2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)

A Deep Learning Method for the detection of Diabetic Retinopathy

Navoneel Chakrabarty
Computer Science and Engineering
Jalpaiguri Government Engineering College
Jalpaiguri, India
nc2012@cse.jgec.ac.in

***Abstract*— Many Diabetic patients suffer from a medical condition in the retina of the eye known as Diabetic Retinopathy. The main cause of Diabetic Retinopathy is high blood sugar levels over a long period of time in the retina known as Diabetes Mellitus. The primary goal is to automatically classify patients having diabetic retinopathy and not having the same, given any High-Resolution Fundus Image of the Retina. For that an initial image processing has been done on the images which includes mainly, conversion of coloured (RGB) images into perfect greyscale and resizing it. Then, a Deep Learning Approach is applied in which the processed image is fed into a Convolutional Neural Network to predict whether the patient is diabetic or not. This methodology is applied on a dataset of 30 High Resolution Fundus Images of the retina. The results, so obtained are a 100 % predictive accuracy and a Sensitivity of 100 % also. Such an Automated System can easily classify images of the retina among Diabetic and Healthy patients, reducing the number of reviews of doctors.**

Keywords*— *Diabetic Retinopathy, Diabetes Mellitus, High Resolution Fundus, Deep Learning, Convolutional Neural Network

I. INTRODUCTION

In simple words, Diabetic Retinopathy is an eye disease related to Diabetes. It is a direct consequence of damage of small blood vessels and neurons of the retina. It can lead to swelling and leakage of blood vessels, preventing blood from passing through and also sometimes growth of abnormal new blood vessels in the Retina. Spots or dark strings in vision, blurred vision, fluctuating vision, impaired colour vision, dark or empty areas in vision and vision loss are absolute symptoms of Diabetic Retinopathy [1]. The various signs and markers of diabetic retinopathy include micro-aneurysms, leaking blood vessels, retinal swellings, the growth of abnormal new blood vessels and damaged nerve tissues. Diabetic Retinopathy can be treated with methods like Focal laser treatment, Scatter laser treatment and Vitrectomy. Surgery often degrades or prohibits the development of diabetic retinopathy, but it is not a complete cure. As it is a lifelong condition, future retinal damage and vision loss is also possible [2]. So, a proper diagnosis of the disease is a necessity. Diagnosis methods like Fluorescein angiography and Optical coherence tomography which involves external fluid or dies to be applied on to the patients' eye after the Retinal Image is taken. But an Automated System which can immediately predict Diabetic Retinopathy without any external agent, is a more comfortable and convenient method both for doctors and patients.

This paper has been structured as an introduction, literature review, proposed methodology, training the model, details of the learning process, implementation details, results and conclusion

II. LITERATURE REVIEW

Many conventional methods, Machine Learning techniques and few Deep Learning approaches have been attempted for Diabetic Retinopathy detection.

- Review on Conventional Methods:
  - – Argade et al. proposed Image Processing and Data Mining Techniques for automatic detection of Diabetic Retinopathy [3].
  - – Mukherjee et al. proposed another conventional technique. The methodology followed by them included Image Processing which involves background normalization and contrast enhancement using histogram equalization. It is followed by Optical Disk Detection, Blood Vessel Extraction and Exudate Detection [4].
- Review on Machine Learning Techniques:
  - – Bhatia et al. proposed a Machine Learning Model for diagnosis of Diabetic Retinopathy using ensemble of classification algorithms, alternating decision tree, AdaBoost, Naive Bayes, Random Forest and SVM and achieved a maximum accuracy of 90 %, sensitivity of 94 % and F1-score of 90 % [5].
  - – Labhade et al. applied soft computing techniques for Diabetic Retinopathy Detection in which they used different classifiers like SVM, Random Forests, Gradient boost, AdaBoost, Gaussian Naive Bayes [6].
  - – Mohammadian et al. proposed a comparative analysis of 9 common Machine Learning Classification Algorithms for Diabetic Retinopathy Detection [7].
- Review on Deep Learning Approaches:
  - – Doshi et al. proposed a Deep Learning Approach involving a Deep Convolutional Neural Network with a specific Network Architecture obtaining a Quadratic Kappa Score of 0.3996 [8].
  - – Xu et al. applied Deep Convolutional Neural Networks for early automated detection of Diabetic

Retinopathy and achieved a highest accuracy of 94.5% [9].

– Gargeya et al. proposed a Deep Learning Model for identification of Diabetic Retinopathy and achieved a Sensitivity of 0.93, Specificity of 0.87 and Area Under the Receiver Operating Characteristic Curve of 0.94 [10].

III. Proposed Methodology

A. The Dataset

The High-Resolution Fundus (HRF) Image Database (benchmark dataset) consists of 30 High Resolution Fundus Retinal Images out of which, 15 images are labelled as Healthy and 15 images are labelled as Diabetic [11]. Sample Images are shown in Fig 1.

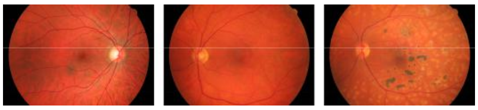

Fig. 1. Dataset Samples

B. Data Pre-Processing

This mainly consists of Image Processing Steps as here Data is referred as High-Resolution Fundus Retinal Images.

1) Conversion to Weighted greyscale: As all the images which were colour (RGB) initially, were converted to greyscale by taking a weighted average of the RGB pixels in which 0.299 of the Red (R) Component, 0.587 of the Green (G) Component and 0.114 of the Blue (B) Component are considered.

$$I = R * 0 \cdot 299 + G * 0 \cdot 587 + B * 0.114$$

where I is the Resultant Pixel

2) Resizing: All the converted greyscale images are resized to a fixed size of 1000 x 1000 pixels.

3) Pixel Rescaling: For every image, each and every pixel values are rescaled into a value between 0 and 1 by dividing by 255 for easy computation.

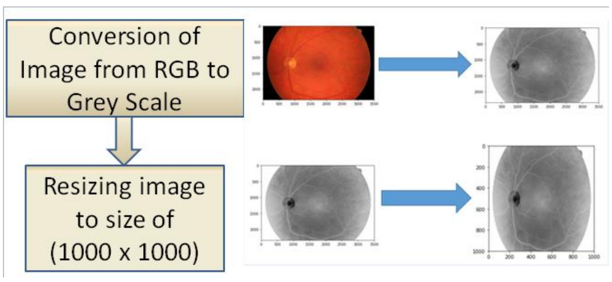

Fig 2. Image Transformations in Pre-Processing

C. The CNN Architecture

The deep-convolutional neural network architecture adopted is shown in Fig 10. The network contains an input layer which takes greyscale images of resolution 1000 x 1000 pixels as input. Then there comes 3-set-combination of Convolution. Each set consists of a Convolution Layer, a ReLU (Rectified Linear Unit) layer and a Max-Pooling Layer. The final set of feature maps (corresponding to a single image) obtained after the 3 sets are flattened or unrolled into a single feature vector in the Flattening Layer. The single feature vector is then fed into an Artificial Neural Network which forms the Dense Layer of the Convolutional Neural Network.

1) Convolutional Layer: Convolution is a combined integration of two functions and it shows how one function modifies the shape of the other. (Fig 3) Each Convolutional Layer in all the 3 sets have 32 features detectors of dimensions, 3 x 3. Each feature detector is convoluted with the input image to generate convolved feature maps corresponding to every feature detector. A small example of a Convolutional Layer is described in Fig 4.

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$

Fig.3. Convolution Operation

$$\begin{array}{|c|c|c|c|c|c|c|}\hline 0&0&0&0&0&0&0\\ \hline 0&1&0&0&0&1&0\\ \hline 0&0&0&0&0&0&0\\ \hline 0&0&0&1&0&0&0\\ \hline 0&1&0&0&0&1&0\\ \hline 0&0&1&1&1&0&0\\ \hline 0&0&0&0&0&0&0\\ \hline \end{array} \otimes \begin{array}{|c|c|c|}\hline 0&0&1\\ \hline 1&0&0\\ \hline 0&1&1\\ \hline \end{array} = \begin{array}{|c|c|c|c|c|}\hline 0&1&0&0&0\\ \hline 0&1&1&1&0\\ \hline 1&0&1&2&1\\ \hline 1&4&2&1&0\\ \hline 0&0&1&2&1\\ \hline \end{array}$$

Fig.4. Example of a Convolutional Layer

2) ReLU Layer or Rectification Layer: ReLU is Rectified Linear Unit defined as shown in Fig 5.

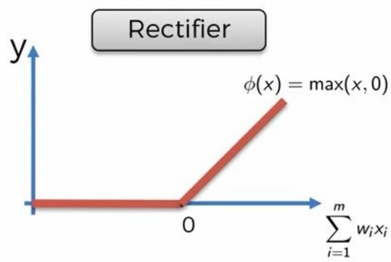

Fig.5. Definition of ReLU

Each and every pixel of the feature maps obtained in the Convolutional Layer is ReLU activated to introduce non-linearity in the feature maps.

3) Max-Pooling Layer: Max-Pooling operation is described in Fig 6.

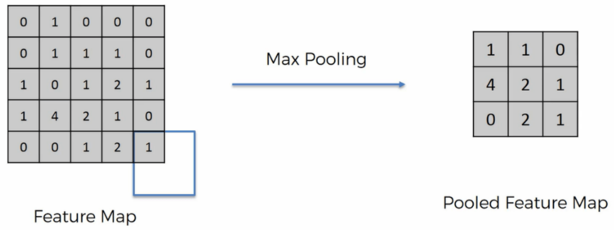

Fig.6. Example of a Max-Pooling Layer taking a pooling stride of 2x2 dimension.

In each and every feature maps obtained in the ReLU Layer, Max Pooling is done where the pooling stride is of dimensions (2 x 2), to preserve the features and for making the Convolutional Neural Network, spatial independent.

4) Flattening Layer: The final feature map, so obtained after the 3 sets, is flattened or unrolled into a single feature vector in this layer as shown by taking an example in Fig 7.

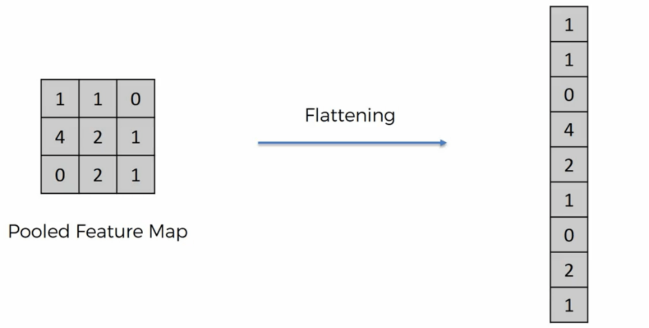

Fig.7. Example of a Flattening Layer

5) Dense Layer or Artificial Neural Network Architecture: The Artificial Neural Network or ANN consists of an input layer which takes a single feature vector as input. Then, it consists of a hidden layer containing 128 nodes to which the feature vector is forward propagated from the input layer by ReLU Activation. The following layer is the output layer which has a single node/unit that can assume values greater than or equal to zero but less than or equal to 1 as feature vector is forward propagated from the hidden layer to the output layer by Sigmoid Activation (shown in Fig 8) since it is a Binary Classification. If the output layer produces a value greater than 0.5, it is treated as Healthy but Diabetic otherwise. After the forward propagation steps, the whole Artificial Neural Network is back-propagated by taking Binary Cross Entropy Function as the loss function (shown in Fig 9). For reducing the loss i.e., the value assumed by the loss function, Adam Optimizer is used with learning rate 0.00005.

$$S(t) = \frac{1}{1+e^{-t}}.$$

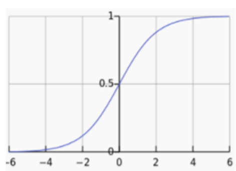

Fig. 8. Sigmoid/Logistic Function

$$L(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} H(p_n, q_n) = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right]$$

where N is the batch size

Fig. 9. Binary Cross Entropy Function

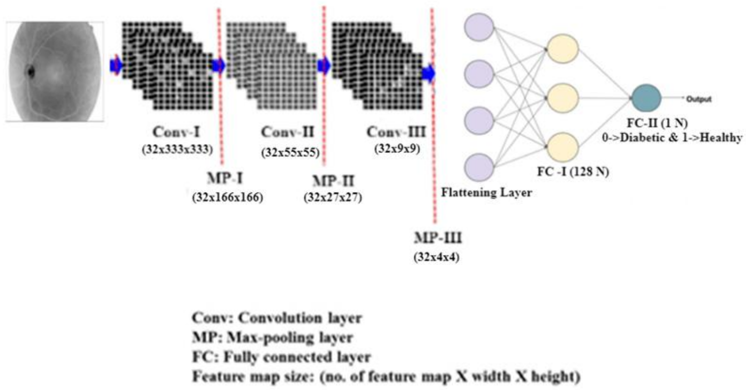

Fig. 10. The Schematic Diagram of the CNN

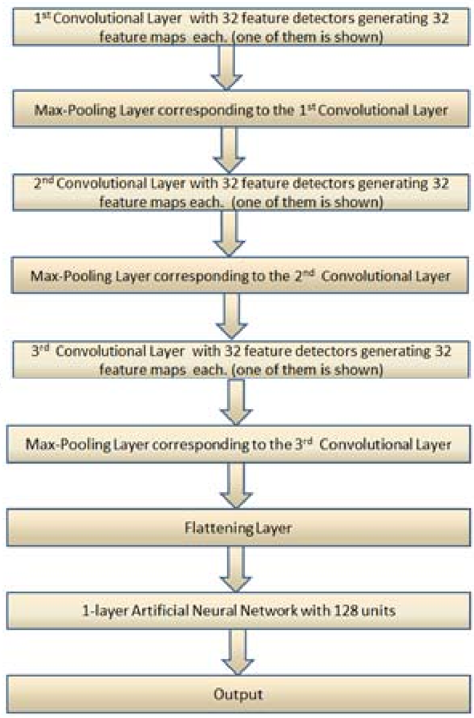

Fig. 11. Summary of CNN layers

D. Image Visualization of the CNN Layers.

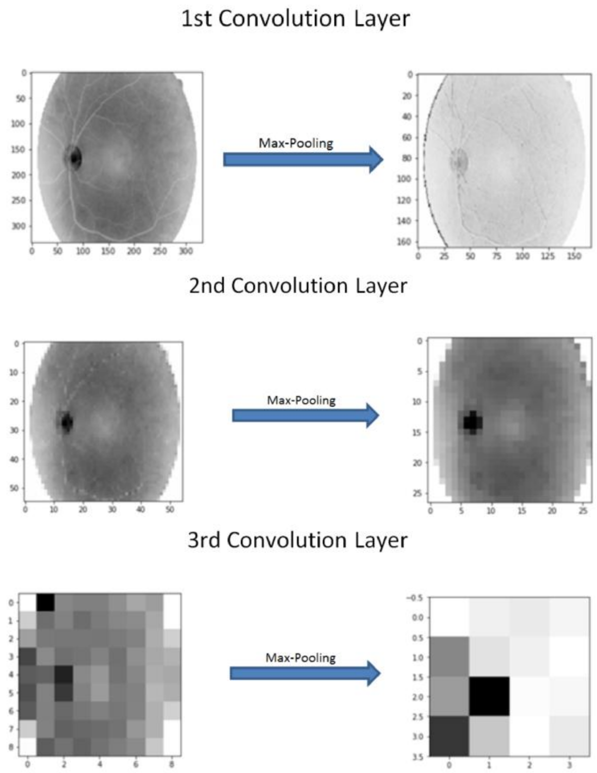

Fig. 12. CNN Layer Visualization

IV. TRAINING THE MODEL

Out of the total 30 images, 80 % of the total images, i.e. 24 images are chosen for training the model out of which 12 are labelled as Healthy and 12 are labelled as Diabetic. The remaining 20 % images, i.e. 6 images are chosen for validating the model, out of which 3 are labelled as Healthy and 3 are labelled as Diabetic. Then, all the 24 images in the Training Set are augmented by creating different samples of the training images by zooming (zooming range=0.2). This is done as the dataset is quite small and Image Augmentation increases the number of training examples from existing samples for better performance of the CNN.

V. DETAILS OF THE LEARNING PROCESS

In the learning process of the Deep Convolutional Neural Network, the Optimizer plays a pivotal role. The Optimizer used here is known as Adam Optimizer. It undergoes a mini-batch gradient descent where the batch size here, is set to 3 images per epoch or iteration. The number of epochs is set to 28 and the Neural Network is made to learn.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 333, 333, 32)	320
max_pooling2d_1 (MaxPooling2	(None, 166, 166, 32)	0
conv2d_2 (Conv2D)	(None, 55, 55, 32)	9248
max_pooling2d_2 (MaxPooling2	(None, 27, 27, 32)	0
conv2d_3 (Conv2D)	(None, 9, 9, 32)	9248
max_pooling2d_3 (MaxPooling2	(None, 4, 4, 32)	0
flatten_1 (Flatten)	(None, 512)	0
dense_1 (Dense)	(None, 128)	65664
dense_2 (Dense)	(None, 1)	129

Total params: 84,609
Trainable params: 84,609
Non-trainable params: 0

Fig. 13. Proposed Deep Learning Approach

VI. IMPLEMENTATION DETAILS

The Data Pre-Processing (Image Processing) is performed using Python's Scipy library's Miscellaneous Routines. The Convolutional Neural Network is trained on a machine with Intel(R) Core(TM) i5-4210U processor, CPU @ 1.70 GHz 2.40 GHz and 4 GB RAM. CPU is used as the interface with Python's Keras Library (TensorFlow in the backend).

VII. RESULTS

- The Training Accuracy describes the accuracy achieved on the training set.
  From this model, a Training Accuracy of 91.67 % is achieved, which implies that 22 out of 24 images were classified correctly whereas 2 images were misclassified.
- The Validation Accuracy describes the accuracy achieved on the test set.
  From this model, a Validation Accuracy of 100 % is achieved, which implies that 6 out of 6 images were classified correctly.
- The Sensitivity or Recall is defined as the proportion of correctly identified positives.

$$Recall = TP/(TP + FN)$$

  As a result, a Sensitivity of 1.0 or 100 % has been achieved from this model.
- Precision is defined as the ratio of correctly predicted positive observations to the total predicted positive observations.

$$Precision = TP/(TP + FP)$$

  As a result, a Precision of 1.0 or 100 % has been achieved
- F1-Score is defined as the Harmonic Mean of Precision and Recall.

$$F1 = 2 * (\frac{Precision * Recall}{Precision + Recall})$$

From this model, F1-Score of 1.0 or 100 % is achieved.

- Cross-Entropy Loss is the value returned by the Binary Cross-Entropy Loss Function.
  Our model resulted in a cross-entropy loss of 0.0339.

All the results are tabulated in Table I.

TABLE I.

Performance Metric	Result
Training Accuracy	91.67 %
Validation Accuracy	100 %
Sensitivity/Recall	100 %
Precision	100 %
F1-Score	100 %
Cross-Entropy	0.0339

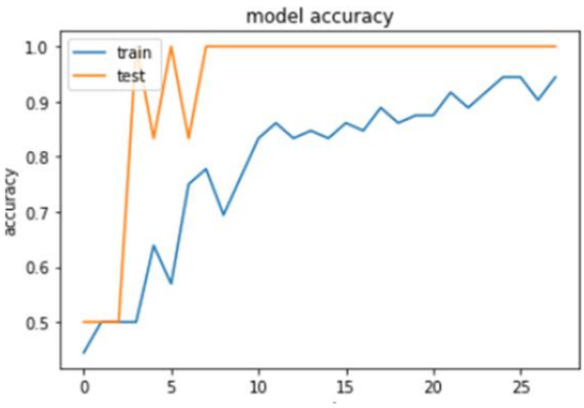

Fig. 14. Training and Validation Accuracy History (Accuracy vs Epoch)

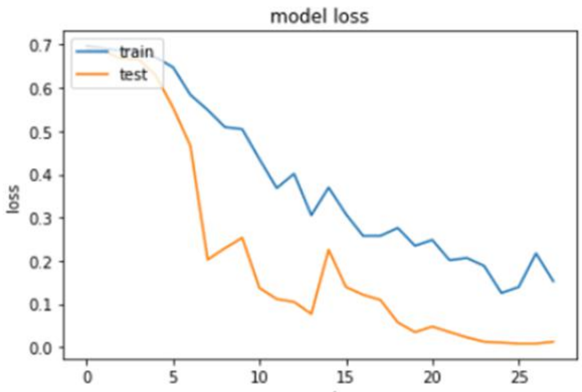

Fig. 15. Training and Validation Loss History (Loss vs Epoch)

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord “Format” pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.

TABLE II.

Comparison Parameters	Xu *et al.* [9]	Gargeya *et al.* [10]	Model
Sample Size	1000 images	75137 images	30 images
Accuracy (%)	94.5	-	100
Sensitivity	-	0.93	1.0

VIII. Conclusion

This paper proposed a Deep Learning approach to Diabetic Retinopathy via Convolutional Neural Network and designing a typical CNN Architecture. Finally, a full 100 % Validation Accuracy is obtained which is, by the best of our knowledge has been the highest ever numeric accuracy reached by any Automated Diabetic Retinopathy Detection Model. The research done in this paper is intended to help diabetic patients to remain cautious about their medical condition.

References

[1] https://www.mayoclinic.org/diseases-conditions/diabetic-retinopathy/symptoms-causes/syc-20371611

[2] http://www.advancedeyecareny.com/retinopathy/

[3] Ketki S. Argade, Kshitija A. Deshmukh, Madhura M. Narkhede, Nayan N. Sonawane and Sandeep Jore: ”Automatic Detection of Diabetic Retinopa-thy using Image Processing and Data Mining Techniques”, International Conference on Green Computing and Internet of Things (ICGCIoT), 2015.

[4] Anupriyaa Mukherjee, Diksha Rathore, Supriya Shree and Asst Prof. Shaik Jameel:”Diagnosis of Diabetic Retinopathy”, Int. Journal of Engineering Research and Applications, ISSN : 2248-9622,Vol. 5, Issue 2, (Part -4) February 2015.

[5] Karan Bhatia, Shikhar Arora and Ravi Tomar:”Diagnosis of Diabetic Retinopathy Using Machine Learning Classification Algorithm”, 2nd International Conference on Next Generation Computing Technologies (NGCT), 2016.

[6] Jyoti Dnyaneshwar Labhade, L. K. Chouthmol and Suraj Desh-mukh:”Diabetic retinopathy detection using soft computing techniques”, Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), 2016.

[7] Saboora Mohammadian, Ali Karsaz and Yaser M. Roshan:”A Com-parative Analysis of Classification Algorithms in Diabetic Retinopathy Screening”, 29th International Conference on Software Engineering and Knowledge Engineering (SEKE), 2017.

[8] Darshit Doshi, Aniket Shenoy, Deep Sidhpura and Prachi Gharpure:”Diabetic retinopathy detection using deep convolutional neural networks”, International Conference on Computing, Analytics and Security Trends (CAST), 2016.

[9] Kele Xu, Dawei Feng and Haibo Mi:”Deep Convolutional Neural Network-Based Early Automated Detection of Diabetic Retinopathy Using Fundus Image”, Molecules, 2017 (Open Access Journal), 22,2054; doi:10.3390/molecules22122054.

[10] Rishab Gargeya and Theodore Leng:”Automated Identification of Diabetic Retinopathy Using Deep Learning”, American Academy of Ophthalmology, 2017.

[11] High Resolution Fundus Retinal Image Database https://www5.cs.fau.de/research/data/fundus-images/

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.