A Project Report

on

Diabetic Retinopathy Detection Using Deep Learning

carried out as part of the course CC1634 Submitted by

Aditya Singh Goliya
189303130

Satvik Pandey
189303165

VI Semester, B. Tech Computer and Communication

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

In

Computer & Communication Engineering

Department of Computer & Communication Engineering,
School of Computing and IT,
Manipal University Jaipur,
July,2021

CERTIFICATE

This is to certify that the project entitled "_***Diabetic Retinopathy Detection Using Deep Learning***_" is a bonafide work carried out as part of the course ***CC1634 Minor Project***_ , under my guidance by ***Aditya Singh Goliya (189303130), Satvik Pandey (189303165)***_, student of ***B. Tech Computer and Communication***_ at the Department of Computer & Communication Engineering , Manipal University Jaipur, during the academic semester _***VI***__, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer & Communication Engineering, at MUJ, Jaipur.

Place: Manipal University Jaipur

Date: 14th June 2021 Signature of the Instructor (s)

DECLARATION

I hereby declare that the project entitled **"*Diabetic Retinopathy Detection Using Deep Learning*"** submitted as part of the partial course requirements for the course ***CC1634 Minor Project***, for the award of the degree of Bachelor of Technology in Computer & Communication Engineering at Manipal University Jaipur during the ***VI, July 2021*** semester, has been carried out by me. I declare that the project has not formed the basis for the award of any degree, associate ship, fellowship, or any other similar titles elsewhere.

Further, I declare that I will not share, re-submit, or publish the code, idea, framework and/or any publication that may arise out of this work for academic or profit purposes without obtaining the prior written consent of the Course Faculty Mentor and Course Instructor.

Signature of the Student:

Place: Manipal University Jaipur.

Date:14th June 2021.

Abstract

Diabetes, referred to as Diabetes Mellitus, describes a group of metabolic diseases in which the person has high blood glucose. Possible complications that can be caused by badly controlled diabetes: Eye complications, Foot complications, Skin complications, Heart problems, Hypertension etc. Diabetic Retinopathy (DR), a common complication of diabetes, affects the blood vessels in the retina. It is due to retina not receiving enough oxygen.

Diabetic Retinopathy (DR) is an eye ailment which influences eighty to eighty-five percent of the patients who have diabetes for more than ten years. The retinal fundus images are commonly used for detection and analysis of diabetic retinopathy disease in clinics. The raw retinal fundus images are very hard to process by machine learning algorithms.

The manual diagnosis process of DR retina fundus images by ophthalmologists is time, effort, and cost-consuming and prone to misdiagnosis unlike computer-aided diagnosis systems. Recently, deep learning has become one of the most common techniques that has achieved better performance in many areas, especially in medical image analysis and classification. Convolutional neural networks are more widely used as a deep learning method in medical image analysis, and they are highly effective.

A dataset containing 63000 images has been used for our study. This dataset was picked up from Kaggle.

In this project, we aim to build a CNN model with high accuracy, which we would further embed into a web app so that it could be accessed anywhere and anyone even without the know-how of how the model works.

Table Of Contents

Table Of Content-Pictures

1. Introduction

Diabetic Retinopathy (DR) is leading cause of blindness. As per World Health Organization report more than 347million people are having diabetic problems in 2030. Recent years maximum diabetes patients are getting malformation in retina which is called diabetic retinopathy (DR). People with age 30 or above have 78% chance of DR if they suffer with diabetes for more than 15 years. This rate increases to 97% if age is below 30 and suffers with diabetes for same period. Hence DR affects working age adult.

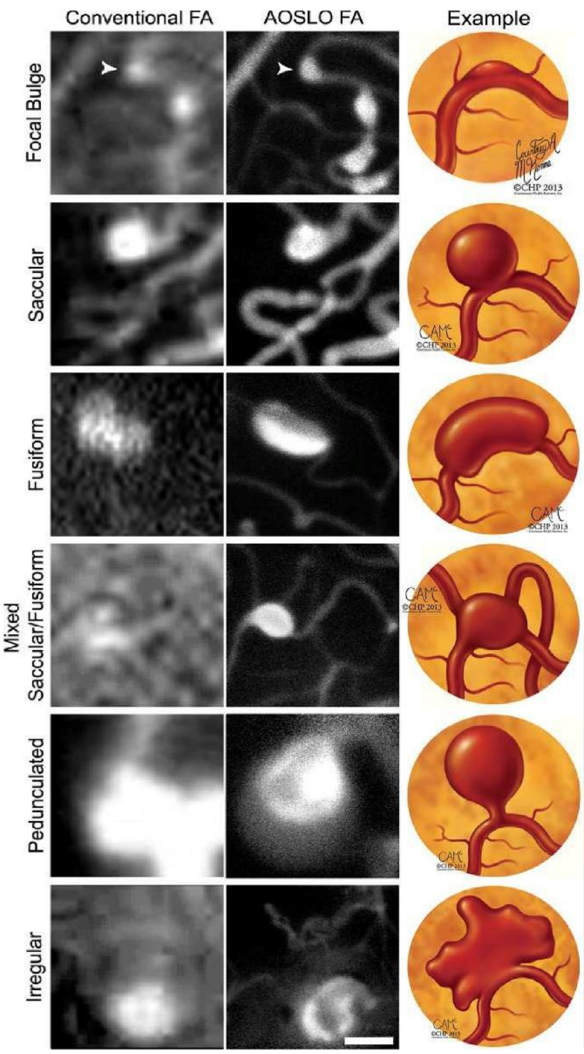
In the recent years, there has been a dramatic increase in the number of diabetic patients suffering from diabetic retinopathy (DR). DR is one of the most chronic diseases which make the key cause of vision loss in middle-aged people in the developed world.

Retinopathy means damage to retina. DR appears due to longstanding of diabetes mellitus and because of that blood vessel becomes blocked, leaky and grows haphazardly. DR never shows any interference with sight till it reaches to advanced phase. Hence, eye screening for DR is essential at early stage. There are two stages of DR. Earlier one is known as non-proliferative diabetic retinopathy (NPDR) and latter one as proliferative diabetic retinopathy (PDR). The features in DR diagnosis are microaneurysms, hemorrhages, hard exudates, cotton-wool spot, abnormal new vessels, venous bending, dilations, and segmentations. The microaneurysms are the small red dots on the retina, and they represent the earliest visible sign of the DR. So, detection of microaneurysms at an early stage is the first step in preventing DR. The conventional detection methods are Visual Acuity Test, pupil dilation, Ophthalmoscopy or Fundus Photography, Fundus Fluorescein Angiography (FFA), Optical Coherence Tomography (OCT), digital retinal screening programs or services, computer vision approach, slit lamp bio-microscopy retinal screening programs.

DR Severity Level	Lesions
No DR	Absence of lesion
Mild non-proliferative DR	Micro-Aneurysms only
Moderate DR	More than just Micro-Aneurysms but less than severe DR
Severe DR	Any of the following: • more than 20 intraretinal HM* in each of 4 quadrants • definite venous beading in 2+quadrants • Prominent intraretinal microvascular abnormalities in 1+ quadrant • no signs of proliferative DR
Proliferative DR	One or more of the following: vitreous/pre-retinal HM, neovascularization

Microaneurysms (MA): - The earliest sign of DR that appears as small red round dots on the retina due to the weakness of the vessel's walls. The size is less than 125 μm and there are sharp margins

Hemorrhages (HM): - Appear as larger spots on the retina, where its size is greater than 125 μm with an irregular margin. There are two types of HM, which are flame (superficial HM) and blot (deeper HM)



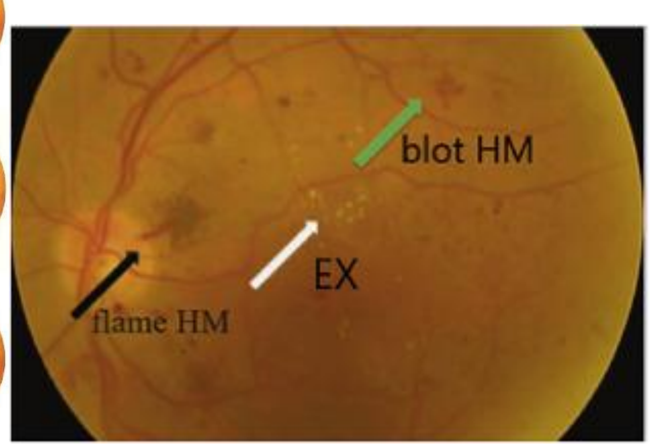
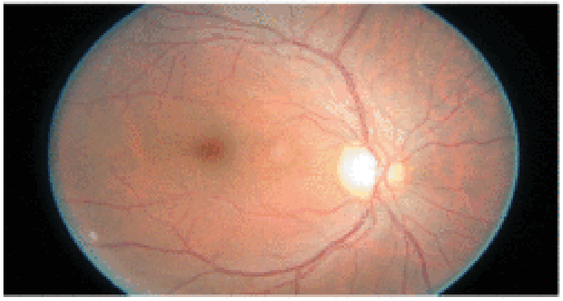
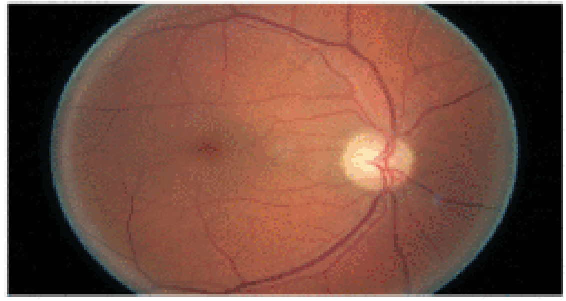


Fig 1-Micro-Aneurysms

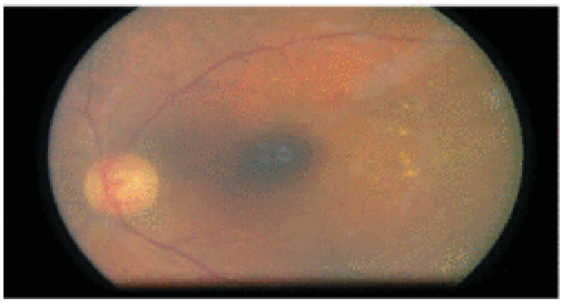
Fig 2-Hemorraghes

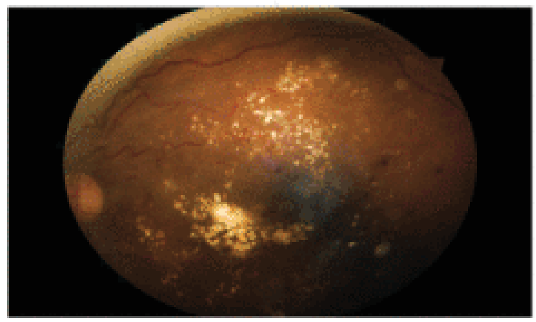




(a) Normal

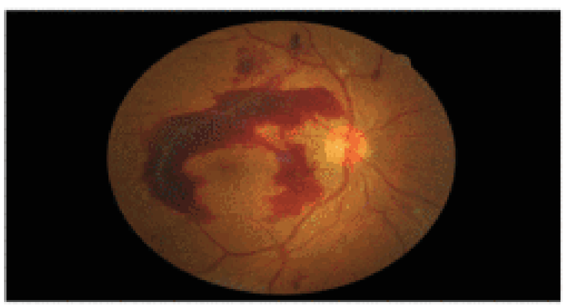
(b) Mild





(c) Moderate

(d) Severe



(e) PDR

Fig 3- Different DR’s

1.1 Motivation

Preventive and early diagnosis of DR is very difficult particularly in rural and remote areas due to acute shortage of ophthalmologists and eye care infrastructure. This problem can be overcome by developing an automated system for detection of DR using fundus images. Mass screening of diabetic patients can be done using automated DR detection algorithms and suspected patients may be sent to eye experts for further diagnosis. For automated DR detection, digital image processing is used for diagnostic feature extraction from fundus images. There are other imaging techniques like FFA, OCT these techniques are costlier and have some side-effect too.

The Different Methods for the detection of DR are: -

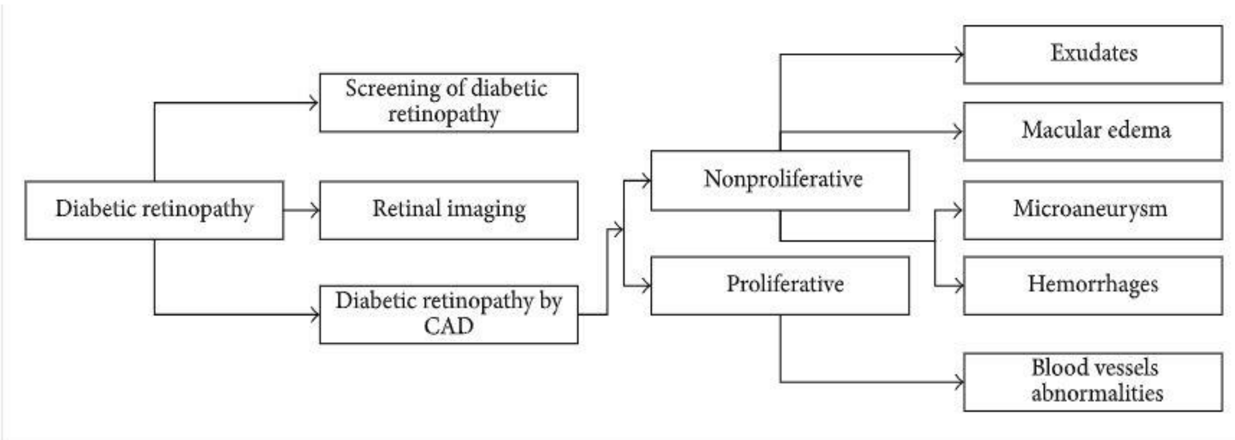


Fig 4- Different Methods for DR Detection

All these methods take a minimum of 1 day before getting the result and hence require a follow-up appointment with the doctor before knowing the result.

Thus, we propose an automated system to deal with such problems and many more.

With regards to a limited medical staff, an automated system can significantly decrease the manual labor involved in diagnosing large quantities of retinal images. While this represents an obvious and significant gain, there is a larger, logistical need for automated and immediate diagnoses in rural settings: patients in rural areas are fundamentally harder to reach than in urban environments. In the mobile hospital setting, if a patient comes for the morning session, there is no guarantee that the same patient will return for the afternoon session unless the need for continued diagnosis is demonstrated.

Consequently, if a patient has diabetic retinopathy, it is essential to convey the urgency of the situation to the patient as soon as a diagnosis can be reached to convince them to travel to a hospital for full treatment. This is a fundamental health care delivery problem in many rural developing regions. With an automated system, the doctor or local health worker can be made aware of the diabetic retinopathy problem during a single session with a patient. This enables the medical personnel to immediately and visually demonstrate the existing problem to the patient which makes it easier to convince them of the urgency of their situation.

2. Literature Review
2.1 DR Feature Extraction

Computer-aided diagnosis of diabetic retinopathy has been explored in the past to reduce the burden on ophthalmologists and mitigate diagnostic inconsistencies between manual reader [1]. Automated methods to detect microaneurysms and reliably grade fundoscopic images of diabetic retinopathy patients have been active areas of research in computer vision [2]. The first artificial neural networks explored the ability to classify patches of normal retina without blood vessels, normal retinas with blood vessels, pathologic retinas with exudates, and pathologic retinas with microaneurysms. The accuracy of being able to detect microaneurysms compared to normal patches of retina was reported at 74% [10].

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 microaneurysm detection [4]. However, a variety of other features besides microaneurysms are efficacious for disease detection.

Different features extraction that may be useful as well are Blood vessels, exudates, hemorrhages, microaneurysms and maculopathy detection techniques.

- For identification of blood vessels from normalized color images, Kirsch's method is used then it is gone through enhancement using Kirsch's template and spatial averaging filtration then histogram equalization and binarization is performed on that image. This method proposed with sensitivity more than 91% and specificity of 90.5% [5].

Exudates are collection of lipid and protein in the retina. On the retina they are seen as bright, reflective, white- or cream-colored lesions. It can cause of increase vessel permeability which is a risk for retinal edema. Though they close to macula centre, they are considered as sight threatening lesions.

- This paper investigated and proposed a set of optimally adjusted morphological operators to detect exudates on diabetic retinopathy patient's non-dilated pupil and low contrast images. The automatically detected exudates by this method was validated with expert ophthalmologist's hand-drawn ground truth data. This system acquired a sensitivity of 80% and specificity of 99.5% respectively [6].

Microaneurysm is a major feature to detect diabetic retinopathy because these structures constitute the earliest recognizable elements to detect diabetic.

- First step of this method included image enhancement, shade correction and image normalization of the green channel whether second step included detecting the candidate in which all possibility of microaneurysms were detected. This method achieved a sensitivity of 88.5%. Based on diameter closing and kernel density estimation for automatic classifications, microaneurysm was detected [7].

2.2 Deep Learning – Literature Review

Deep learning (DL) is a branch of machine learning techniques that involves hierarchical layers of non-linear processing stages for unsupervised features learning as well as for classifying patterns. DL is one computer-aided medical diagnosis method [8]. DL applications to medical image analysis include the classification, segmentation, detection, retrieval, and registration of the images.

Recently, DL has been widely used in DR detection and classification. It can successfully learn the features of input data even when many heterogeneous sources integrated [9]. There are many DL-based methods such as restricted Boltzmann Machines, convolutional neural networks (CNNs), auto encoder, and sparse coding. The performance of these methods increases when the number of training data increase [10] due to the increase in the learned features unlike machine learning methods. Also, DL methods did not require hand-crafted feature extraction.

A Convolutional Neural Network (CNN) is comprised of one or more convolutional layers (often with subsampling step) and then followed by one or more fully connected layers as in a standard multilayer neural network. The architecture of a CNN is designed to take advantage of the 2D hierarchical structure of an input image (or other 2D input such as a speech signal). This is achieved with local connections and tied weights followed by some form of pooling which results in translation invariant features. Another benefit of CNNs is that they are easier to train and have many fewer parameters than fully connected networks with the same number of hidden units [11]. CNNs also consider the hierarchical representation of images while training by stacking multiple trainable stages on each other

2.3 Related Work- Literature Review

Methods of detecting microaneurysms and grading DR involving k-NN [12], support vector machines [13], and ensemble-based methods [14] have yielded sensitivities and specificities within the 90% range using various feature extraction techniques and preprocessing algorithms.

Previous CNN studies [15] for DR fundus images achieved sensitivities and specificities in the range of 90% for binary classification categories of normal or mild vs moderate or severe on much larger private datasets of 80,000 to 120,000 images. However, accuracy measures for the detection of four classes of DR, that is: no DR (DR0), mild (DR1), moderate (DR2), and severe (DR3) depend nontrivially on disease graded class collection ratios. While DR0 and DR3 stages can achieve high sensitivity, the DR1 and DR2 computed recall rates are often low. Experiments from publicly available datasets suggest this is primarily attributable to the relative difficulty of detecting early-stage DR.

In [16], the authors checked the performance of the Kaggle dataset over different CNN models. Garcia [16] proposed a method of using the right and left eye images separately and applied CNN (Alexnet, VGGnet16, etc.). The preprocessing and augmentation phases were performed on the dataset to improve the contrast of images. They achieve the best results on VGG16 with no fully connected layer and achieved 93.65% specificity, 54.47% sensitivity, and 83.68% accuracy. However, DR stages were not explicitly classified in their work.

Dutta [12] used Kaggle dataset with three deep learning models (Feed Forward Neural Network (FNN), Deep Neural Network (DNN), and Convolutional Neural Network (CNN). They used 2000 images out of 35128 images with a 7:3 validation split. They applied many preprocessing steps (median, mean, Std deviation, etc.) and then trained their model on the training dataset. The Best training accuracy of 89.6% was obtained on DNN.

2.4 Problem Statement

Diabetic retinopathy is a leading cause of blindness. Over 65.1 million people in India are diagnose with diabetes and this number will rise to 109 million by 2035. Therefore, it becomes need of the hour to build safe and reliable system that will work on early detection of this disease and will provide optimum and genuine results. Development of such project will lead to a social cause that will help to provide solution for diabetic retinopathy.

In medical field, diagnosis of diseases competently carried out by using the image processing. Therefore, that to retrieve the relevant data from the amalgamation of resulting image is too difficult. Here the segmentation technique is very useful by semi-supervised learning then the result can be tuned by using Deep Learning Neural Network. Deep neural networks have been investigated in learning latent representations of medical images, yet most of the studies limit their approach in a single supervised convolutional neural network (CNN), which usually rely heavily on a large-scale annotated dataset for training. To learn image representations with less supervision involved, this problem can be solved using a deep CNN architecture that can be trained with only binary image pair information. Some researchers evaluated the learned image representations on a task of content-based medical image retrieval using a publicly available multiclass diabetic retinopathy fundus image dataset. The problem can be solved using deep CNN which requires much less supervision for training.

2.5 Research Objective

Our research objective is to learn about the ways of automated detection of Diabetic Retinopathy in the early stages and to make a model that would provide better accuracy with considerably less time to train.
We want to cut down the time it takes to take a photo of the cornea and then get it processed in a lab and then schedule another meeting with the patient to tell the result, this entire process takes a lot of time, and we want to cut down the middleman so that the image can be processed anywhere and anytime without any difficult after just having access to a PC or Laptop with internet capabilities.
We want to develop a model which provides high accuracy not only in the training dataset but also with unseen data.
We aim to research various existing Deep Learning models and CNN structures so that we can formulate an ensemble model of our own that can provide with high accuracy and help us reach one step closer to our objective.

3. Methodology and Framework

3.1 System Architecture

The System used to perform the development of following models is: -

- Processor: -Intel(R) Core (TM) i7-9750H CPU @ 2.20GHz, 2208 MHz, 6 Core(s), 12 Logical Processor(s)
- GPU: - GeForce RTX 2070 With Max-Q Design
- VRAM: - 8192 MB
- RAM: - 16GB

Software and packages used while developing our model: -

- Language Used: - Python 3.9
- Packages Used: - Keras, TensorFlow, Pandas, Matplotlib, Seaborn, skimage, PIL, CV2, sklearn, time, os, sys, numpy, itertools.
- Environment Used: - Anaconda
- Text Editor Used: - Spyder
- Notebook Used: - Jupyter Notebook

3.2 Algorithm Techniques

First, we performed EDA to know about our data a little bit and about how balanced our dataset is.

Then, we started our work with preprocessing of the images so that can be used as an input for our models. While pre-processing our aim was to reduce the size of the files as the data was over 85 GB and due to computation limitations, we had to reduce the image sizes, otherwise processing them would not be possible for us.

After preprocessing, we augmented our images and thus our images were ready for input.

Then we proceeded towards our first model i.e., EyeNet that consisted of 3 convolutional layers that we solely got after a huge amount of hit and trials. We realized that this is not the way to go due to the time it was consuming, and the accuracy was still only about 80%.

Thus, we moved on towards preexisting models such as VggNet, GoogleNet, ResNet, AlexNet. Then after spending some time tuning our hyper parameters, we realized we have reached a stump and our model could not be improved anymore if we wanted to avoid over-fitting.

Thus, after some research we moved towards building an Ensemble Model and finally settled using an Ensemble Model.

3.3 Detailed Design Methodologies
EDA

All images are taken by different people, using different cameras, and of different sizes. Pertaining to the preprocessing section, this data is extremely noisy, and requires multiple preprocessing steps to get all images to a useable format for training a model.

The training data is comprised of 35,126 images, which are augmented during preprocessing.

The very first item analyzed was the training labels. While there are five categories to predict against, the plot below shows the severe class imbalance in the original dataset.

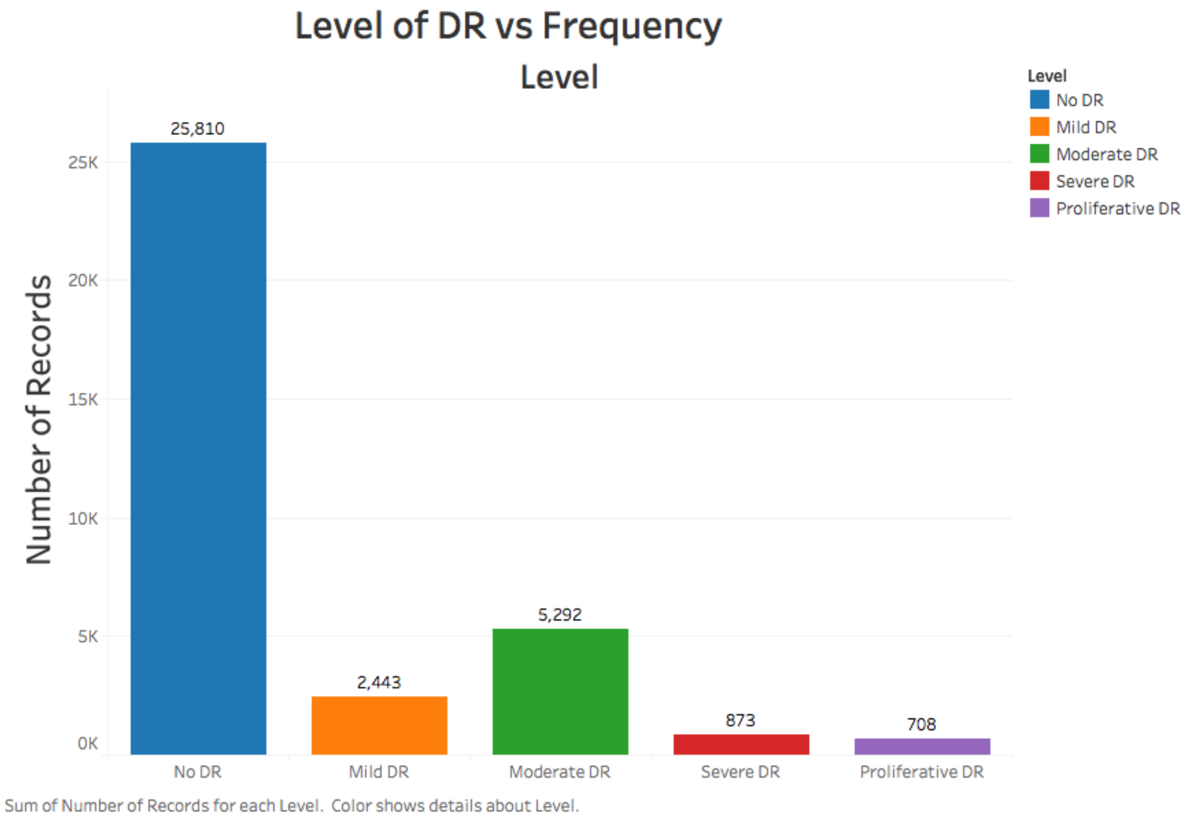


Fig 5

Of the original training data, 25,810 images are classified as not having retinopathy, while 9,316 are classified as having retinopathy. Due to the class imbalance, steps taken during preprocessing to rectify the imbalance, and when training the model. Furthermore, the variance between images of the eyes is extremely high. The first two rows of images show class 0 (no retinopathy); the second two rows show class 4 (proliferative retinopathy).

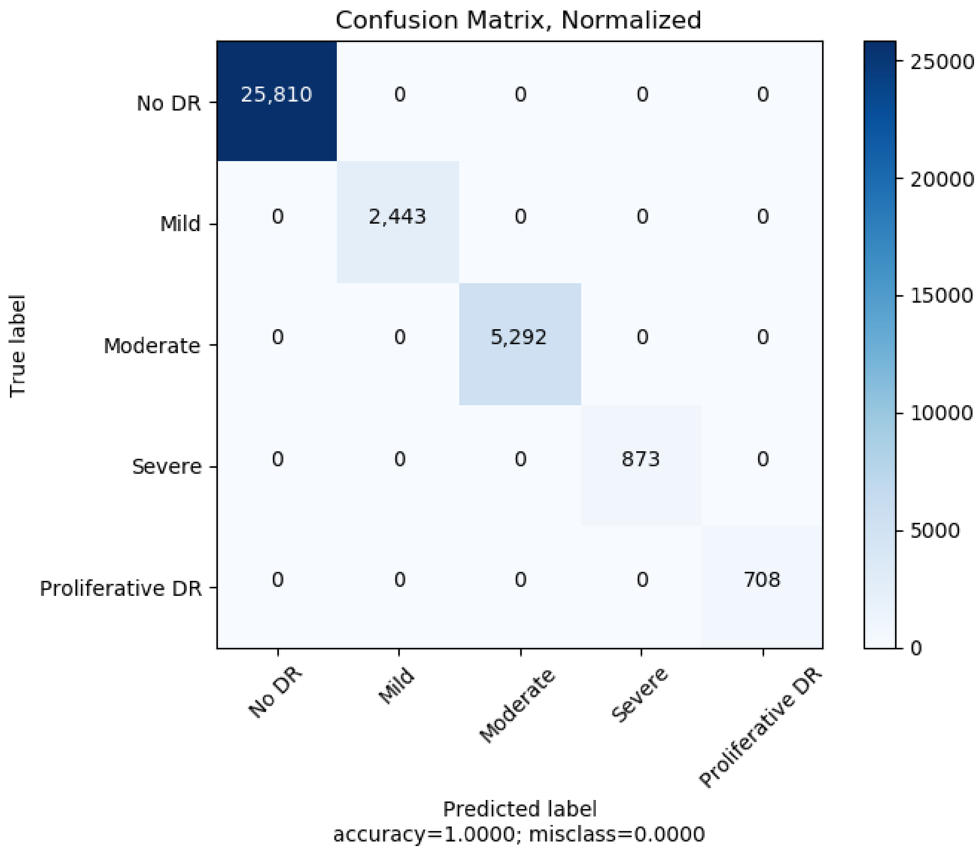


Fig 6

Image Preprocessing

Crop images into 1800x1800 resolution

- In total, the original dataset totals 89.8 gigabytes. All images were cropped to 1800x1800 pixels to retain consistent image over the dataset.

Resize images to 512x512/256x256 resolution

- All images were scaled down to 512x512 and 256x256 pixels. Despite taking longer to train, the detail present in photos of this size is much greater then at 128x128 pixels.

Remove totally **black images** form dataset

- Additionally, 403 images were dropped from the training set as Scikit-Image raised multiple warnings during resizing. The reason behind raising the warning is that some images does not have color space. Because of this, any images that were completely black were removed from the training data.

Rotate and Mirror (Rotate DR images to 90°,120°,180°,270° + mirror, and only mirror non-DR images)

- All images were rotated and mirrored. Images without retinopathy were mirrored; images that had retinopathy were mirrored, and rotated 90, 120, 180, and 270 degrees.
- The first images show two pair of eyes, along with the black borders. Notice how most of the noise is removed

  Before: -

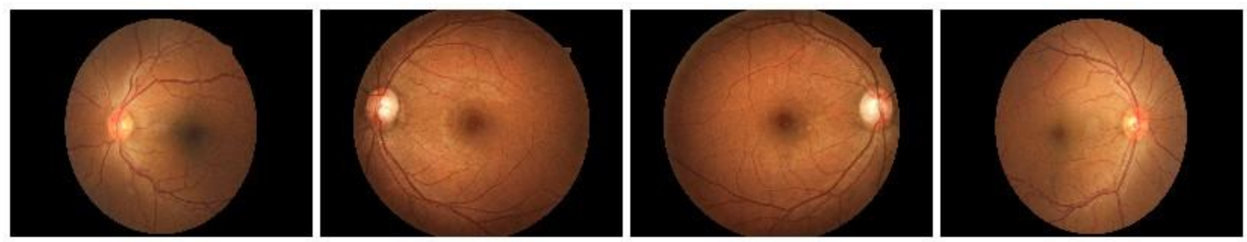


Fig 7

After: -

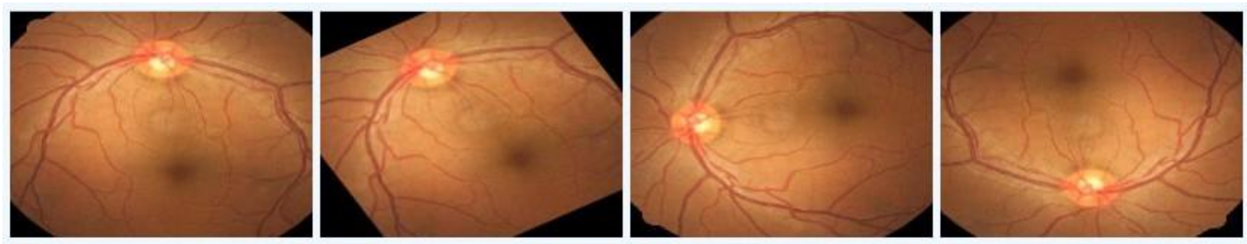


Fig 8

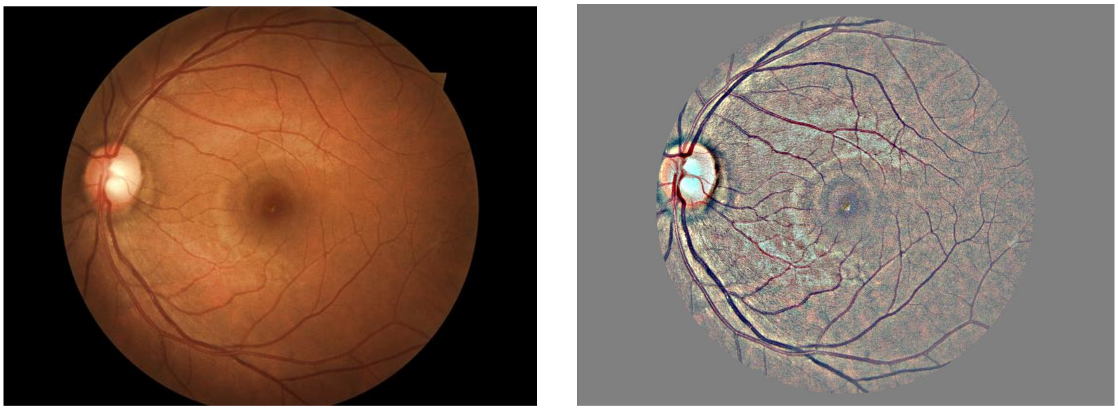
After rotations and mirroring, the class imbalance is rectified, with a few thousand more images having retinopathy. In total, there are 106,386 images being processed by the neural network.

Subtract the local average color; the local average gets mapped to 50% gray,

Clip the images to 90% size to remove the "boundary effects".

This was intended to remove some of the variation between images due to differing lighting conditions, camera resolution, etc. Here are two before/after examples.

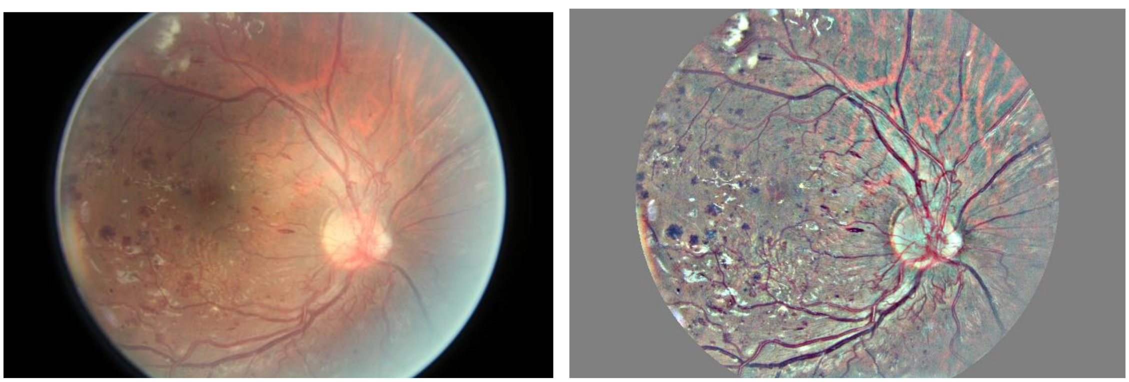
021_left -



Before After

Fig 9

030-right -



Before After

Fig 10

We then update the CSV so that it has the new augmented images and their respective labels.

After preprocessing we can see that now the imbalanced class has been balanced a bit.

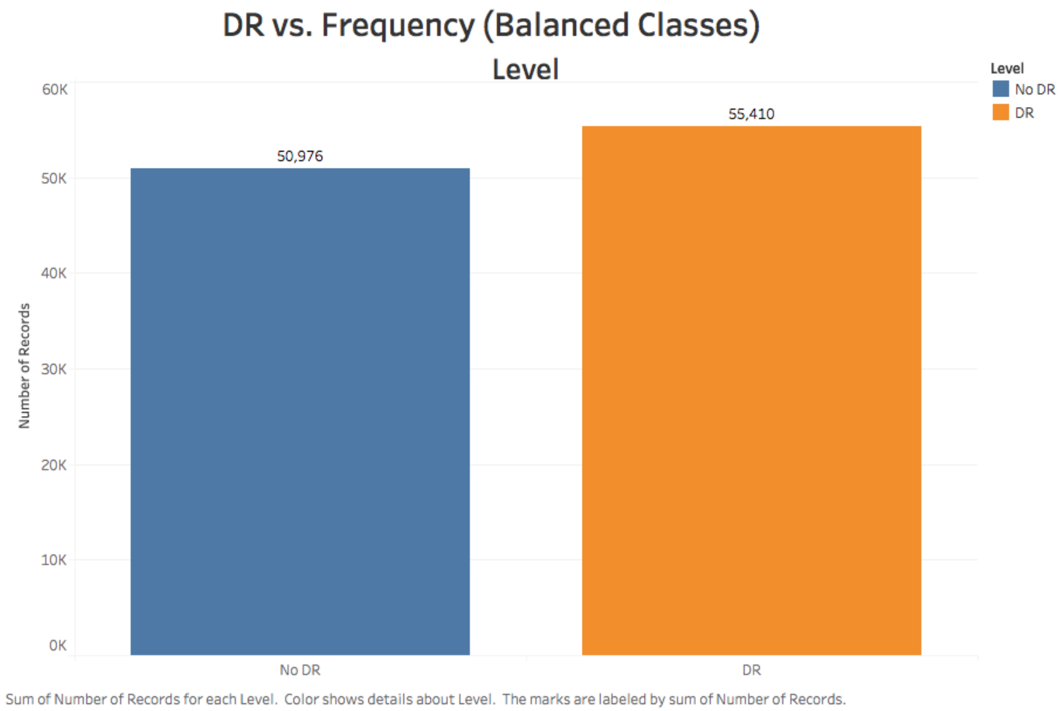


Fig 11

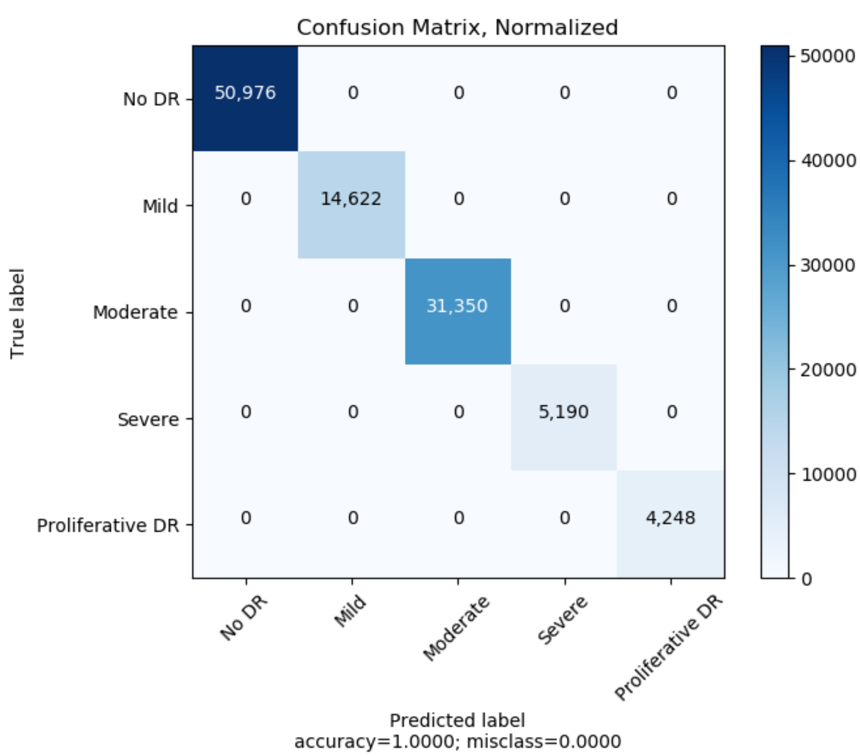


Fig 12

We decided to not convert the images into numpy array. The reason behind not converting the images into a single array is due to its memory consumption. The dataset was large as it is and due to limited computational capability and memory limitation, we decided to skip this preprocessing part.

Performance Parameters

Some Important Definitions

TP (True Positives): - The number of correctly classified instances of the class under observation
TN (True Negatives): - The number of correctly classified instances of rest of the classes
FP (False Positives): - The number of miss-classified instances of rest of the classes
FN (False Negatives): - is the number of miss-classified instances of the class under observation.

Accuracy: The accuracy can be calculated in terms of positive and negative classes:

Accuracy= $$\frac{TP+TN}{TP+TN+FP+FN}$$

Recall: it is the ratio of TP and TP+FN

Recall= $$\frac{TP}{TP+FN}$$

Precision: it is the ratio of TP and TP+FP:

Precision= $$\frac{TP}{TP+FP}$$

F1-Score: it is the weighted harmonic mean of precision and recall:

F1-Score= $$2 \times \frac{Precision \times Recall}{Precision+Recall}$$

Receiver Operating Curve (ROC): Plots the true positive rate (TPR) against the false positive rate (FPR).

Area Under the Curve (AUC): It represents the degree or measure of separability of different classes. The higher the AUC score means the better the model and vice versa.

Model 1

The model is built using Keras, utilizing TensorFlow as the backend. TensorFlow was chosen as the backend due to better performance over Theano, and the ability to visualize the neural network using TensorBoard.

For predicting two categories, EyeNet utilizes three convolutional layers, each having a depth of 32. A Max Pooling layer is applied after all three convolutional layers with size (2,2).

After pooling, the data is fed through a single dense layer of size 128, and finally to the output layer, consisting of 2 SoftMax nodes.

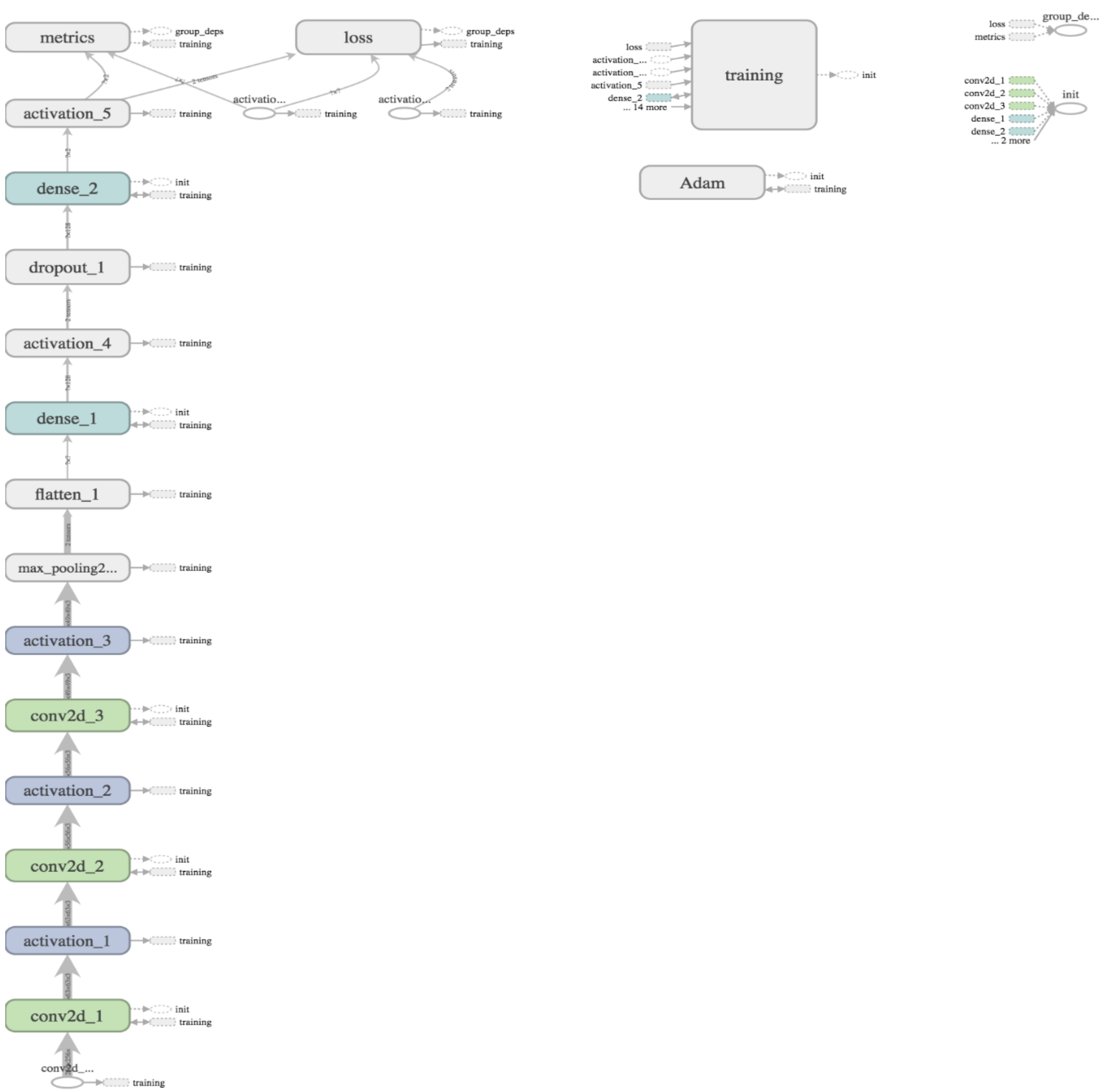


Fig 13-Using TensorBoard

The EyeNet classifier was created to determine if a patient has retinopathy. The current model returns the following scores.

Metric	**Value**
Accuracy (Train)	82%
Accuracy (Test)	80%
Precision	88%
Recall	77%

This model provided us with good intel for our first model due to the help of TensorBoard and its capability to visualize the layers, we could see how the model was working and it provided good practice for our further model development.

We decided to scrap this model and move forward due to many factors such as: -

- Lower accuracy as compared other pre-trained models
- Messy hit-and-trial method to determine the right number of layers
- The complicated the model was, the higher computational power was needed due to the nature and size of the dataset used
- Tuning of hyperparameters led to no further increase in accuracy

Model 2

We used various **pre trained models** available on the internet. The models used are: -

AlexNet

Alex Krizhevsky [21], proposed a deep CNN called AlexNet in 2012, which could be deeper and wider than LeNet. Compared to traditional methods, The architecture of AlexNet is one of the first deep CNN model to promote ImageNet Classification accuracy significantly. For top-5 classification tasks, the error rate is 16.2%.

AlexNet contains 630 million connections, 60 million parameters, and 650,000 neurons, with 5 convolutional layers, of 3 which are followed by the max pooling layer, and finally with 3 fully connected layers.

VggNet

VggNet [22] is a deep CNN developed by researchers at the University of Oxford Vision Geometry Group and Google's DeepMind, working on the connection between the depth and performance of a CNN. It succeeds in constructing a convolution of 16 to 19 deep layers Neural Networks by stacking $3 * 3$ small convolution kernels and $2 * 2$ maximum pooling layers repeatedly. For top-5 classification tasks, the error rate is 24.2%.

Although the network at each level gradually gets darker, the amount of network parameters does not increase much because it is mostly consumed in the last three fully connected layers. Though deep in front of the Convolution, parameters are not consumed largely. However, training is more time-consuming part of the convolution for its computational complexity. VggNet has 5 sections of convolution, each section has 2–3 convolution layers, each section will be connected to the end of a maximum pool to reduce the size of the picture.

GoogleNet

GoogleNet/ InceptionNet V1[23] introduces the inception structure, which maintains computational performance with dense matrices as well as intensives the sparsity of CNNs structure. The way to better the accuracy of the model is to increase the complexity of the model.

GoogleNet used a 22 layers deep CNN in the 2014 ILSVRC competition, which is smaller and more speedily than VggNet, and smaller and more precise than AlexNet on the original ILSVRC images. For top-5 classification tasks, the error rate is 5.5%. The network structure is more complex than VggNet, adding 'Inception' layers to the network structure. Each an 'Inception' layer contains six convolutional and one pooling operation, which decreases the thickness of fusion feature image.

ResNet

Residual Neural Network (ResNet) is put forward by Kaiming [17]. By means of using the Residual Unit, it successfully trains 152 deep neural network to win the ILSVRC 2015 championship and get a 3.57% error rate classification for top 5 classes, which is quite prominent though the number of parameters is less than VggNet.

The ResNet structure is decent that can greatly accelerate the training of ultra-deep neural networks and improve the accuracy of the model.

The Results were: -

Classification results with randomly initialized parameters

of CNNs model.

Model	Prec	Recall	AUC	ACC
AlexNet	90.07%	39.12%	0.7968	73.04%
VggNet-s	93.98%	33.43%	0.7901	73.66%
VggNet-16	29.09%	86.37%	0.5512	48.13%
VggNet-19	96.05%	54.51%	0.7938	82.17%
GoogleNet	86.84%	64.83%	0.7756	86.35%
ResNet	90.53%	73.77%	0.826	78.68%

The parameters of the CNN which need to be set by the user prior to the filter learning are called hyper-parameter. Hyper-parameters are the variables related to the structure of the network (e.g., number of layers and number units in each layer) training (e.g., learning rate). These parameters are adjusted before training (before optimizing the weights and bias). To set the values of other hyper-parameter, we have adopted good practices from literature. Then we did some hyperparameter tuning and we got improved results which are mentioned below in the table.

Classification results with hyperparameter-tuning.

Model	Prec	Recall	AUC	ACC
AlexNet	94.07%	81.27%	0.9342	89.57%
VggNet-s	97.43%	86.47%	0.9786	95.68%
VggNet-16	94.32%	90.78%	0.9616	93.17%
VggNet-19	96.49%	89.31%	0.9684	93.73%
GoogleNet	93.45%	77.66%	0.9272	93.36%
ResNet	95.56%	88.78%	0.9365	90.4%

Here we can observe that different CNNs architectures have different classification performance and the overall classification performances are poor. At the same time, in the process of training, we found that there existed over-fitting phenomenon, to work out the over-fitting problem, we use transfer learning and hyperparameter-tuning methods to classify the fundus images more accurately.

Transfer learning experimental settings are as follows: the fundus images data was increased to 20 times of the original, with 30 training iterations, the learning rate is linear variation between [0.0001-0.1], as well as the stochastic gradient descent optimized method is used to update the weights values. Five times the cross-validation is to compute the results. The accuracy of VggNet-s model classification in the experiment is 95.68%.

The other have poor classification accuracy. This may be because the other architectures have larger and more complicate structure and more training parameters than VggNet-s. More tracing parameters and less training data would produce over-fitting phenomenon, which may yield the less inaccurate classification performance.

Here, we could not move forward with more hyper tuning of the parameters as it seemed that anymore tuning and our models will become overfitted. We could already start to see that our models had high accuracy on the dataset that it was trained on, but with unseen data our model started performing poorly as we tuned our parameters to get the highest accuracy possible more and more.

Thus, we decide to stop and go towards our Third and final model

Model 3

ENSEMBLE MODEL

Ensemble method is a meta-algorithm that combine several machine learning techniques into one predictive model.

The proposed approach ensembles the five deep CNN models Resnet50 [17], Inceptionv3 [18], Xception [19], Dense - 121 [20], Dense169 [20]. Algorithm 1 presents the proposed model in detail. Let H = {Resnet50, Inceptionv3, Xception, Dense121, Dense169} be the set of pre-trained models. Each model is fine-tuned with the Fundus Images dataset (X, Y); where X the set of N images, each of size, 512 × 512, and Y contain the corresponding labels, Y = {y/y ∈ {Normal, Mild, Moderate, Severe, PDR}}

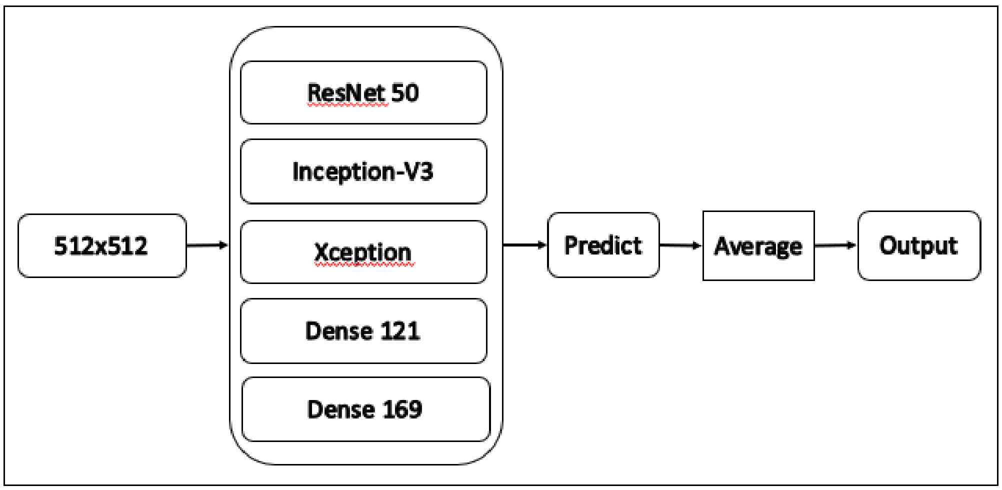


Fig 14

Here we have divided our dataset into 5 different classes, namely: -

Class 0: - Normal (No DR)

Class 1: - Mild DR

Class 2: - Moderate DR

Class 4: - Severe DR

Class 5: - PDR

And then we check the accuracy, precision, F1 score and recall values for each of the classes so that we can see in which class out model is the most accurate.

For the learning rate, we have considered three different values while two optimizers are considered. Fig 14 here shows five architectures as mentioned above are trained with different hyperparameters. After completion of training, all architectures are ensembled. The table here shows the accuracy, recall, precision, specificity, and F1-score of SGD and Adam optimizer with different learning rates. The learning rate is decreased from 0.01 to 1e-05. The performance of the model increases with a decrease in the learning rate. Also, it can be noted that most of the time SGD has better performances than Adam.

For finding the learning rate we make use of fastai and torchvision.

We tested three decays i.e., 1e-2,1e-4,1e-6.

Fig 15

The results we get after fine tuning of hyperparameters are as follow: -

Class	Recall	Precision	F1-Score	Specificity	Accuracy
0	0.97	0.84	0.90	0.40	96.12%
1	0.80	0.51	0.15	0.99	91.47%
2	0.41	0.65	0.50	0.95	86.45%
3	0.51	0.48	0.49	0.98	92.71%
4	0.56	0.69	0.62	0.99	93.54%

Comparing results with existing ensemble model [24].

	Recall		Precision		Specificity		F1-Score	
Class	Our Model	Pratt et. al[22]	Our Model	Pratt et. al[22]	Our v	Pratt et. al [22]	Our Model	Pratt et. al[22]
0	**0.97**	0.95	**0.84**	0.78	**0.40**	0.19	**0.90**	0.85
1	**0.80**	0.00	**0.51**	0.00	**0.99**	1.00	**0.15**	0.00
2	**0.41**	0.23	**0.65**	0.40	**0.95**	0.93	**0.50**	0.29
3	**0.51**	0.78	**0.48**	0.52	**0.98**	0.99	**0.49**	0.10
4	**0.56**	0.44	**0.69**	0.32	**0.99**	0.97	**0.62**	0.37

Hyperparameters used are-

S.no	Model	Layers	Batch size	Momentum	Epoch	Learning Rate	Optimizer
1	Res-Net 50 Inception v3 Xception DenseNet121 DenseNet169	168 159 126 121 169	8	0.9	20	0.01,0.001, 0.0001,0.00001	SGD
2	Res-Net 50 Inception v3 Xception DenseNet121 DenseNet169	168 159 126 121 169	8	0.9	20	0.001, 0.0001,0.00001	Adam

SGD: - Stochastic Gradient Descent

Adam: - Adaptive Moment Estimation

Here, we could not go past 20 epochs due to computational limitations of our GPU. Up till 20 epochs also the model was taking a very long amount of time to train, and our laptop was heating up, so we could not continue further. This was also partly due to the large size of the images that we are dealing with.

We are sure that if provided with more computational power, we can further advance the accuracy and other performance parameters of our model.

Web-App Implementation

We used HTML, CSS, and JavaScript for UI/UX and for the backend we used Flask (Python)

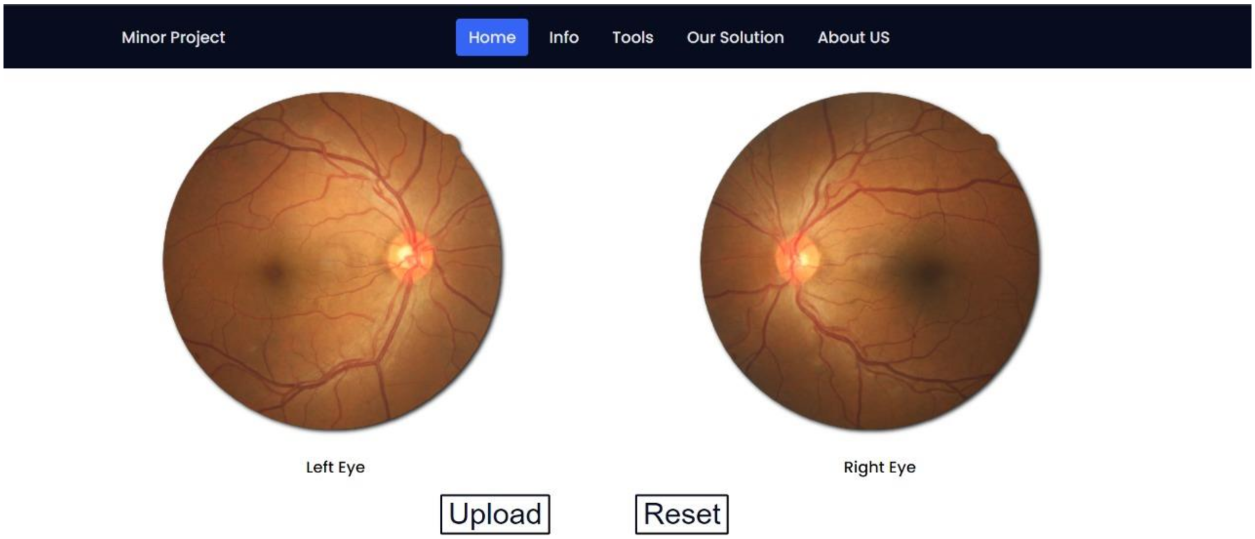


Fig 16

We upload the right and left eye photo, then the model works in the backend and gives the following output

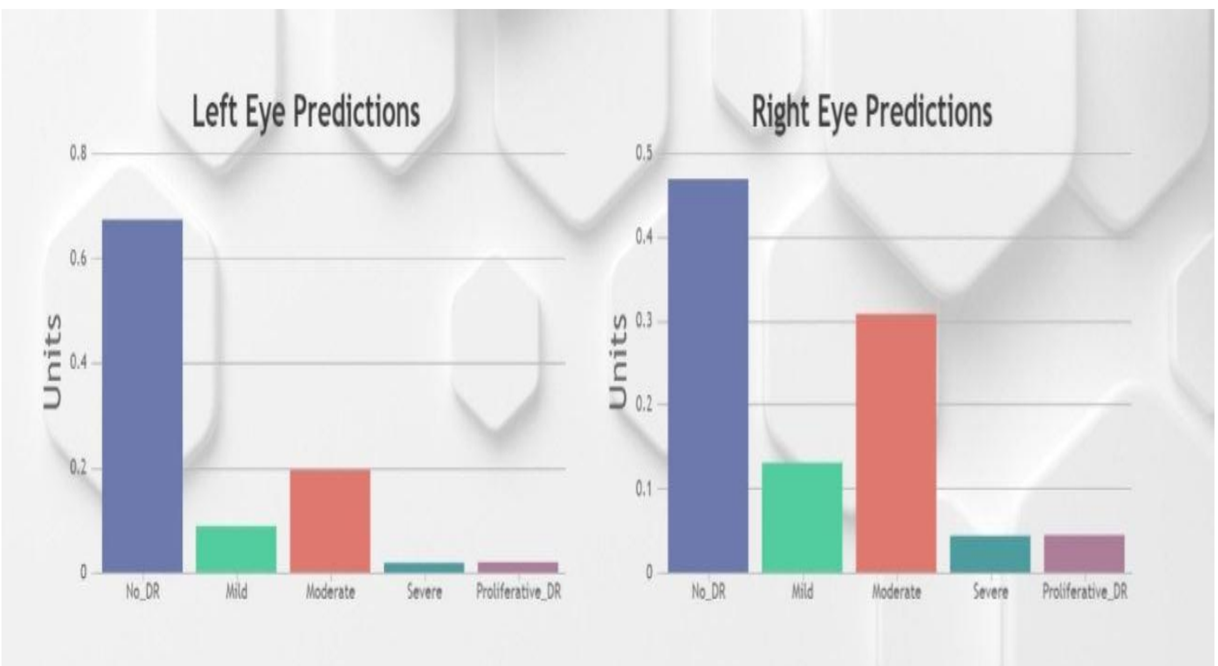


Fig 17

4. Work Done
4.1 All Topics Covered

- Exploratory Data Analysis
- Preprocessing
- Model -1 (EyeNet)
- Model -2 (Many pre-existing models such as ResNet, AlexNet, GoogleNet, VggNet)
- Model -3 (Final Model ; Ensemble Model using ResNet, Inception V3, Xception, Dense 121, Dense 169)
- Implementation of Web App (Currently On-going)

4.2 Results and Discussion

Throughout the course of our Minor Project, we researched many research papers on Diabetic Retinopathy and on its automated detection using many methods such as Deep Learning, Image Processing, using CAD, and much more. The one we found most interesting was the Deep Learning methods as it knew no bounds and could be advanced further and further over time.

We saw that there is much to be done in this field as even a single error in detection of DR due to fault in programming can lead to major blunders in real world as it is based on physical health of the people.

Considering how DR is growing at such a rapid rate and is not becoming a very common phenomena, our aim was to create something that can be used to detect DR when it is in the early stages of development. Thus, we focused on building different models to see which one can be best suitable for this purpose.

All the three models we built had their own advantages and disadvantages.

Such as the first model we built, although was built on hit and trial and was a little bit messy as compared to other pre trained models, it has its ups such as after you got the hyperparameters correct, the time it took to train was not much as compared to the future models we discussed.

Our second model consisted of all the major pre-trained models available world-wide. Although it provided satisfying result in the end, but it suffered from over-fitting, as the default parameters were giving moderate accuracy and other performance scores. After tuning it for a while it gave good parameters score but it suffered from over-fitting, and this did not work as well on unseen data

Our third and final proposed model is an ensemble model and provides good accuracy, but its disadvantage is that it takes a huge amount of time and space while training as it consists of five different models into one. Thus, the computation capability required while building this model is high and we could not go past 20 epochs due to certain limitations.

4.3 Individual Contribution of Project Members

Aditya Singh Goliya: -

- **Exploratory Data Analysis** on our dataset to check for any imbalance.
- Study of TensorFlow and how CNN work to create our **First Model.**
- Created EyeNet model and determined the number of layers and what depth each layer should have.
- Research about Ensemble models and which models should be chosen for our **Model 3.**
- Understanding and tuning the hyper-parameters used in **Model 3.**
- Formation of Web App for better accessibility.
- Making of the final Power Point Presentation.

Satvik Pandey: -

- **Pre-processing** of images to remove external noise, black images and make it so that it can be processed by our neural networks.
- Researching about the **pre-trained models** available throughout.
- Fine tuning of Hyper-Parameters on our **Second model** to get better accuracy.
- Finding out about the calculation of Learning Rate for **Model 3.**
- Researching about Adam and SGD.
- Learning how to deploy our model and integrate into our Web App seamlessly.
- Formulation of the Final Report.

5.Conclusion and Future

We used a combination of pre-processing techniques and by that our images were preprocessed quite well.

We used Crop, resize, remove black portions, and Rotate and Mirror

We proposed 3 models to get the desired results and found that every model has its own advantages and disadvantages.

Model 1

The model is built using Keras, utilizing TensorFlow as the backend. TensorFlow was chosen as the backend due to better performance over Theano, and the ability to visualize the neural network using TensorBoard.

Model 2

We used various pre trained models available on the internet and fed the data to them and following were the results.

Classification results with randomly initialized parameters and then tuned the hyperparameters to achieve greater accuracy.

Model 3

The proposed approach ensembles the five deep CNN models Resnet50, Inceptionv3, Xception, Dense - 121, Dense169

Model 3 gives us the best result which is an ensemble of different pre trained models.

Ensemble always gives good results if the models used are good and the hyperparameters chosen are appropriate.

We achieved a F1 score of 0.9 which is the highest of the 3 models.

This model predicts rightly and classifies the retinopathic image into 5 classes i.e., Normal, Mild, Moderate, Severe, PDR.

We have also created a web app on which one can upload the eye image and get the results instantly. Our model 3 is used in that web app and it is quite fast and gives accurate results.

5.1 Future Work

Our future-plan includes: -

- Further tuning of hyper-parameters to get better accuracy while dealing with class 2 (Moderate DR cases)
- Finishing the implementation of our web-app, to make our model deployable and accessible to everyone
- Further testing our model on more unseen data, to check it real-world capabilities.
- Research a bit more about ensemble models and see if there are any other model, I can add to our Model 3 i.e., Ensemble model or if we should replace any other existing model so that we can get better training time, more memory conservation and most importantly, better accuracy.

References

[1] Mookiah M., Acharya U., Chua C., Lim C., Ng E., Laude A. Computer-aided diagnosis of diabetic retinopathy: A review.

[2] Philip S., Fleming A., Goatman K., Mcnamee P., Scotland G. The efficacy of automated disease/no disease grading for diabetic retinopathy in a systematic screening program.

[3] Gardner G., Keating D., Williamson T., Elliott A. Automatic detection of diabetic retinopathy using an artificial neural network: a screening tool.

[4] Niemeijer M., Van Ginneken B., Cree M. J., Mizutani A., Quellec G., Sanchez C. I., Zhang B., Hornero R., Lamard M., Muramatsu C., et al. Retinopathy online challenge: automatic detection of microaneurysms in digital color fundus photographs.

[5] G.S. Annie Grace Vimala and S. Kajamohideen, "*Detection of Diabetic Maculopathy in Human Retinal Images Using Morphological Operations*".

[6] Sopharak, A., Uyyanonvara, B., Barman, S., and Williamson, H. T., "*Automatic 90detection of diabetic retinopathy exudates from non-dilated retinal images using mathematical morphology methods*".

[7] Walter, T., Massin, P., Erginay, A., Ordonez, R., Jeulin, C., and Klein, J. C., "*Automatic detection of microaneurysms in color fundus images*".

[8] A. V Vasilakos, Y. Tang, Y. Yao., "Neural networks for computer-aided diagnosis in medicine": a review

[9] X.W. Chen, X. Lin., "Big data deep learning: challenges and perspectives". IEEE Access, 2 (2014), pp. 514-525

[10] L. Deng, D. Yu., "Deep learning: methods and applications"

[11] M.T. Esfahani, M. Ghaderi, R. Kafiyeh., "Classification of diabetic and normal fundus images using new deep learning method".

[12] Abràmoff M. D., Reinhardt J. M., Russell S. R., Folk J. C., Mahajan V. B., Niemeijer M., Quellec G. Automated early detection of diabetic retinopathy.

[13] UR A. Decision support system for diabetic retinopathy using discrete wavelet transform. *Proceedings of the Institution of Mechanical Engineers*

[14] Antal B., Hajdu A. An ensemble-based system for microaneurysm detection and diabetic retinopathy grading

[15] Gulshan V., Peng L., Coram M., Stumpe M. C., Wu D., Narayanaswamy A., Venugopalan S., Widner K., Madams T., Cuadros J., et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs.

[16] S. Dutta, B. C. Manideep, S. M. Basha, R. D. Caytiles and N. C. S. N. Iyengar, "Classification of diabetic retinopathy images by using deep learning models".

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition"

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the inception architecture for computer vision"

[19] F. Chollet, "Xception: Deep learning with depth wise separable convolutions", Proc. IEEE Conf. Computer Vis. Pattern Recognition.

[20] G. Huang, Z. Liu, L. van der Maaten and K. Q. Weinberger, "Densely connected convolutional networks".

[21] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton University of Toronto, “ImageNet Classification with Deep Convolutional Neural Networks”.

[22] Karen Simonyan, Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”.

[23] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, “Going Deeper with Convolutions”.

[24] H. Pratt, F. Coenen, D. M. Broadbent, S. P. Harding, and Y. Zheng, "Convolutional neural networks for diabetic retinopathy", *Procedia Comput. Sci.*