# Diabetic Retinopathy Prediction using Convolutional Neural Networks

Rajib Das Bhagat (CS17M034),
Guide: Prof. Balaraman Ravindran
Department of Computer Science and Engineering,
Indian Institute of Technology, Madras

**Abstract**

Diabetic Retinopathy can be termed as any damage caused to the retina of an eye, which causes vision impairment to the people suffering from diabetes. Diabetic Retinopathy refers to retinal vascular disease, or damage to the retina caused by abnormal blood flow. The identification of features related to abnormalities in the images taken from a diabetic patient correctly is troublesome and time-consuming. The aim of this study is to explore how we can automate the process of correctly classifying and predicting DR. The classification can be performed and the correctness can be studied based upon its sensitivity, specificity and accuracy. The network was trained using GPU-cluster, which is equipped with high-end graphics processing unit (GPU), providing high computational power. This report is based on binary classification as abnormal and normal.

**Index Terms**

Diabetic Retinopathy, Convolutional Neural Network, Retinal Images, Diabetes, Image Pre-processing, Feature Extraction, Image Classification, Eye.

## I. Introduction

DIABETIC RETINOPATHY, as stated earlier is an eye disease. Diabetic retinopathy has no early visible symptoms. According to a medical report, a person suffering from diabetes for 20 years or more have high chances of vision loss, due to diabetic retinopathy [11]. The complications occur when people with high blood sugar levels causing blood vessels damage in the retina. This complication, can lead to swelling and then leakage of blood vessels which can lead to permanent vision loss.

### *A. Eye Structure and Diabetic Retinopathy*

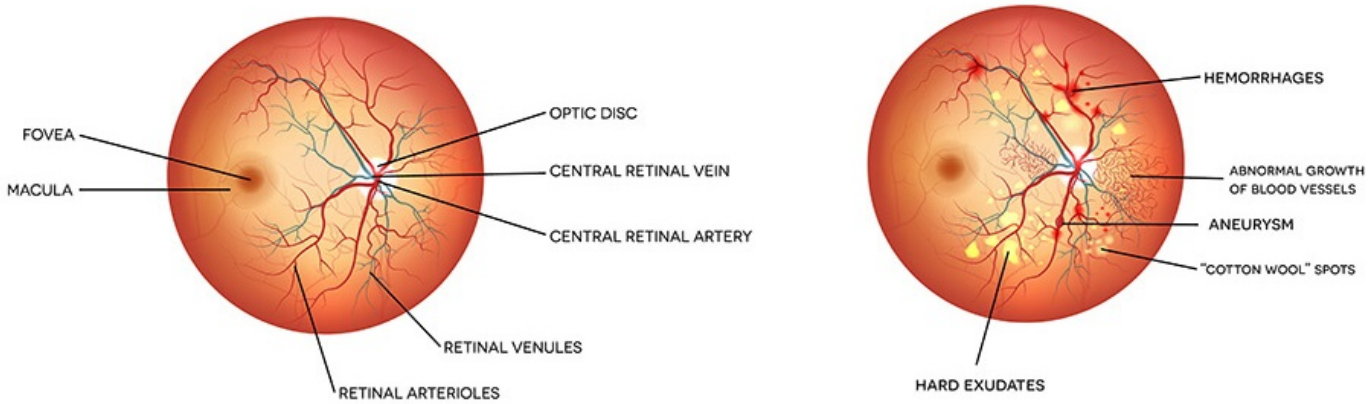
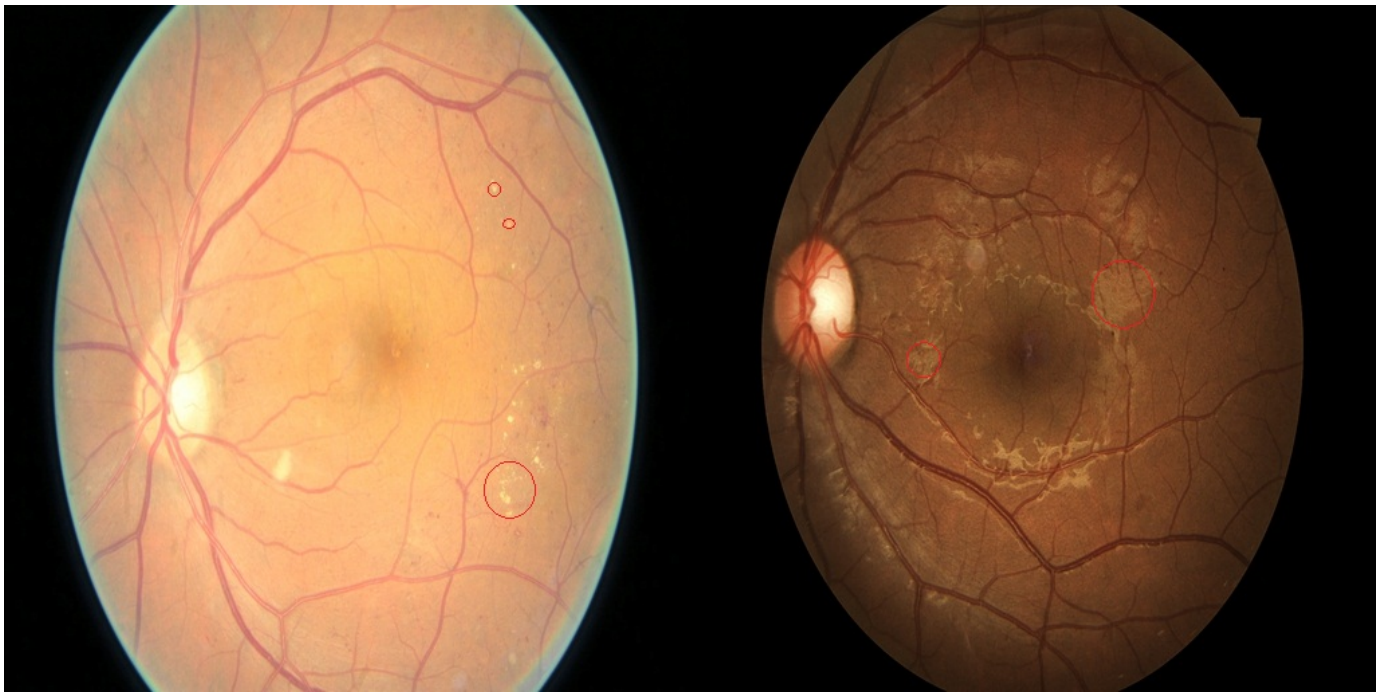
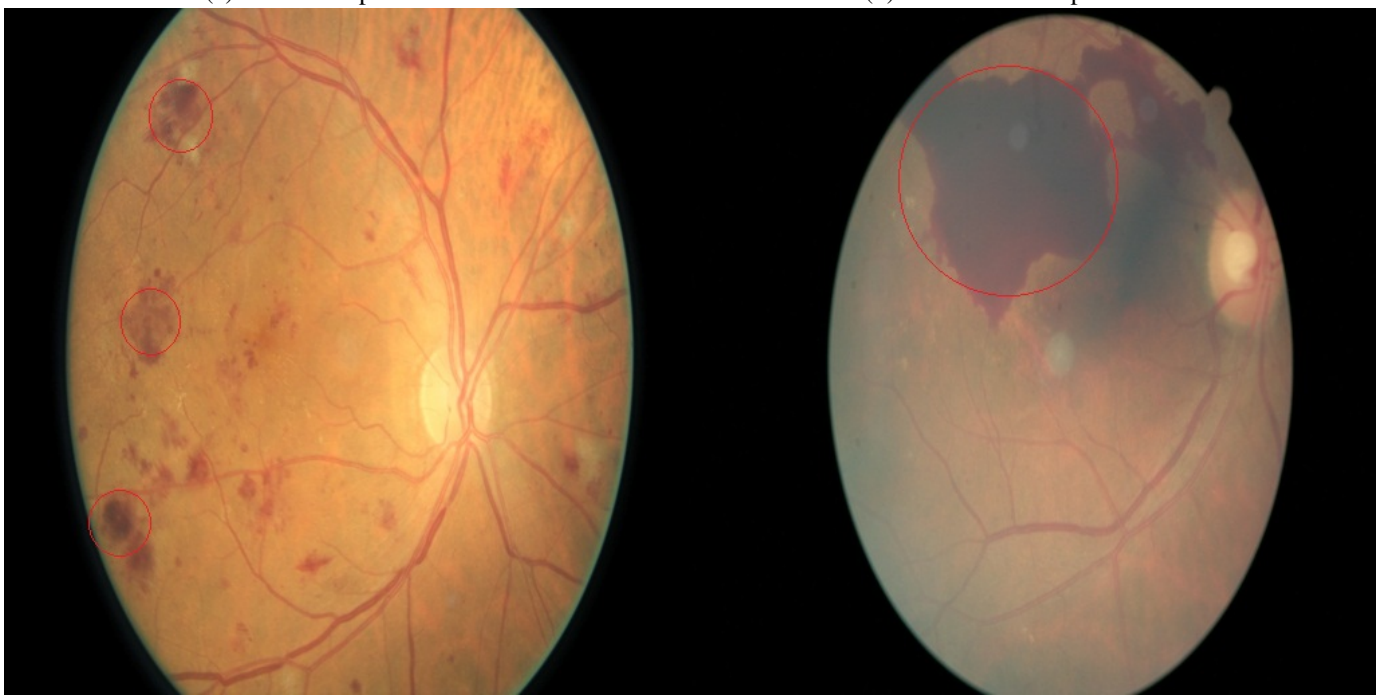



Fig. 1: The structure of a normal eye (left) and related abnormalities (right).



(a) Mild Non-proliferative (b) Moderate Non-proliferative



(c) Severe Non-proliferative (d) Proliferative

Fig. 2: Different stages of Diabetic Retinopathy (related abnormalities are circled).

The various abnormalities related to DR are shown in Fig. 1 (right). In this report, section I includes the introduction with a brief explanation of on eye structure and DR classification and convolutional neural network. Section II presents the literature survey in the related field. Section III highlights the methodology used, while section IV previews the results and section V concludes with future work.

Based on DR stages [2] (Table I), the abnormalities in the fundus images can be classified as non-proliferative and proliferative.

TABLE I: Retinopathy stages and lesions

| No | Type | Stage | Size | Shape | Color | Others |
|---|---|---|---|---|---|---|
| 1 | Micro-aneurysm | mild non-proliferative | tiny | round | darkish red | |
| 2 | Soft Exudate | moderate non-proliferative | small to medium | oval | whitish | edges blur |
| 3 | Hard Exudate | severe non-proliferative | varies | irregular | yellow | clear edges |
| 4 | Hemorrhage | proliferative | varies | dot or flame like | darkish red | |
| 5 | Neovascularization | proliferative | varies | varies | red | new blood vessels |

*i. Normal:* An eye without any presence of abnormal lesions or any other type of defects is defined as a normal eye (Fig. 1 [left]).

*ii. Mild non-proliferative:* The earliest visible detectable lesions are micro-aneurysms. Micro-aneurysms are usually round, red, tiny dots, which may leak fluid into the retina. Micro-aneurysms may occur in a group or in isolation, this is classified as mild non-proliferative (Fig. 2a).

*iii. Moderate non-proliferative:* As mild diabetic retinopathy progresses, the blood vessels that nourish the retina may swell or distort a little. Blood vessel losses their ability to transport blood. The lesion present in this is known as soft exudates. The lesion is like cotton wool spots or micro-infarctions. Soft exudates are small, white lesions not clearly visible and are termed as moderate non-proliferative (Fig. 2b).

*iv. Severe non-proliferative:* Hard exudates lesion present and are leaked lipid from the weakened blood vessels. Lesions are normally yellow colored and clear edges are waxy alike. Hard exudates tend to accumulate in groups. This stage is severe non-proliferative (Fig. 2c).

*v. Proliferative:* When micro-aneurysm ruptures, leakage of blood occurs from the blood vessels. This is known as hemorrhages. Hemorrhages are the red dot, flame-like presence in cluster or rings. New blood vessels also tend to develop, which are weak and gets tear off easily. The bleeding can cause permanent vision loss. This occurs due to lack of oxygen and known as neovascularization. This presence of hemorrhage and neovascularization is the final stage of diabetic retinopathy and is termed as proliferative (Fig. 2d).

### *B. Convolutional Neural Network (CNNs)*

A neural network, receives input, changes their activation according to the input feed, and produces output depending on the input and activation. Series of steps involved are feed-forwarding the input values and back-propagation with updating weights and bias. A CNN comprises of convolution layers, pooling layers and fully connected layers. A convolutional layer applies an operation known as a convolution operation to the input and pass the result to the next layer. Each neuron processes data for its local receptive fields. Taking training data, feed forwarding in the neural network which is known as a convolutional neural network for training and get its minimized weights and bias. And, the accuracy is calculated. A convolutional neural network is used in various applications such as image and speech recognition and natural language processing including medical imaging too.

The working steps for convolutional neural network architecture are described as:

*i. Local Receptive Fields:* Input size of n x n neurons is fed into a CNN. These n x n neurons are termed as input pixels. Input pixels are connected to the first hidden layers comprising of many hidden neurons. Here, every input neurons are not connected to every hidden neuron. Generally, only a few connections are made in small. These small connections are based on localized regions for the input neurons.

The localized region for the input neurons is the local receptive field for the hidden neurons. This is like a

small window on the input neurons present. Thus, the connections from input neurons to hidden neurons represents both weights and biases. The local receptive field is slided and the output is different for different hidden neurons. The sliding of the local receptive field is done in accordance with stride length. Stride can be termed as the number of pixels need to be slide for the small window present. If stride length is 1, 2, ..., n; the local receptive field can be slide by 1, 2, ..., n neurons respectively.

*ii. Filters:* Filter is the mapping of the input layer to the next hidden layer. The weights present in the filter is termed as shared weights, while the bias is termed as shared bias. This shared weights and bias is defined as a kernel or filter. Filters are used to blur images, sharpen images, and also performs edge detection. In CNN, filters are not pre-defined. The weights in each filter is learned as the training process is carried on.

*iii. Pooling:* Pooling layers are used after each and every convolution layer. Pooling technique combines the neuron cluster output at one layer into a single neuron in the next layer. The feature use of pooling is to simplify the output from the convolutional layer. Pooling progressively reduces the number of parameters and the huge number of computation associated with CNN. The simplification is done to generate a compressed model of the feature map. The pooling techniques available are max-pooling, average-pooling and l2-pooling.

### *C. Issues in convolutional network*

Neural networks are basically hard to train, because of many known issues involve. Few of them are described here. While the model starts to learn, after some time it happens that the CNN stops learning. This issue is termed as neuron saturation. Neuron saturation is a huge problem in neural networks. This neuron saturation can be solved by using a proper choice of activation function such as ReLu or LeakyReLu.

The second issue in the neural network is overfitting. Overfitting can be defined as an error in training data when the model fit to the noise, but unable to generalize to new data. The error tends to be huge. A general rule to prevent overfitting is reducing the size of the network, increasing the number of training data and while training instead of using test data, validation data is used. Accuracy can be computed at the end of each epoch on the validation data itself. If the neuron saturates the training process is stopped. This scheme is known as early stopping. Regularization too helps in overcoming the overfitting issue. Types of different regularization are L1, L2 or mixed regularization; which is used in addition to cost function.

Other known issues, such as vanishing gradient problem [10] and exploding gradient problem [10] do also exist. Neurons learn much slowly in the earlier layer in compared to neurons in the later layers. This is the vanishing gradient problem. In converse to this, is exploding gradient problem. Last, but not the least neural networks sometimes learns at a different pace and is known as an unstable gradient problem.

## II. Related Work

Correctness in screening for diabetic retinopathy is the first and foremost important aspect for further treatment. The identification accuracy is important for both cost and treatment effectiveness.

As stated earlier, a diabetic retinopathy classification for a physician is highly time-consuming. Giving an alternate option for real-time classification is much profound for an available training data. Moreover, much work has been done on the classification of abnormal and normal diabetic retinopathy. The techniques used earlier were related to machine learning such as k-nearest neighbors (K-NN) classifiers and support vector machine (SVM) techniques were used.

Numerous amount of research work (Table II) has been carried out, and still in progress on the types of classification of Diabetic Retinopathy. Diabetic Retinopathy classification such as binary classification, three-class classification and five-class classification. The related work done previously can be seen in Table II.

TABLE II: Literature Survey

| | Authors | Paper | Year | Classification Type | Techniques Used | Accuracy |
|---|---|---|---|---|---|---|
| 1 | G G Gardner, D Keating, T H Williamson, A T Elliott | Automatic detection of diabetic retinopathy using an artificial neural network:a screening tool [1] | 1996 | Binary | Neural Network | 88.4% |
| 2 | Nayak J, Bhat PS, Acharya R, Lim CM, Kagathi M | Automated identification of diabetic retinopathy stages using digital fundus images [3] | 2008 | Three class | Neural Network | 93.0% |
| 3 | Acharya UR, Chua CK, Ng EY, Yu W, Chee C | Application of higher order spectra for the identification of diabetes retinopathy stages [4] | 2008 | Five Class | SVM | 82.0% |
| 4 | Acharya UR, Lim CM, Ng EY, Chee C, Tamura T | Computer-based detection of diabetes retinopathy stages using digital fundus images [5] | 2009 | Five Class | SVM | 85.9% |
| 5 | P. Adarsh, D. Jeyakumari | Multiclass SVM-based automated diagnosis of diabetic retinopathy [6] | 2013 | Five Class | SVM | 96.0% |

From Table II, it can be clearly seen that much of the works have been carried out using the SVM technique. This technique requires feature extraction methods before being feed into the SVM classifier. For five-class classification, features such as micro-aneurysm, exudates, hemorrhages were used [5]. A very small number of the dataset was used for training, testing or validation purposes. For the real-time application, a convolutional neural network is much suitable in terms of classification and prediction.

## III. Methodology

The convolutional neural network architecture [8] used is shown in Table III, this similar architecture was used by Harry Pratt. The architecture comprising of 10 convolutional layers and 3-fully connected layers. The input layer was feed with size 512 x 512 neurons. Each convolution layer is followed by a leaky rectified linear unit (LeakyReLu) activation function, along with batch normalization. For pooling layers, a max-pooling technique is used with kernel size 3 x 3 and 2 x 2 strides. A kernel size is basically an odd number. Before adding to the fully-connected layer, the network from the final convolutional layer was flattened to one dimension.

Hyper-parameters was selected based on trial and error method. For all the known issues as stated earlier, hyper-parameters and other settings were done accordingly. The loss function, entropy cost function was used to address the saturation problem. Similarly, L2 regularization for weights and bias initialization and dropouts on the dense layer was used to avoid overfitting. In the final hidden layer, a softmax activation was used.

The classification is binary classification as abnormal and normal. And, many of the other related parameters were used in this convolutional neural network. The total number of trainable and non-trainable parameters are 7,863,554 and 2,040 respectively.

TABLE III: Convolutional Neural Network Architecture

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d_1 (Conv2D) | (None, 32, 512, 512) | 320 |
| leaky_re_lu_2 (LeakyReLU) | (None, 32, 512, 512) | 0 |
| batch_normalization_1 (BatchNormalization) | (None, 32, 512, 512) | 2048 |
| max_pooling2d_1 (MaxPooling2) | (None, 32, 256, 256) | 0 |
| conv2d_2 (Conv2D) | (None, 32, 256, 256) | 9248 |
| leaky_re_lu_3 (LeakyReLU) | (None, 32, 256, 256) | 0 |
| batch_normalization_2 (BatchNormalization) | (None, 32, 256, 256) | 1024 |
| max_pooling2d_2 (MaxPooling2) | (None, 32, 128, 128) | 0 |
| conv2d_3 (Conv2D) | (None, 64, 128, 128) | 18496 |
| leaky_re_lu_4 (LeakyReLU) | (None, 64, 128, 128) | 0 |
| batch_normalization_3 (BatchNormalization) | (None, 64, 128, 128) | 512 |
| max_pooling2d_3 (MaxPooling2) | (None, 64, 64, 64) | 0 |
| conv2d_4 (Conv2D) | (None, 64, 64, 64) | 36928 |
| leaky_re_lu_5 (LeakyReLU) | (None, 64, 64, 64) | 0 |
| batch_normalization_4 (BatchNormalization) | (None, 64, 64, 64) | 256 |
| max_pooling2d_4 (MaxPooling2) | (None, 64, 32, 32) | 0 |
| conv2d_5 (Conv2D) | (None, 128, 32, 32) | 73856 |
| leaky_re_lu_6 (LeakyReLU) | (None, 128, 32, 32) | 0 |
| batch_normalization_5 (BatchNormalization) | (None, 128, 32, 32) | 128 |
| max_pooling2d_5 (MaxPooling2) | (None, 128, 16, 16) | 0 |
| conv2d_6 (Conv2D) | (None, 128, 16, 16) | 147584 |
| leaky_re_lu_7 (LeakyReLU) | (None, 128, 16, 16) | 0 |
| batch_normalization_6 (BatchNormalization) | (None, 128, 16, 16) | 64 |
| max_pooling2d_6 (MaxPooling2) | (None, 128, 8, 8) | 0 |
| conv2d_7 (Conv2D) | (None, 256, 8, 8) | 295168 |
| leaky_re_lu_8 (LeakyReLU) | (None, 256, 8, 8) | 0 |
| conv2d_8 (Conv2D) | (None, 256, 8, 8) | 590080 |
| leaky_re_lu_9 (LeakyReLU) | (None, 256, 8, 8) | 0 |
| batch_normalization_7 (BatchNormalization) | (None, 256, 8, 8) | 32 |
| max_pooling2d_7 (MaxPooling2) | (None, 256, 4, 4) | 0 |
| conv2d_9 (Conv2D) | (None, 512, 4, 4) | 1180160 |
| leaky_re_lu_10 (LeakyReLU) | (None, 512, 4, 4) | 0 |
| conv2d_10 (Conv2D) | (None, 512, 4, 4) | 2359808 |
| leaky_re_lu_11 (LeakyReLU) | (None, 512, 4, 4) | 0 |
| batch_normalization_8 (BatchNormalization) | (None, 512, 2, 4) | 16 |
| max_pooling2d_8 (MaxPooling2) | (None, 512, 2, 2) | 0 |
| dropout_1 (Dropout) | (None, 512, 2, 2) | 0 |
| flatten_1 (Flatten) | (None, 2048) | 0 |
| dense_1 (Dense) | (None, 1024) | 2098176 |
| leaky_re_lu_12 (LeakyReLU) | (None, 1024) | 0 |
| dropout_2 (Dropout) | (None, 1024) | 0 |
| dense_2 (Dense) | (None, 1024) | 1049600 |
| leaky_re_lu_13 (LeakyReLU) | (None, 1024) | 0 |
| dense_3 (Dense) | (None, 2) | 2050 |
| Total params: 7,865,554<br>Trainable params: 7,863,514<br>Non-trainable params: 2,040 | | |

### A. Dataset:

A competition was hosted was Kaggle on the classification of DR. This dataset is the part of the competition. The image data was collected from Kaggle website (https://kaggle.com/c/diabetic_retinopathy_detection/data). The website contains a total of 88702 images for diabetic retinopathy classification and the total size of around 89 GB. The image dimensions are around 6M pixels per image. The image dimensions were really huge. The image's dimensions were re-sized to 512 x 512 pixels, in order to compensate for the memory issue. The re-sized images of testing and training data were barely 5.7 GB and 3.73 GB respectively. The dataset (Table IV), training and testing files were sorted according to respective categories as normal, mild, moderate, severe, proliferative as 0, 1, 2, 3, 4 respectively (Table V, VI).

TABLE IV: Diabetic Retinopathy Dataset

| | Count | Size | Re-sized |
|---|---|---|---|
| Test | 53576 | 53.7 GB | 5.70 GB |
| Train | 35126 | 35.3 GB | 3.73 GB |
| Total | 88702 | 89.0 GB | 9.43 GB |

TABLE V: DR as per Training Dataset

| | DR Type | Count |
|---|---|---|
| 0 | Normal | 25810 |
| 1 | Mild Non-proliferative | 2443 |
| 2 | Moderate Non-proliferative | 5292 |
| 3 | Severe Non-proliferative | 873 |
| 4 | Proliferative | 708 |
| Total | | 35126 |

TABLE VI: DR as per Testing Dataset

| | DR Type | Count |
|---|---|---|
| 0 | Normal | 39533 |
| 1 | Mild Non-proliferative | 3762 |
| 2 | Moderate Non-proliferative | 7861 |
| 3 | Severe Non-proliferative | 1214 |
| 4 | Proliferative | 1206 |
| Total | | 53576 |

For, classification in this report category (3,4) as abnormal and category (0) as normal was chosen for training and testing respectively. The reason for grouping is that in category (1,2); there were no visual symptoms available for our naked eye. Hence, in category (3,4) symptoms were clearly visible and it was grouped and chosen for training and testing purpose. Total of 3600 files chosen as training data and 400 was chosen for testing data for each category named abnormal and normal. Hence, the total of 8000 files was selected for training and testing purpose.

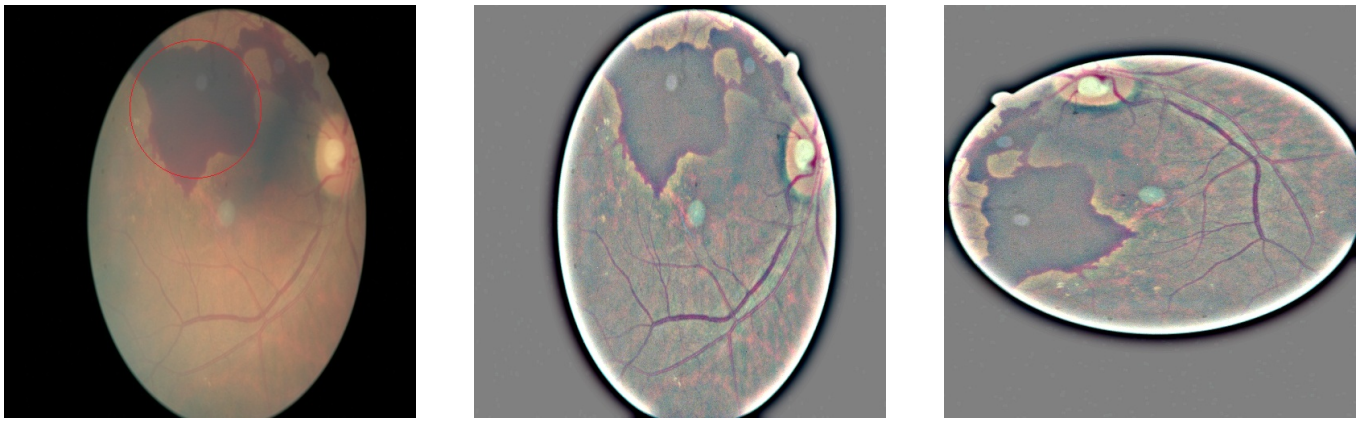
### B. Hardware and Software

For running our CNN, GPU cluster was used. The cluster has high-end GPU. Random GPU type available were used such as k20, k80, TitanX or 1080Ti. The GPU contains deep neural network libraries for machine learning and training, with TensorFlow support at the back-end. Along with GPU, keras package which is a high-level neural network API is used in order to train and test the neural network model. Keras documentation is available at https://keras.io.

*C. Pre-processing:*

The images contained a huge amount of noise. Some images were corrupted, blur, not properly focused, thus making the pixel intensity varying a lot. Lighting effects also create unnecessary distortion. Thus, Gaussian Blur color normalization (Fig. 3b) was implemented using OpenCV (https://opencv.org/) package numpy. Here, along with color normalization, the images were re-sized to 512 x 512. The pre-processing algorithm [7] is used by one of the winners in Kaggle diabetic retinopathy competition and the algorithm is stated as:

```
pre_processing(source):
    for file in glob.glob(source):
        input_image = cv2.imread(file, 1)
        resized_image = cv2.imread(input_image, (512, 512))
        gauss_blur = cv2.GaussianBlur(resized_image, (0, 0), 10)
        processed_image = cv2.addweigted(resized_image, 3, gauss_blur, -3, 128)
        return processed_image
```



(a) Original Proliferative (b) Gaussian Blur Proliferative (c) Augmented Proliferative

Fig. 3: Showing an original, pre-processed and augmented image.

*D. Augmentation:*

After pre-processing the images, the images were fed into the convolutional neural network. The pre-processed images were used for training the convolutional neural network. Later, during every epoch performed the images were randomly augmented (Fig. 3c). The augmentation performed details can be found in the Table VII.

TABLE VII: Augmentation Details

| | Attributes used | Values |
|---|---|---|
| 1 | featurewise_center | False |
| 2 | featurewise_std_nomalization | False |
| 3 | rescale | 1.0/255.0 |
| 4 | rotation_range | 90 |
| 5 | horizontal_flip | True |
| 6 | vertical_flip | True |

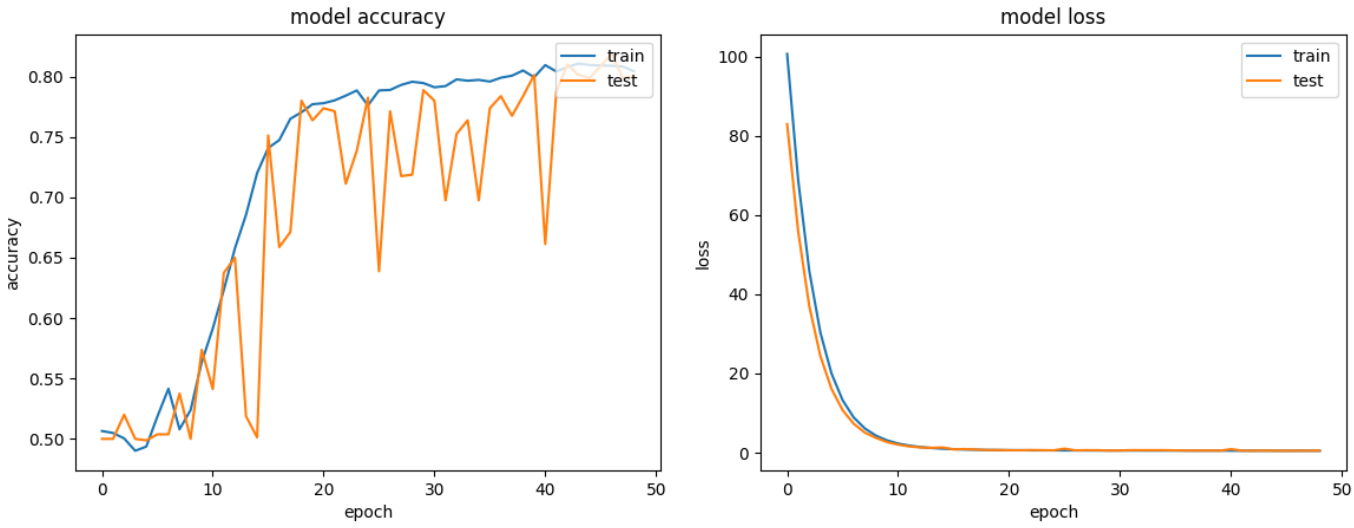
Point to be noted that the augmentation is performed on training data only and not on test data. Although, re-scaling of input images is performed both on training and testing data.

*E. Training:*

After all the set-up being constructed and done, the CNN was trained with 7200 training files and 800 validation files. The initial epochs with value 30 were used for training the neural network. This was basically done to know whether all the parameters used was capable of handling the neural network or not. It was found that the validation accuracy fluctuated a lot. Hyper-parameters was tuned and many experiments were performed for tuning and training the convolutional network. For tuning the learning rate if the cost loss is too high, then the learning rate is decreased by multiples of 10. And, if cost loss is too low, the learning rate is increased by multiples of 10. Stochastic gradient descent optimizer was used for training along with Nestrov momentum. With trial and error, some optimal result was found. Some of the basic hyper-parameters were used as per described in the Table VIII.

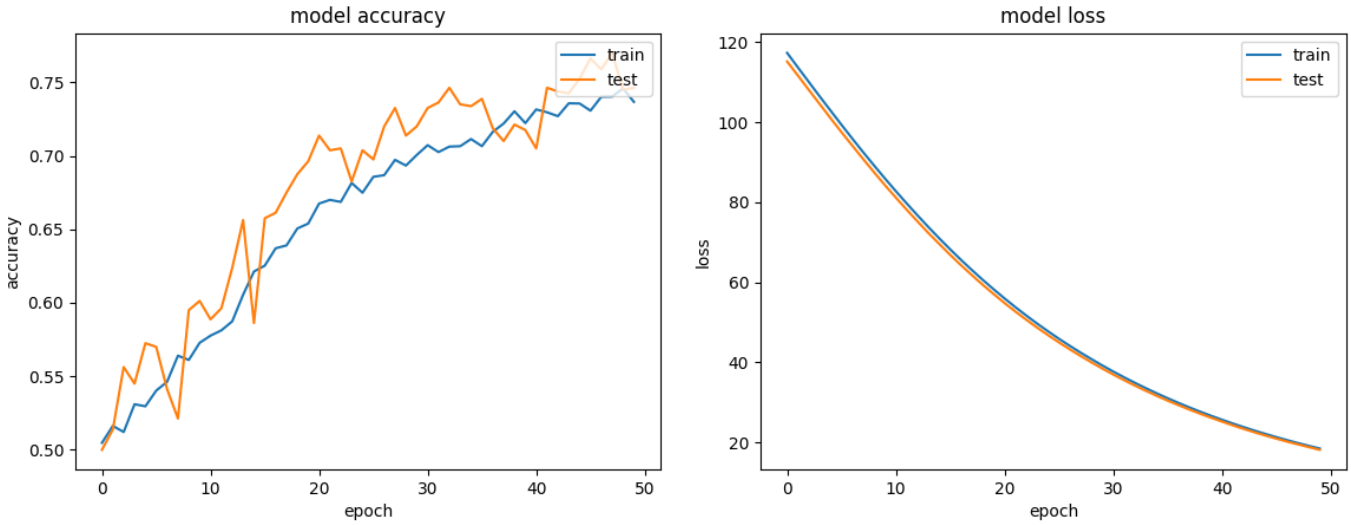
TABLE VIII: Hyper-parameters and Other Details (for different experiments)

| | Attributes Used | Experiment 1 | Experiment 2 | Experiment 3 |
|---|---|---|---|---|
| 1 | batch_size | 32 | 32 | 16 |
| 2 | learning_rate | 0.0001 | 0.00001 | 0.00001 |
| 3 | epochs | 50 | 50 | 100 |
| 4 | drop_out | 0.5 | 0.5 | 0.5 |
| 5 | input_size | 512 x 512 | 512 x 512 | 512 x 512 |
| 6 | kernel_size | 3 x 3 | 3 x 3 | 3 x 3 |
| 7 | pool_size | 3 x 3 | 3 x 3 | 3 x 3 |
| 8 | strides | 2 x 2 | 2 x 2 | 2 x 2 |
| 9 | padding | same | same | same |
| 10 | color_mode | grayscale | grayscale | grayscale |



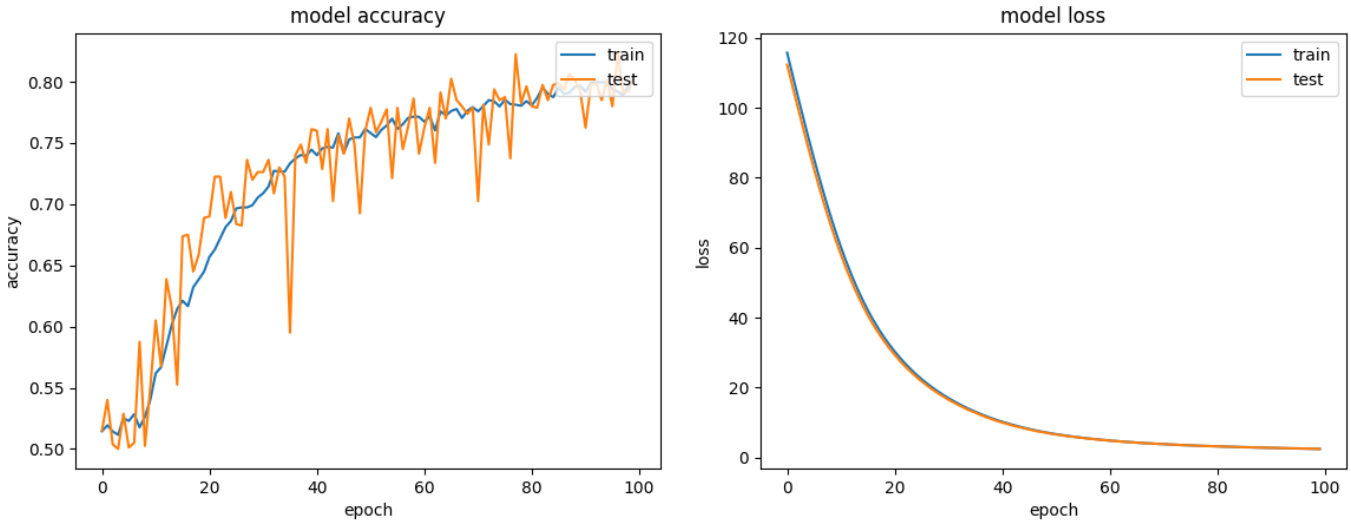

(a) Accuracy curve (b) Loss curve

Fig. 4: Graph showing training and testing curves (Experiment 1).




(a) Accuracy curve (b) Loss curve

Fig. 5: Graph showing training and testing curves (Experiment 2).




(a) Accuracy curve (b) Loss curve

Fig. 6: Graph showing training and testing curves (Experiment 3).

## IV. RESULTS

For the prediction, a total of 800 testing images, 400 belonging to each class were used. As the classification is based on binary being abnormal and normal. Hence, sensitivity, specificity, accuracy, precision, f-score, and support need to be defined in terms of confusion matrix (Fig. 5) and for the classification report (Table IX) too.

*i. Sensitivity:* Actual positive classes that are correctly predicted is sensitivity. Sensitivity is also termed as recall or true positive rate. Sensitivity must be as high as possible.

$$sensitivity = \frac{TruePositive}{TruePositive + FalseNegative}$$

*ii. Specificity:* Specificity or true negative rate can be defined as actual negative classes that are correctly predicted.

$$specificity = \frac{TrueNegative}{TrueNegative + FalsePositive}$$

*iii. Accuracy:* The measure of correctness is defined as accuracy.

$$accuracy = \frac{TruePositive + TrueNegative}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

*iv. Precision:* Precision is the number of correct prediction wrt all the classes. Precision must be high.

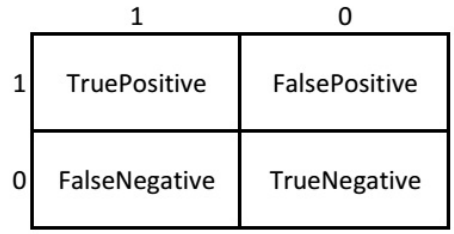
$$precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

*v. F-score:* F-score (F1-score) is the calculation of harmonic mean of sensitivity (recall) and precision. When any model having high recall and low precision (and vice-versa), the model isn't comparable. Therefore, f-score is used to make the model comparable. The best score is identified as 1, while the worst is 0.

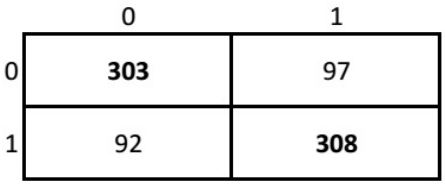
$$precision = \frac{2 * recall * precision}{recall + precision}$$

*vi. Support:* Support is the count of actual occurrences of the data present in a particular class. This indicates whether the count of data in classes are unbalanced or balanced.

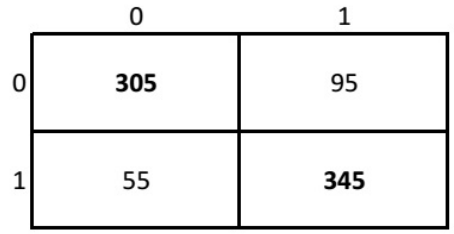
$$precision = \frac{TruePositive}{TruePositive + FalsePositive}$$



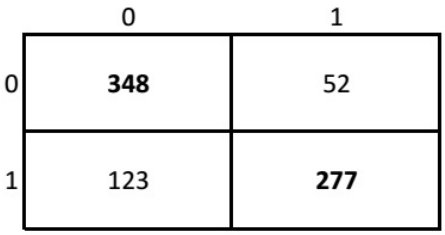

(a) CM Layout




(b) CM (Experiment 1)




(c) CM (Experiment 2)




(d) CM (Experiment 3)

Fig. 7: Confusion Matrix for binary classification for different experiments (0: abnormal 1: normal).

TABLE IX: Classification report

| Experiment no. | | Precision | Recall | F1-score | Support |
|---|---|---|---|---|---|
| 1 | 0(abnormal) | 0.74 | 0.87 | 0.80 | 400 |
| | 1(normal) | 0.84 | 0.69 | 0.76 | 400 |
| | Total/Avg | 0.79 | 0.78 | 0.78 | 800 |
| 2 | 0(abnormal) | 0.77 | 0.76 | 0.76 | 400 |
| | 1(normal) | 0.76 | 0.77 | 0.77 | 400 |
| | Total/Avg | 0.76 | 0.76 | 0.76 | 800 |
| 3 | 0(abnormal) | 0.85 | 0.76 | 0.80 | 400 |
| | 1(normal) | 0.78 | 0.86 | 0.82 | 400 |
| | Total/Avg | 0.82 | 0.81 | 0.81 | 800 |

After experimenting for fewer number of times, the result obtained is shown in Table IX. Confusion matrix (Fig. 7) shows that the CNN is able to predict most of the images as abnormal and normal. The accuracy graph for the experiment no. 1 and 3 is around 80.00%, while the experiment no. 2 is around 74.00%. From the graph (Fig. 4a, 5a, 6a), it can be found that the accuracy curve is not much smooth and more tuning of hyper-parameters needed to be done, which is an error and trail approximation. The loss curve seems to be good enough as the cost entropy loss is around 0.5 for the experiment no. 1 and 3. The testing run time on the convolutional neural network took 14.8 seconds approximately.

## V. Future Work

The report shows that the diabetic retinopathy binary classification screening can be performed with much higher accuracy than the result found. For the real-time simulation, much more training images can be added; as for

800 images, it took few seconds to classify. Sometimes it might also happen that the abnormal images might be classified as normal. This adversarial effect can be solved by implementing a process called adversarialization [9]. The convolutional neural network can be of more complex architecture such as VGGNet, ResNet, Inception, Xception and other architectures. Similarly, five-class classification can be performed with more less noisy dataset.

## ACKNOWLEDGMENT

I would like to vote my special thanks to my project guide, Prof. Balaraman Ravindran (Dept. of Computer Science & Engineering) and also to Karthik Thiagarajan (MS Scholar, Dept. of Computer Science & Engineering) and Saurabh Desai (Project Associate, RBC-DSAI laboratory) for guiding me all through my hard times. The project would not have been possible without their earnest and sincere guidance. I learned lot many things about neural networks and deep learning through this topic entitled, ”Diabetic Retinopathy Classification using Convolutional Neural Network”. Last but not least, special thanks to my guide for giving me an opportunity to work under him.

## REFERENCES

[1] G. G. Gardner, D. Keating, T. H. Williamson, A. T. Elliott.. Automatic detection of diabetic retinopathy using an artificial neural network: a screening tool. *British Journel Ophthalmology* 1996;**80**(11):940-944.
[2] Markku Kuivalainen. Retinal Image Analysis Using Machine Vision. *Thesis report Lappeenranta Univ of Tech, Dept of IT* 2005.
[3] Nayak J, Bhat PS, Acharya R, Lim CM, Kagathi M.. Automated identification of diabetic retinopathy stages using digital fundus images. *J Med Sys* 2008;**32**(2):107-115.
[4] Acharya UR, Chua CK, Ng EY, Yu W, Chee C..Application of higher order spectra for the identification of diabetes retinopathy stages. *J Med Sys* 2008;**32**(6):481-488.
[5] Acharya UR, Lim CM, Ng EY, Chee C, Tamura T.. Computer-based detection of diabetes retinopathy stages using digital fundus images. *P I Mech Eng H* 2009;**223**(5):545-553.
[6] P.Adarsh, D.Jeyakumari.. Multiclass SVM-based automated diagnosis of diabetic retinopathy. In: *Communications and Signal Processing (ICCSP),*2013 *International Conference on.* IEEE; 2013,p. 206-210.
[7] Ben Graham. Kaggle Diabetic Retinopathy Detection competition report. 2015.
[8] Harry Pratt, Frans Coenen, Deborah M Broadbent, Simon P Harding, Yalin Zheng.. Convolutional Neural Network for Diabetic Retinopathy. *Procedia Computer Science* 90 2016:200-205.
[9] Linda Roach. Artificial Intelligence, The Next Step in Diagnostics. In: *EyeNet* November 2017.
[10] Pascanu, Razvan et al. Understanding the exploding gradient problem. In: *CoRR* 2012.
[11] Wikipedia. https://en.wikipedia.org/wiki/Diabetic_retinopathy