,EVALUATION AND RESULTS OF TRANSFER LEARNING USING XCEPTION MODEL

System Implementation

Develops tools and environment used

- Python 2.7/3.6
- Theano
- TensorFlow 1.5
- Keras
- Scikit-learn
- Anaconda Distribution
- Jupyter Notebook
- PIL

Evaluation and Results

For this experiments Xception Model is downloaded from Github using

V3 = Xception (input_shape=[224,224,3], weights='imagenet', include_top=False)

and a Dense Layer is added to the model with softmax activation.

Xception Model Summary.

```
dense_1 (Dense)                 (None, 5)            501765      flatten_1[0]
[0]
==================================
Total params: 21,363,245
Trainable params: 501,765
Non-trainable params: 20,861,480
__________________________________
```

After Training the model on Data We have got 68% training Accuracy and 50% Validation Accuracy as the training accuracy is 68% which is good as model is not over trained.

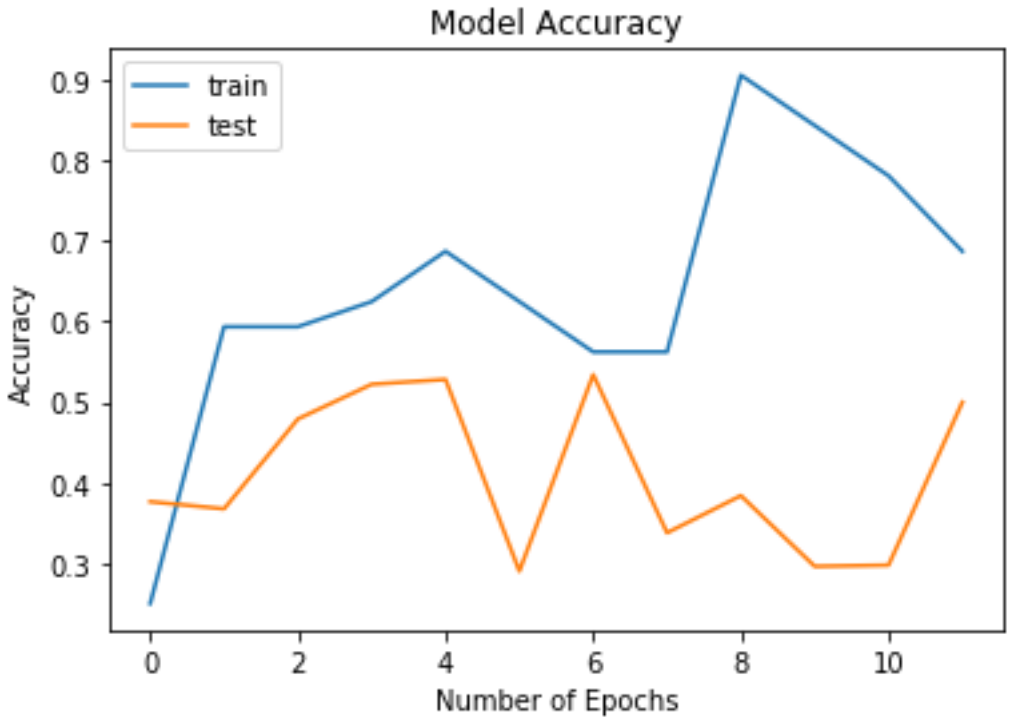

FIGURE 14: XCEPTION MODEL ACCURACY PLOT

Categorical Classification is done with Xception Model and the accuracy is not bad and with 50% of validation accuracy model can be consider to show good results. Here in figure blue solid represents the training accuracy of the model on the other hand yellow solid line presents the testing accuracy of the model along with epochs on x-axis and accuracy on y-axis.

Below confusion matrix shows the accuracy and loss of the classifiers on test data

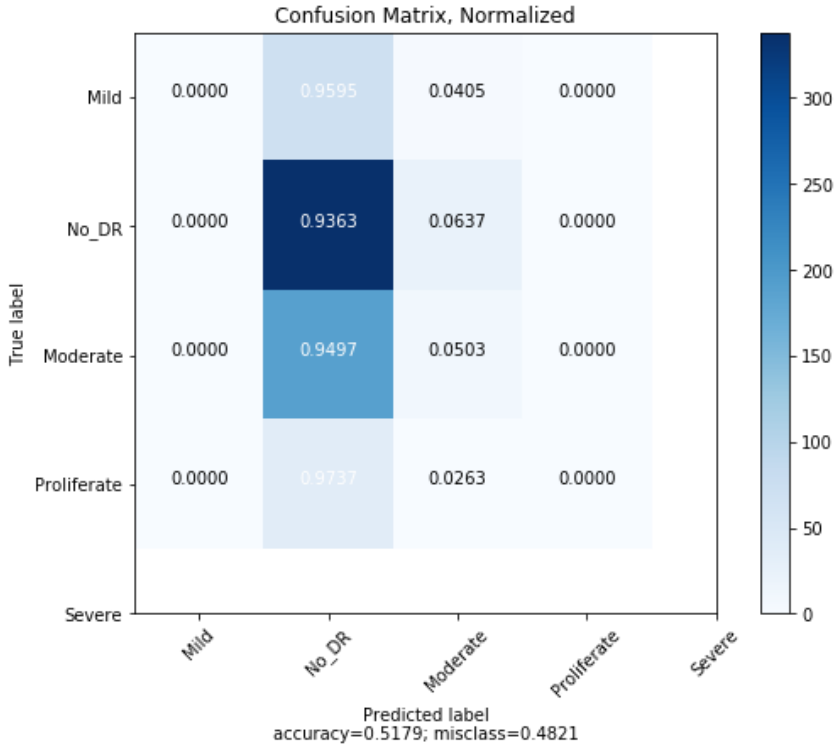

FIGURE 15: XCEPTION MODEL CONFUSION MATRIX

4.8 IMPLEMENTATION,EVALUATION AND RESULTS OF TRANSFER LEARNING USING INCEPTIONV3

System Implementation

Develops tools and environment used

- Python 2.7/3.6
- Theano
- TensorFlow 1.5
- Keras
- Scikit-learn
- Anaconda Distribution
- Jupyter Notebook
- PIL

Evaluation and Results

For this experiments InceptionV3 Model is downloaded from Github using

V3 = InceptionV3 (input_shape=[224,224,3], weights='imagenet', include_top=False)

and a Dense Layer is added to the model with softmax activation.

InceptionV3 Model Summary.

```
dense_1 (Dense)                 (None, 5)            256005      flatten_1[0]
[0]
==================================
Total params: 22,058,789
Trainable params: 256,005
Non-trainable params: 21,802,784
```

After Training the model on Data We have got 78% training Accuracy and 62% Validation Accuracy as the training accuracy is 62% which is good as model is not over trained.

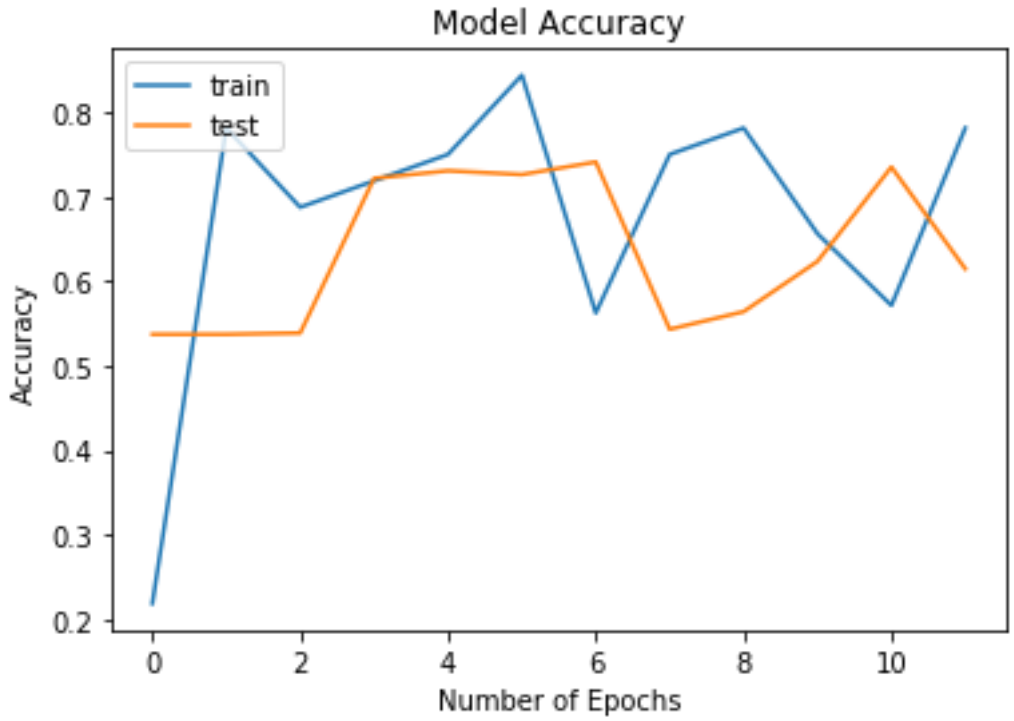

FIGURE 16: INCEPTIONV3 MODEL ACCURACY PLOT

Categorical Classification is done with InceptionV3 Model and the accuracy is not bad and with 53% of validation accuracy model is consider to show good results. Here in figure blue solid represents the training accuracy of the model on the other hand yellow solid line presents the testing accuracy of the model along with epochs on x-axis and accuracy on y-axis.

Below confusion matrix shows the accuracy and loss of the classifiers on test data

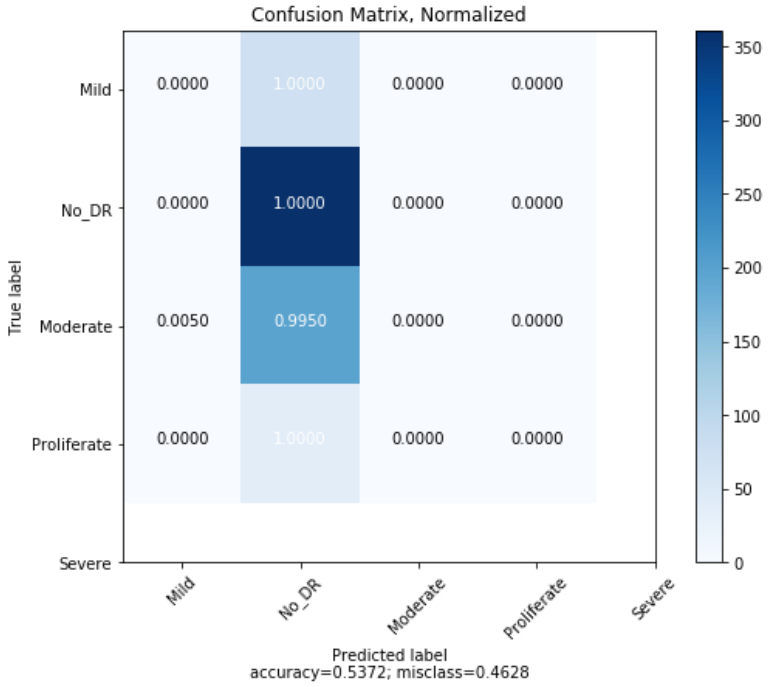

FIGURE 17: INCEPTIONV3 MODEL CONFUSION MATRIX

4.9 IMPLEMENTATION,EVALUATION AND RESULTS OF TRANSFER LEARNING USING VGG8

System Implementation

Develops tools and environment used

- Python 2.7/3.6
- Theano
- TensorFlow 1.5
- Keras
- Scikit-learn
- NVIDIA CUDA Toolkit 9.0
- NVIDIA cuDNN 7.0.5
- Anaconda Distribution
- Jupyter Notebook
- PIL

Evaluation and Results

For this experiment the VGG8 Model is created from scratch. First image Pixals normalization for quick and accurate training by dividing images with 255 than Decreases the resolution of the images.

Splitting the data into traingset and testset for model training 20% test data remaing 80% is consist of 20% validation and rest for training.

Data shape

(2343, 64, 64, 3)

(586, 64, 64, 3)

(733, 64, 64, 3)

Model Creation

Three Different models are created using different hyper parameters so the best model with best hyper parameters is selected. And train it on selected data. Training and validation accuracy of these three models is shown below

Model 1

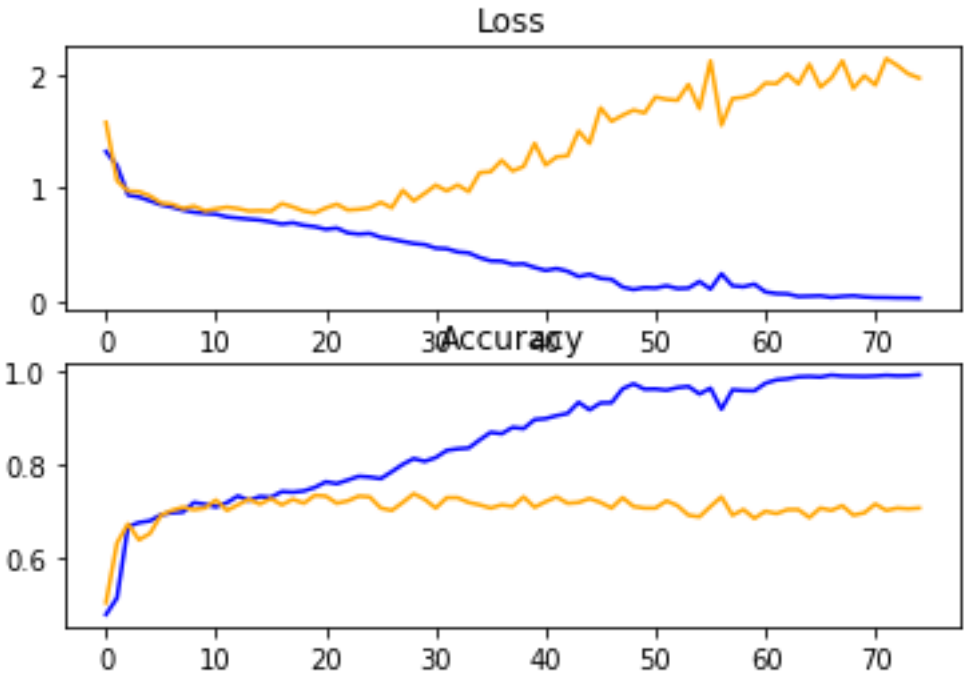

FIGURE 18: VGG8 MODEL 1 ACCURACY PLOT

Here blue solid line presents the training performance and on the other hand yellow solid line shows the testing performance of the model.

Model 2

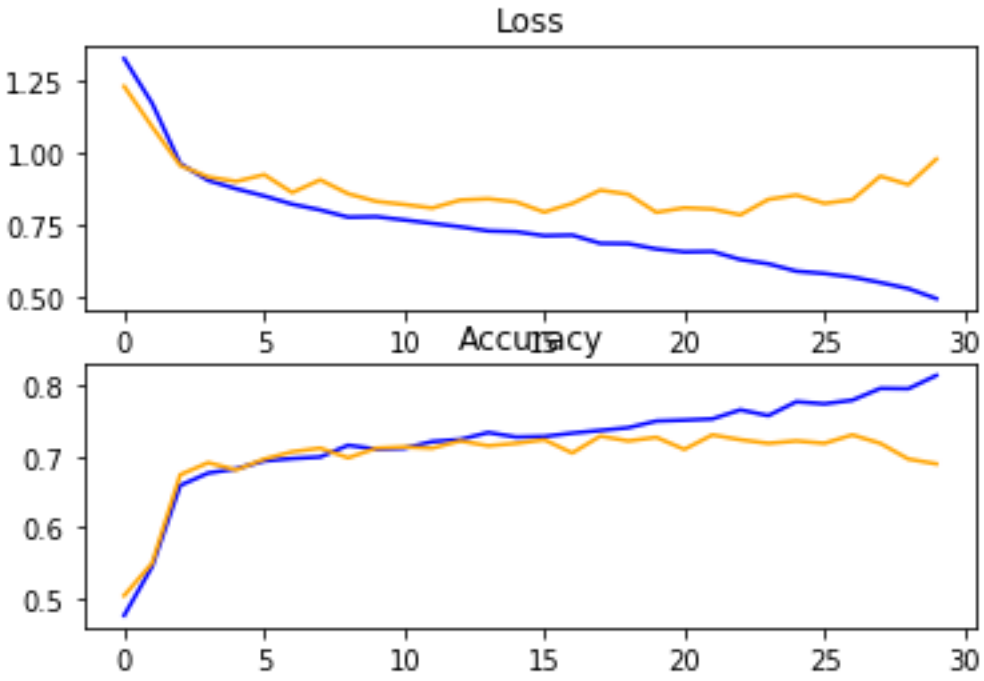

FIGURE 19: VGG8 MODEL 2 ACCURACY PLOT

Here blue solid line presents the training performance and on the other hand yellow solid line shows the testing performance of the model.

Model 3

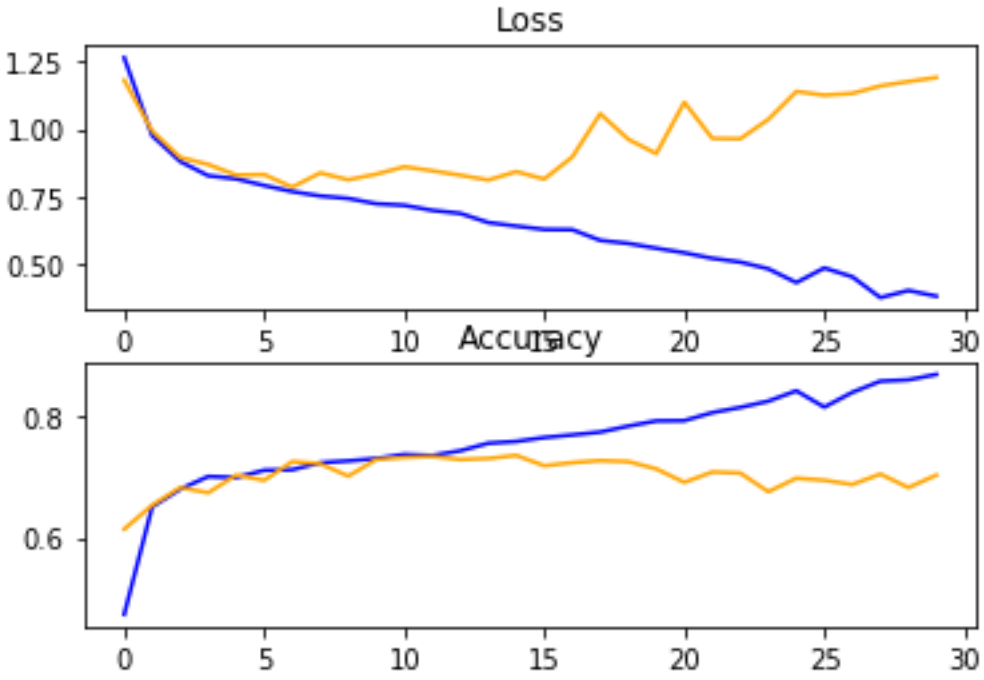

FIGURE 20: VGG8 MODEL 3 ACCURACY PLOT

Here blue solid line presents the training performance and on the other hand yellow solid line shows the testing performance of the model.

Out of the models, Model 3 seems to perform the best. It doesn't seem to overfit and achieves the best performance. Let's use these specifications to create the final model.

Summary of final Model is shown below:

```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================
conv2d_25 (Conv2D)           (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d_13 (MaxPooling (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_26 (Conv2D)           (None, 29, 29, 64)        18496
_________________________________________________________________
conv2d_27 (Conv2D)           (None, 27, 27, 64)        36928
_________________________________________________________________
max_pooling2d_14 (MaxPooling (None, 13, 13, 64)        0
_________________________________________________________________
conv2d_28 (Conv2D)           (None, 11, 11, 128)       73856
_________________________________________________________________
conv2d_29 (Conv2D)           (None, 9, 9, 128)         147584
_________________________________________________________________
conv2d_30 (Conv2D)           (None, 7, 7, 128)         147584
_________________________________________________________________
max_pooling2d_15 (MaxPooling (None, 3, 3, 128)         0
_________________________________________________________________
flatten_5 (Flatten)          (None, 1152)              0
_________________________________________________________________
dense_9 (Dense)              (None, 512)               590336
_________________________________________________________________
dropout_5 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_10 (Dense)             (None, 5)                 2565
=================================
Total params: 1,018,245
Trainable params: 1,018,245
Non-trainable params: 0
_________________________________
```

Final model Accuracy and Loss Plot is shown below

FIGURE 21: VGG8 FINAL MODEL ACCURACY PLOT

Here in figure blue solid represents the training accuracy of the model on the other hand yellow solid line presents the testing accuracy of the model along with epochs on x-axis and accuracy on y-axis.

For validation we can test the mode on test data

```
733/733 [==============================] - 3s 4ms/step
0.6998635530471802
```

the model shows 70% accuracy on test set

After creating it the model is trained again on different dataset and we got 74% of accuracy on test data.

4.10MODEL COMPARISON

All the developed Models such as VGG16, VGG19, InceptionV3, Xception Model, ResNet, Keras CNN and VGG8 are compared below and it is clear from the figure VGG 16 has the highest accuracy.

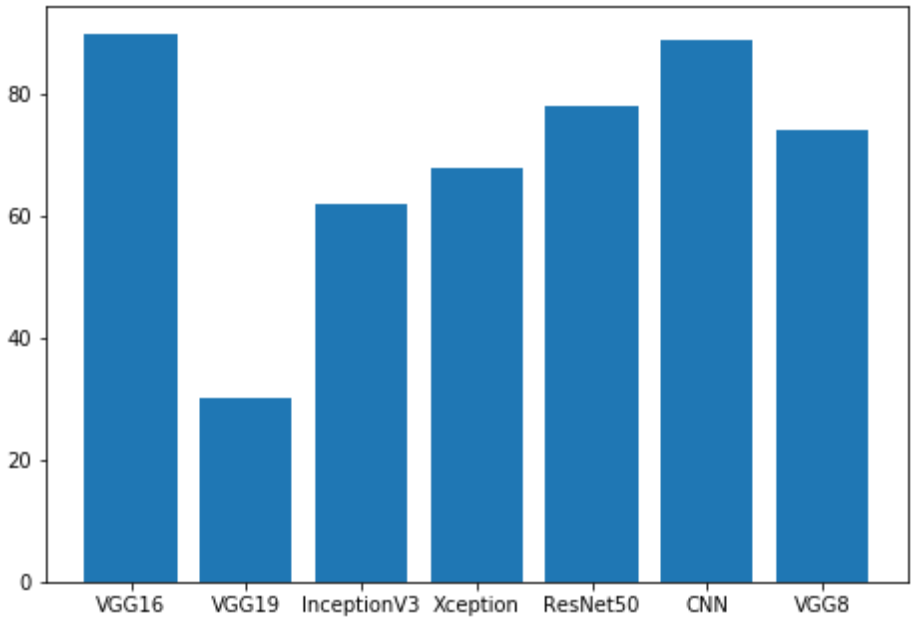

FIGURE 22: MODELS ACCURACY COMPARISON

CONCLUSION

Based on the implemented models and produced results the project has fully answered research question in section 1.2. In addition, all the research objectives (refer section 1.3) have been tackled and the results have been answered. The developed models and results will contribute significantly to the body of knowledge and also improve the field of Diabetic Retinopathy Detection.

5. CONCLUSION

From these experiments we come to know that transfer learning a deep learning approach is a good technique for doing human work. and we see that in future human work will be replaced by machine to make human life easier. In this report pre-trained deep learning are used to perform diabetic retinopathy classification from Gaussian filtered images of eye retina from kaggle. Best transfer learning techniques are used for this task. We showed the performance of different transfer learning models and their result for proposed dataset. The deep transfer learning models selected are Vgg8, vgg16, InceptionV3, Xception Model, ResNet50, CNN and vgg19 and with basic machine learning techniques (Decision Tree, Random Forest and SVM). Theoverall performance of these techniques showed that transfer learning models showed efficient performance and vgg16 Model (transfer learning model) achieved highest accuracy with less training time and computational complexity.

Future work

In detecting Diabetic retinopathy transfer learning models performs well as compare to other t approaches discussed in report and in literature as they are fast learning algorithm and shows promising results on less data. it all depends on data we are feeding to our model. In our data there are less samples of level 4 and level 2 and have many outliers. We also have less resources to process the image with this much dimension. In the future with the help of more powerful machines this work be done more accurately by machines.

BIBLIOGRAPHY/REFERENCES

[1] "Facts About Diabetic Eye Disease | National Eye Institute." https://nei.nih.gov/health/diabetic/retinopathy (accessed May 10, 2019).

[2] "Diabetic Retinopathy Symptoms." https://www.diabetes.co.uk/diabetes-complications/retinopathy-symptoms.html (accessed May 10, 2019).

[3] J. Lachure, A. V. Deorankar, S. Lachure, S. Gupta, and R. Jadhav, "Diabetic Retinopathy using morphological operations and machine learning," in *2015 IEEE International Advance Computing Conference (IACC)*, 2015, pp. 617–622.

[4] "Prevalence of Diabetic Retinopathy and influence factors among newly diagnosed Diabetics in Rural and Urban Areas of Pakistan: Data analysis from the Pakistan National Blindness & Visual Impairment Survey 2003." https://pjms.com.pk/issues/octdec208/article/article1.html (accessed May 16, 2019).

[5] "Data and statistics," May 15, 2019. http://www.euro.who.int/en/health-topics/noncommunicable-diseases/diabetes/data-and-statistics (accessed May 16, 2019).

[6] "The Prevalence of Diabetic Retinopathy Among Adults in the United States | National Eye Institute." https://nei.nih.gov/eyedata/pbd3 (accessed May 16, 2019).

[7] "Diabetic Retinopathy - Asia - American Academy of Ophthalmology." https://www.aao.org/topic-detail/diabetic-retinopathy-asia (accessed May 10, 2019).

[8] myDr, "Retinopathy in diabetes (diabetic retinopathy)." https://www.mydr.com.au/diabetes/retinopathy-in-diabetes (accessed May 16, 2019).

[9] "The 4 Stages of Diabetic Retinopathy: What You Can Expect," *Griswold Home Care*, Jan. 15, 2015. https://www.griswoldhomecare.com/blog/the-4-stages-of-diabetic-retinopathy-what-you-can-expect/ (accessed May 10, 2019).

[10] A. C. MacCuish, "Early detection and screening for diabetic retinopathy," *Eye*, vol. 7, no. 2, pp. 254–259, Mar. 1993, doi: 10.1038/eye.1993.59.

[11] S. Jones and R. T. Edwards, "Diabetic retinopathy screening: a systematic review of the economic evidence," *Diabet. Med.*, vol. 27, no. 3, pp. 249–256, Mar. 2010, doi: 10.1111/j.1464-5491.2009.02870.x.

[12] "Diabetic Retinopathy," *Research to Prevent Blindness*. https://www.rpbusa.org/rpb/resources-and-advocacy/resources/rpb-vision-resources/diabetic-retinopathy/ (accessed May 16, 2019).

[13] R. Raman, S. Srinivasan, S. Virmani, S. Sivaprasad, C. Rao, and R. Rajalakshmi, "Fundus photograph-based deep learning algorithms in detecting diabetic retinopathy," *Eye*, vol. 33, no. 1, Art. no. 1, Jan. 2019, doi: 10.1038/s41433-018-0269-y.

[14] A. Rakhlin, "Diabetic Retinopathy detection through integration of Deep Learning classification framework," Pathology, preprint, Nov. 2017. doi: 10.1101/225508.

[15] M. T. Hagos and S. Kant, "Transfer Learning based Detection of Diabetic Retinopathy from Small Dataset," p. 9.

[16] N. E. M. Khalifa, M. Loey, M. H. N. Taha, and H. N. E. T. Mohamed, "Deep Transfer Learning Models for Medical Diabetic Retinopathy Detection," *Acta Inform Med*, vol. 27, no. 5, pp. 327–332, Dec. 2019, doi: 10.5455/aim.2019.27.327-332.

[17] C. Lam, D. Yi, M. Guo, and T. Lindsey, "Automated Detection of Diabetic Retinopathy using Deep Learning," *AMIA Jt Summits Transl Sci Proc*, vol. 2018, pp. 147–155, May 2018.

[18] M. T. Hagos and S. Kant, "Transfer Learning based Detection of Diabetic Retinopathy from Small Dataset," *arXiv:1905.07203 [cs]*, May 2019, Accessed: Jul. 23, 2020. [Online]. Available: http://arxiv.org/abs/1905.07203.

[19] S. Masood, T. Luthra, H. Sundriyal, and M. Ahmed, "Identification of diabetic retinopathy in eye images using transfer learning," in *2017 International Conference on Computing,*

Communication and Automation (ICCCA), May 2017, pp. 1183–1187, doi: 10.1109/CCAA.2017.8229977.
[20] M. Maliha, A. Tareque, and S. S. Roy, "Diabetic Retinopathy Detection Using Machine Learning," p. 41.
[21] R. Priya and P. Aruna, "Diagnosis of diabetic retinopathy using machine learning techniques," *ICTACT Journal on soft computing*, vol. 3, no. 4, pp. 563–575, 2013.
[22] W. Khan, "Diabetic Retinopathy Detection using Image Processing: A Survey," Accessed: May 16, 2019. [Online]. Available: https://www.academia.edu/26986641/Diabetic_Retinopathy_Detection_using_Image_Processing_A_Survey.
[23] S. Gupta, "Diagnosis of Diabetic Retinopathy using Machine Learning," vol. 3, no. 2, p. 6, 2015.
[24] "Neural Network Technique for Diabetic Retinopathy Detection," *ijeat*, vol. 8, no. 6, pp. 440–445, Aug. 2019, doi: 10.35940/ijeat.E7835.088619.
[25] W. L. Yun, U. R. Acharya, Y. V. Venkatesh, C. Chee, L. C. Min, and E. Y. K. Ng, "Identification of different stages of diabetic retinopathy using retinal optical images," *Information sciences*, vol. 178, no. 1, pp. 106–121, 2008.
[26] N. Singh and R. C. Tripathi, "Automated Early Detection of Diabetic Retinopathy Using Image Analysis Techniques," *IJCA*, vol. 8, no. 2, pp. 18–23, Oct. 2010, doi: 10.5120/1186-1648.
[27] O. Faust, R. Acharya U., E. Y. K. Ng, K.-H. Ng, and J. S. Suri, "Algorithms for the Automated Detection of Diabetic Retinopathy Using Digital Fundus Images: A Review," *J Med Syst*, vol. 36, no. 1, pp. 145–157, Feb. 2012, doi: 10.1007/s10916-010-9454-7.
[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385 [cs]*, Dec. 2015, Accessed: Aug. 23, 2020. [Online]. Available: http://arxiv.org/abs/1512.03385.

APPENDICES

Content of Artifacts

- DataSet Link.txt
- Descriptive Analysis Diabetic.py
- Diabetic retinopathy using CNN Keras.py
- Diabetic Retinopathy using Decision_Tree.py
- Diabetic retinopathy using InceptionV3.py
- Diabetic Retinopathy using random forest.py
- Diabetic retinopathy using ResNet50.py
- Diabetic Retinopathy using svm.py
- Diabetic retinopathy using vgg8 from scratch.py
- Diabetic Retinopathy using VGG19.py
- Diabetic Retinopathy using Xception.py
- Diabetic transfer learning using vgg16.py
- Readme.txt
- Requirement.txt

Pre-requisites

To conduct this research in any other system the system must have installed **Anaconda-python** in it that automatically installs python and following libraries:

- Numpy
- Sklearn
- Time
- Scipy
- os
- Seaborn
- Matplotlib
- Pandas
- Defaultdict
- Tensorflow
- Keras
- Pickle
- Random

Glossary of terms

- ANN (Artificial Neural Networks; systems inspired by human biological neural networks used for learning)
- Machine learning (the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions.)
- CNN (Convolutional Neural Networks; deep feed forward artificial neural networks for analyzing visual records and images.)
- Deep Learning (hierarchical learning inspired by the function of human brain)

- LSTM (Long short term memory; capable of learning long term Dependencies, a subdivision of RNN)
- RNN (Recurrent neural networks; to process sequence of information)
- Transfer Learning (An approach that uses knowledge gained on one model and apply to other but similar problem)