Diabetic Retinopathy Detection

A Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Bachelor of Technology
in
Information Technology

by

Registration No.	Name
20155084	Rishabh Mishra
20152019	Jeste Mohit Sandeep
20158006	Vidit Bhalla
20158018	Deepti Shahi
20158004	Rajat Kumar Bernwal
20158066	Rashmi Rekha Tudu

to the

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD

April, 2018

UNDERTAKING

We declare that the work presented in this report titled "*Diabetic Retinopathy Detection*", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology, Allahabad, for the award of the ***Bachelor of Technology*** degree in ***Information Technology***, is our original work. We have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, we accept that our degree may be unconditionally withdrawn.

April, 2018
Allahabad

Rishabh Mishra
Mohit Sandeep Jeste
Vidit Bhalla
Deepti Shahi
Rajat Kumar Bernwal
Rashmi Rekha Tudu

CERTIFICATE

This is to certify that this is a bonafide record of the mini project presented by the students whose names are given in the section *Undertaking* during 2018 for the project **Diabetic Retinopathy Detection**.

April, 2018
Allahabad

(Prof A.K. Singh)
Computer Science and Engineering Dept.
M.N.N.I.T, Allahabad

Preface

Diabetic retinopathy (DR) is one of the leading causes of preventable blindness globally. Diabetic Retinopathy is an eye condition that affects people with diabetes — is the fastest growing cause of blindness, with nearly 50+ million diabetic patients at risk in India itself [4]. One of the most common ways to detect diabetic eye disease is to have a specialist examine pictures of the back of the eye and determine whether there are signs of the disease, and if so, how severe it is.

The primary motivation for us is to improve access to low-cost, high-quality medical diagnostics, and to develop a technology that positively impacts everyone.

The objective of this study was to develop robust diagnostic technology to automate DR screening. Referral of eyes with DR to an ophthalmologist for further evaluation and treatment would aid in reducing the rate of vision loss, enabling timely and accurate diagnoses.

Specifically in India where in some villages there are no doctors or the equipment needed to diagnose DR, our automated method will come handy.

Contents

Chapter 1

Introduction

This thesis presents the details of detecting diabetic retinopathy through machine learning models. Diabetic retinopathy is a medical condition in which damage occurs to the retina due to diabetes mellitus and is a leading cause of blindness.

We have used the dataset provided by kaggle ,which are high resolution images of the retina. There are two images for each individual, one of the left and other of right retina.

1.1 Motivation

Diabetic retinopathy is the leading cause of blindness in the working-age population of the developed world. It is estimated to affect over 93 million people[5]. Diabetic Retinopathy (DR) is one of the most frequent causes of visual impairment in developed countries and is the leading cause of new cases of blindness in the working age population. Altogether, nearly 75 people go blind every day as a consequence of DR [6]. Effective treatments for DR require early diagnosis and continuous monitoring of diabetic patients, but this is a challenging task as the disease shows few symptoms until it is too late to provide treatment. Currently, diagnosis of DR is performed by manual evaluation of retinal images by expert clinicians who identify presence of lesions in the eye such as micro-aneurysms (red lesions), hemorrhages and exudates (bright lesions).This turns out to be a slow and demanding process. Further, the

expertise and equipment required for such evaluation may be lacking in many areas with a large DR affected population. It is evident that an automated system for DR detection of color fundus images could have a huge impact in making timely treatment accessible to more patients. Towards this end, we have explored machine learning techniques to automatically detect the severity of DR using information from retina images in this work.We try to classify the data in five classes based upon the severity of the DR in the eye

1.2 Classes of Diabetic Retinopathy

1. Class 1(Mild DR)
    - About 1 micro aneurysm, exudate or dot hemorrhage.
2. Class 2(Moderate DR)
    - About 5-6 micro aneurysm, exudate or dot and blot hemorrhage.
    - Cotton wool spots.
    - Venous Beading.
3. Class 3(Severe DR)
    - Characterized by 4-2-1 rule which state: diffuse intraretinal hemorrhages and micro-aneurysms in 4 quadrants, venous beading in $\geq$ 2 quadrants, or IRMA in $\geq$ 1 quadrant.
    - Cotton wool spots.
4. Class 4(PDR)
    - Neovascularisation of disk.
    - Neovascularisation elsewhere.
5. Class 0(Healthy eye)
    - None of the above symptoms. [7]

1.3 Dataset

We have used the dataset provided by kaggle ,which are high resolution images of the eye and we have tried to identify defects in the eye using each individual pixel of the image in order to classify the image. The following table represents the percentage of each of the classes present in the data[5].

Class	Name	Number of images	Percentage
0	Normal	25810	73.48%
1	Mild NPDR	2443	6.96%
2	Moderate NPDR	5292	15.07%
3	Severe NPDR	873	2.48%
4	PDR	708	2.01%

Chapter 2

Related Work

Kaggle conducted a competition on the same problem in 2015 . All the top-performing Kaggle teams used sophisticated neural network models that require many days of training on high-end GPUs. Most commonly used algorithms were CNN and the top performing teams were able to get an accuracy of upto 85% and they used certain image processing techniques to improve their accuracy. The team from Stanford University tried to focus on trying to come up with simpler innovative method that can give comparable results. We focus on pre-processing the images and feature engineering using traditional image processing techniques and utilise classifiers to solve the multi-class classification problem.

Chapter 3

Proposed Work

3.1 Data Prepossessing

Each image contained a black background apart from the retina fundus. That black background was of no use to the algorithm. Hence it had to be removed. This reduced the size of the image which made processing faster. Then we used **Adaptive Histogram Equalization(CLAHE)** to make the blood vessels,exudates and hemorrhage more prominent. Then before applying any algorithm the data was re-sized according to the needed specifications. The dimensions of an image used were

- $64 \times 64 \times 3$ pixels
- $128 \times 128 \times 3$ pixels
- $224 \times 224 \times 3$ pixels
- $299 \times 299 \times 3$ pixels

3.2 Algorithms used

The following approaches were considered by the team to solve the problem of **Classification of Diabetic Retinopathy**

- Logistic Regression
- Support Vector Machine
- Convolutional Neural Networks
  - Transfer Learning using ResNet50
  - Transfer Learning using InceptionV3
  - ResNet50
- Deep Neural Network

Chapter 4

Experimental Setup and Results Analysis

4.1 Pre-processing the Data

The images we had were high resolution images with lots of useless information in them. Hence it was necessary to preprocess the images to remove this useless information and also to enhance certain features of the image for better results. The following steps were done while pre-processing the data:

1. *Removing black border*:The retina fundus was surrounded by a black border which was removed by us as it was not useful to the algorithm.

2. *Contrast Limited Adaptive Histogram Equalization(CLAHE):* Adaptive histogram equalization is a computer image processing technique used to improve contrast in images. It differs from ordinary histogram equalization in the respect that the adaptive method computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image. It is therefore suitable for improving the local contrast and enhancing the definitions of edges in each region of an image. CLAHE helped us to make the exudates,hemorrhages,blood vessels and micro-aneurysms more prominent.

3. Finally according to the requirement of the algorithm the image was resized to $64 \times 64, 128 \times 128, 224 \times 224$ or 299×299

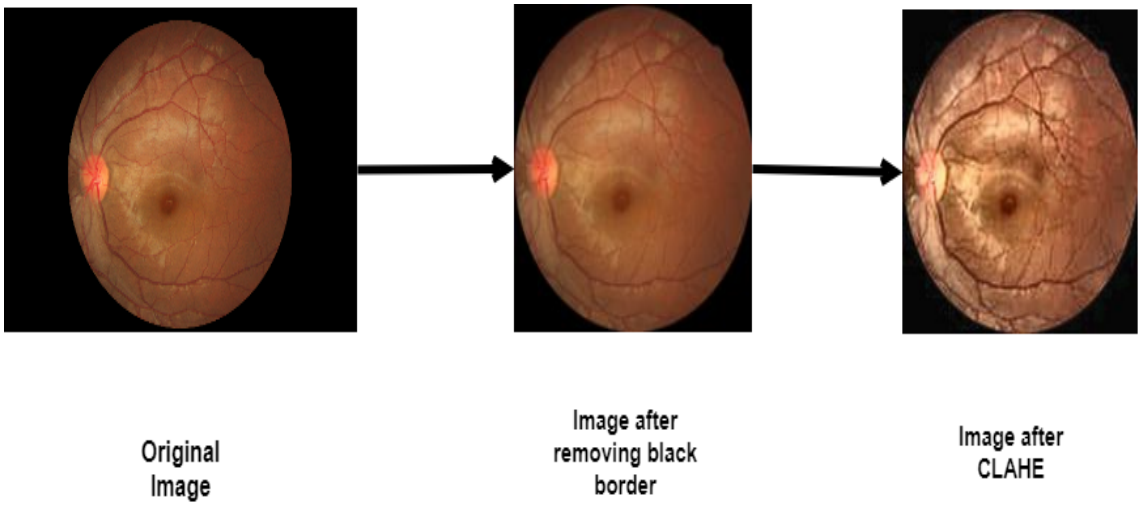

Figure 1: Stages of preprocessing

4.2 Logistic Regression

We begin by trying a simple logistic regression over preprocessed $64 \times 64 \times 3$ images. Here we used very trivial method by training our logistic regressor over the intensity values of images for each pixel.

We reshaped the $64 \times 64 \times 3$ to a flat $12288 (= 64 \times 64 \times 3)$ vector. Using this as feature vector for each image we trained our classifier. Though, we didn't get very high accuracy (around 59%) but this worked as a benchmark for rest of our classifiers.

4.3 Support Vector Machine

Here also we tried the same approach as in Logistic Regression. But here we were able to improve our accuracy (around 70%) as SVM would have been able to fit better hyper-plane as expected.

We used the SVM with default parameters though we tried various parameter tuning techniques but that didn't help much.

4.4 Deep Neural Network

The neural network we used has an architecture as shown below. After prepossessing the image is converted to a numpy array of size (m,12288/49152). Then this data is split into the training set and test set. The training labels from the file trainLabels.csv are loaded in a numpy array. The test data is fed to the input layer. Using 64×64 image then we would have an input layer of 12288 neurons. This would be because each image has 64×64 pixels over the 3 planes ie RGB. Similarly if we use an image of 128×128 we would have an input layer of 49152 neurons

The **learning rate used by us is 0.0001 i.e** α. The neural network runs for 1500 epochs with a mini batch size of 32. Then the following steps take place:

1. Placeholders are created. Placeholders take any input we wish to provide in TensorFlow. We have created placeholders for the training data,training labels and the dropout probability.

2. Then the parameters of the neural network are initialized. The weights are initialized using Xavier Initializer. This is better than random initialization of weights. It initializes weight between 0 to $\sqrt{\frac{2}{n^l+n^{l-1}}}$ where n^l is the number of neurons in layer l and n^{l-1} is the number of neurons in layer $l-1$ The bias parameters have been assigned to 0.

3. Then comes the forward propagation step. We have used **Leaky ReLu activation** for all layers but the last.

4. After that the cost of the epoch is calculated. The output from the forward_propagation function is fed to the compute_cost where the cost is computed using a **softmax function**. **L2 regularization** is applied here with a regularization parameter $\beta = 0.01$

5. TensorFlow takes care of back-propagation by itself. Finally we use **Gradient Descent with Adam Optimizer** to tune the parameters.

Here are the results of the Deep Neural Network:

Data set	Accuracy
Training data(128×128 images)	74.23%
Test data(128×128 images)	72.3%
Training data(64×64 images)	73.65%
Test data(64×64 images)	70.30%

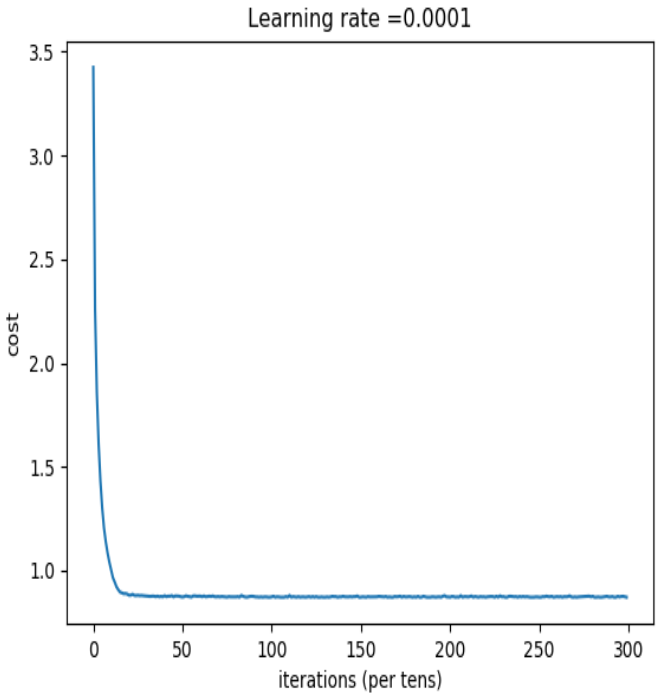

Figure 2: Cost vs Number of epochs for 128×128 images

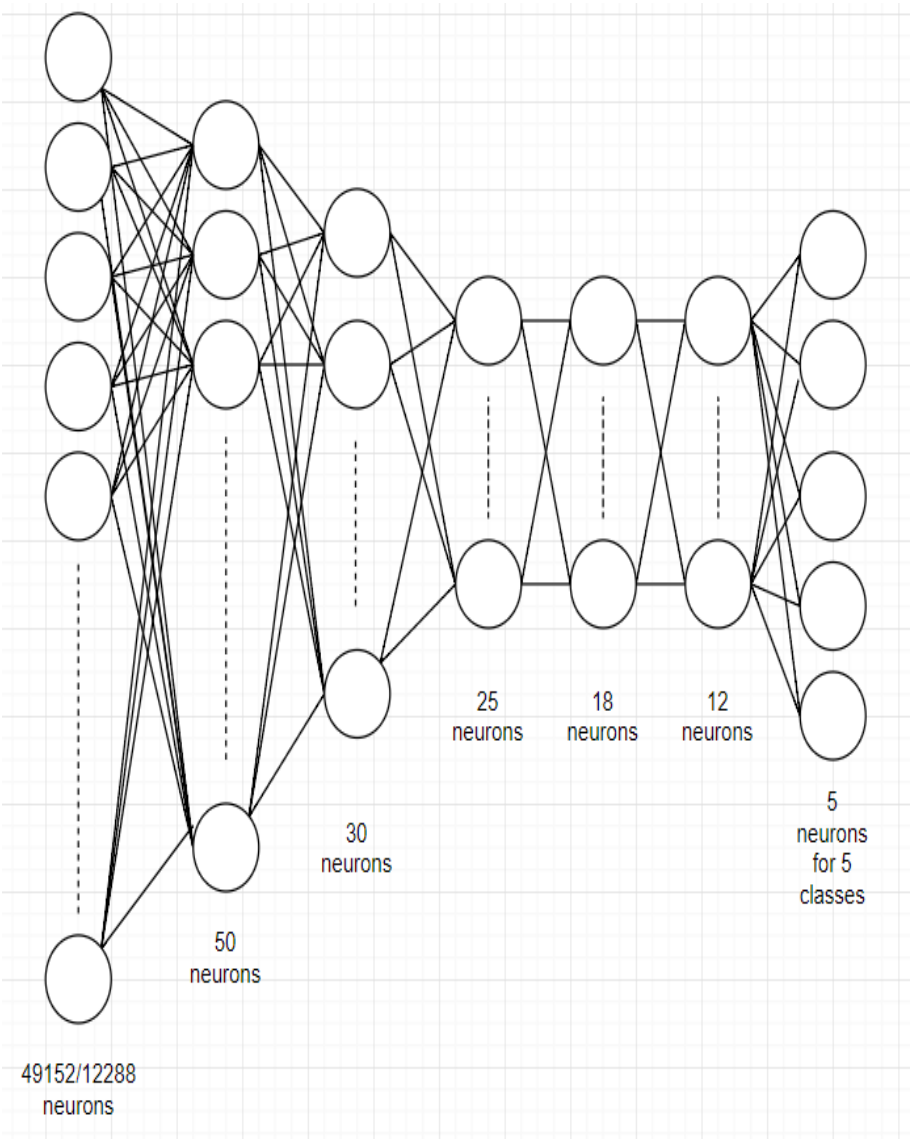

Figure 3: Deep Neural Network

4.5 Convolution Neural Network

[1]

We mainly tried two CNN models :

- ResNet50

- InceptionV3

We focused mainly on transfer learning rather than creating the whole model from scratch as the these models have already been trained rigorously which requires much more hardware support.

4.5.1 ResNet50

We did transfer learning using standard ResNet50 CNN architecture with pre-trained ImageNet weights. As the standard ImageNet weights classify the objects into 1000 categories so we excluded the top layer and added some layers of our own in the model. The standard input image size for ResNet50 is $224 \times 224 \times 3$. We used ResNet50 model in two ways :

- *As Classifier :*

Here we removed the top layer and then added following layers to the original model :

- Flatten Layer
- Softmax Layer (with 5 classes)

We freezed the original layers and trained only the last layer. The method works as it is found that the kind of information needed to distinguish between all the 1000 classes in ImageNet is often also useful to distinguish between new kinds of images (fundoscopic retinal images in our case).

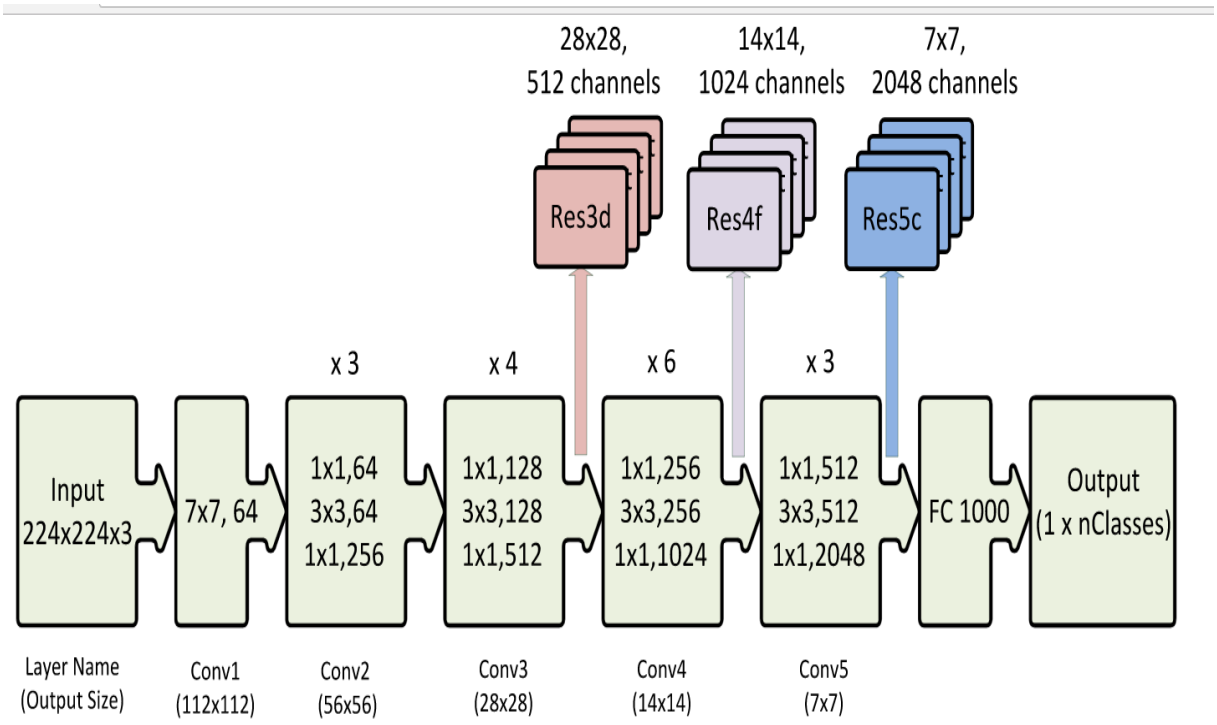

Figure 4: ResNet50 architecture

- *Fine Tuning :*

Here we removed the top layer and then added following layers to the original model:

- Flatten Layer
- Fully Connected Layer (512 neurons and relu activation)
- Dropout Layer(50 percent dropout and relu activation)
- Fully Connected Layer (256 neurons and and relu activation)
- Softmax Layer (with 5 classes)

Here we freezed the original layers and trained only the last 5 layers that we added to make the model adapt our input.

We used **adam optimizer** and **categorical crossentropy** as loss function in both cases. We used categorical crossentropy as we fed the labels in **one-hot encoded** format. The model is trained for **20 epochs** and a batch size of 32 with 8000 training and 2000 validation samples in each epoch.

4.5.2 InceptionV3

Here also we used standard InceptionV3 CNN architecture with pre-trained ImageNet weights. Images are resized to $299 \times 299 \times 3$ as required by the standard model. We customized the architecture by removing the top layer and adding a **Softmax Layer** with 5 output classes.

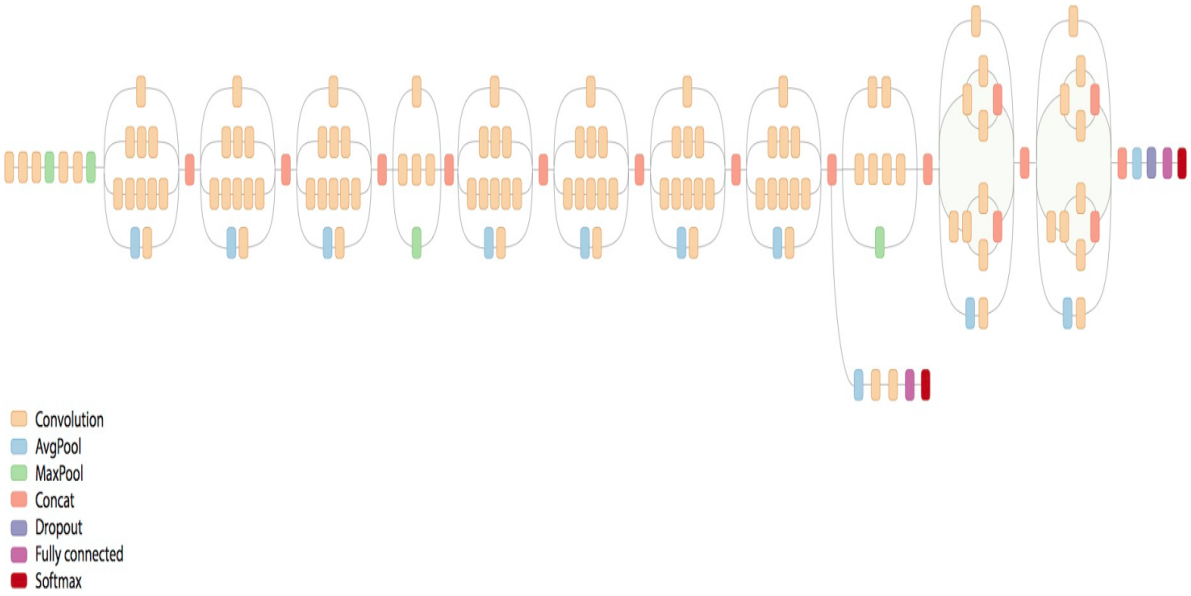

Figure 5: InceptionV3 architecture

4.5.3 Results of CNN

Model	Training Accuracy	Test Accuracy
ResNet50 without Fine tuning	72.29%	71.45%
ResNet50 with Fine tuning	74.74%	73.28%
ResNet50 without ImageNet weights	75.89%	75.49%
InceptionV3 without Fine tuning	77.83%	76.17%

Chapter 5

Conclusion and Future Work

Across all models, using CLAHE for histogram equalization improved the accuracy by a great difference (10%). This is probably due to it made the determining features like exudates, hemorrhage, blood vessels etc. more prominent.

We obtained our best results using the **InceptionV3** model with ImageNet pre-trained weights and a softmax layer with 5 output units at the end, training the additional layer and the top one block of the the base model.

As there are very less images for the classes 3 and 4 so our model got less information about these classes. Therefore, either we need more data for these classes or we need to augment the data. Augmentation is flipping, rotating, altering brightness, zooming etc. to increase the samples artificially.

One of the challenges of the project was the limitations of the computational resources we had available. Deep CNN models take a long time to train, and the resources necessary to train them become expensive over time. There are multiple network architectures and preprocessing/data augmentation techniques that could have been explored had there been more computational resources available. Because of these time and computational constraints, there was a tradeoff between training time and accuracy.

Our next goal is to combine different classifiers using ensemble approach like **AdaBoost-CNN** referred as ACNN. Also, we are not using whole dataset available for CNN training as it will be very resource intensive to run on a single core. So,

we will write our scripts such that they may utilise all the available cpu/gpu cores.

References

[1] Alex Tamkin, Iain Usiri and Chala Fufa. *Deep CNNs for Diabetic Retinopathy Detection*, Stanford University, 2016.

[2] Sagar Honnungar, Sanyam Mehra and Samuel Joseph. *Diabetic Retinopathy Identification and Severity Classification*, Stanford University, 2016.

[3] Marco Alban, Tanner Gilligan. *Automated Detection of Diabetic Retinopathy using Fluorescein Angiography Photographs*, Stanford University, 2016.

[4] Gargeya R1, Leng T. *Automated Identification of Diabetic Retinopathy Using Deep Learning*, Stanford University School of Medicine, Palo Alto, California, 2017.

[5] Kaggle
`https://www.kaggle.com/c/diabetic-retinopathy-detection`

[6] Hackernoon Contest
`https://hackernoon.com/diabetic-retinopathy-detection-9bdceac75752`

[7] DR Classes
`https://webeye.ophth.uiowa.edu`

[8] Machine Learning,
`https://www.coursera.org/learn/machine-learning/home/welcome`

[9] Neural Networks and Deep Learning,
`https://www.coursera.org/learn/neural-networks-deep-learning/home/welcome`

[10] Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization,
https://www.coursera.org/learn/deep-neural-network/home/welcome

[11] Convolutional Neural Networks,
https://www.coursera.org/learn/convolutional-neural-networks/home/welcome

[12] Structuring Machine Learning Projects,
https://www.coursera.org/learn/machine-learning-projects/home/welcome

[13] Structuring Machine Learning Projects,
https://www.coursera.org/learn/machine-learning-projects/home/welcome

[14] Keras Documentation,
https://keras.io/

[15] Tensorflow Documentation,
https://www.tensorflow.org/get_started/

[16] Python Documentation,
https://docs.python.org/3/

[17] Data Science Stackexchange,
https://datascience.stackexchange.com/

[18] StackOverflow,
https://stackoverflow.com/