IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A DEEP LEARNING APPROACH FOR DIABETIC RETINOPATHY DETECTION

[1]Supriya.A, [2]Dr. Rajeshwari. J

Department of ISE, Dayananda Sagar College of Engineering, VTU, Bengaluru, Karnataka, India

Abstract: Diabetic retinopathy is an eye ailment caused due to prolonged diabetes. If unacknowledged or ineffective treatment can lead to irreversible vision loss. This ailment can be identified by scrutinizing colour fundus photographs of the human retina. Our work assesses in deep learning field which is used for the detection and class segregation of diabetic retinopathy into binary or multiclass and collate them based on different metric measures like precision, recall, accuracy. The classification of diabetic retinopathy involves multiple steps like pre- processing of image, feature extraction, dimensionality reduction and classification of exudates, woolen spot, hemorrhages & microaneurysm. In this paper, Deep neural network approach is implemented by making use of online available Kaggle dataset, various metrics are calculated and applied pre-processing techniques.

Keywords: Convolution Neural Network, Deep learning, diabetic retinopathy, Kaggle dataset, Dropout.

1. INTRODUCTION

Diabetic retinopathy DR for short is an eye ailment that vandalizes the blood veins in tissues of the retina and is caused due to long-standing diabetes or uncontrolled sugar, which is currently the paramount reason for blindness in the world [1]. Diabetes can be classified into two categories. Diabetic retinopathy is very customary in people with type-II diabetes (40% of the total diabetic population). The main components in the fundus image that are suitable in detecting DR are optic nerve head, retinal vessels, macula, fovea, hemorrhages, microvascular lesions, and soft and hard exudates. Based on the various studies of the retinal fundus, DR can be classified into broad two categories – Non-Proliferative DR (NPDR) and Proliferative DR (PDR) [2]. NPDR is the early stage of DR in which there are hardly any signs of DR presence, it's an asymptomatic stage. NPDR can be additionally be classified into mild, moderately severe, and very severe. In the above-mentioned stages, additional pathological structural signs are visible in the eye fundus. Majority of people affected by diabetic retinopathy approach an ophthalmologist only after they start experiencing vision distortion and blurry vision thus reaching the last or second last stage of DR as no signs of DR presence can be found in the NPDR stages. Various grades of diabetic retinopathy are presented in Fig.1 One of the major advantages of using a Deep CNN [3] model is that it can learn very complex functions and patterns in the images, buy extracting features from the data while the model gets trained and stores this info into weight and biases matrices. Deep Convolutional networks requires the user to feed into raw images and with a very little pre-processing stage like noise removal, resizing/cropping the model will be able to classify the DR into 5 categories. This paper is assembled as follows. Section 2 reveal related work on classification of diabetic retinopathy using various machine learning approaches. Then Section 3 discuss various advance deep learning techniques for binary classification in details. While section 4 contains experimental results and section 5 contains conclusion.

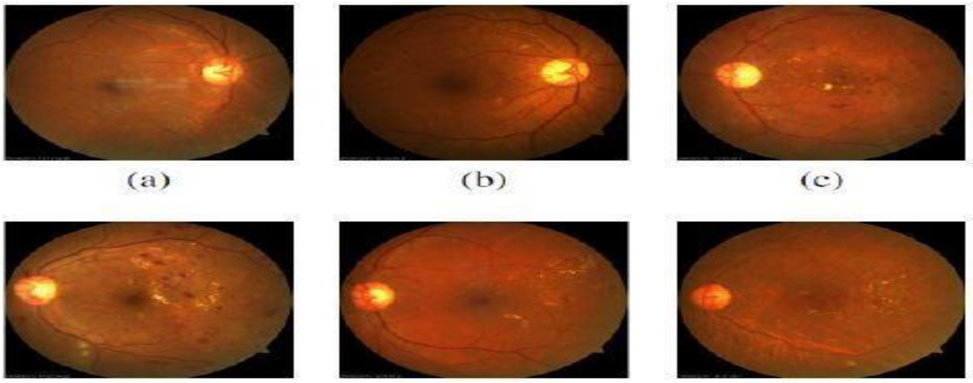

Fig 1: Classification of DR (Left to Right) (a) Mild Non-Proliferative DR (b) Moderate Non-Proliferative DR (c) Proliferative DR (d)Severe NPDR (e) Stage-V NPDR

2. RELATED WORK

Earlier research works done on detecting diabetic retinopathy were based on using human-designed features to detect the presence of anomalies such as hard exudates, soft exudates, hemorrhages, and red lesions, and many more. Classification and detection were performed using these features as input to various non-deep learning models like SVM and KNN [4] [5]. Deep learning models have proven their potent in image classification tasks in the last few years. For example, Neural architectures trained on the ImageNet dataset were used in [6]. In [7] Pratt et al. developed a deep learning ensemble technique on the Kaggle Diabetic retinopathy dataset, which uses both random up sampling and down sampling technique to mitigate the heavy class imbalance problem. It also employs various augmentation and patches for 512X512 keeping the aspect ratio unchanged. It uses pertained models like ResNet, VGG-19, InceptionV3, and InceptionV5 model in parallel and does a majority vote for classification and comes up with the accuracy of 80.8%, precession of 63.8% and F1-score of 53.7.%. M. Jahiruzzaman in [8] tries to achieve the classification task using an unsupervised learning technique called K-means clustering for lesion segmentation followed by a multilayer perceptron (aka simple feed-forward neural network). The dataset used is Messidor which is a binary classification problem of No DR and DR. The results obtained an accuracy of 0.90. But the major disadvantage is the simplicity of the Messidor dataset as it is very small and binary. Makes comprehensive comparability between Deep neural networks & classical machine learning techniques and inter compares deep learning models in the classification of DR based on their AUC score. The author in [9] developed a binary-stage classification model, in which the type II errors are dismissed in the 1st stage itself. In the 2nd stage for lesion classification, they have used classifiers like Gaussian Mixture Model (GMM), k-NN, and SVM (Support Vector Machine). They attain a recall value of 1, but the precision of 0.531, and AUC 0. 904. However, the features used in this approach was very limited in nature. In [10] Y. Sun tries to achieve the result using the 1D CNN model along with Batch normalization on the patient medical reports like blood pressure, glycogen index, and other vital signs rather than using the eye fundus imaging achieving test accuracy up to 93.02% using Logistic regression, 0.975 accuracies using CNN architecture. But the caveat is instead of using the eye fundus images this paper investigates the reports of the patient, which is more time consuming as first all the vital information needs to be gathered which involves sophisticated machinery and manpower, thus increasing the time and cost. In recent times authors like in [11] have used mean squared error (MSE) as cost function (aka loss function) thus transforming the classification problem into a regression problem. To achieve better score people have also tried ensemble, by an additive combination of various models like NB Classifier, SVM, Logistic regression, as well as modern deep neural networks, with mean squared error as cost function used to detect/classify DR problem with the correct classification rate of 0.736, Kappa score of 0.676, and F1-score of 0.417.

3. PROPOSED METHOD

3.1 Dataset Load

In this project we are using Kaggle image fundus. The fundus image contains mainly macula, optic disc, blood vessels, and some other structures. There are several publicly available datasets with fundus images used for training machine and deep learning models. For training the model, we split the dataset into training and validation set, ideally, we should have split the dataset into training, validation and test dataset but due to limited size of the dataset we split it into two. This split is done, and the training dataset is used to train the model and validation set is used for testing the test accuracy, as the model performance is more reliable on a completely unseen dataset. Test accuracy of a model largely depends on the size (No. Of images used for training) and quality of images. Therefore, all noisy and corrupted data points must be removed from the dataset and one very popular dataset is Kaggle dataset.

3.2 Image Pre-Processing Techniques

The model expects a fixed input size, so we have resized all the images to 224x224 dimensions, using the OpenCV2 library. The 224 dimensions were chosen keeping in mind that most of the visible patterns must not be lost due to cropping and as per recommendation given by Graham, B [12]. Since the images have been captured in real environment and not in restricted lab conditions under varied lighting conditions .The other image preprocessing techniques like augmentation is applied as discussed in 3.3 section. Fig 2. Illustrates the Pre-processed image.

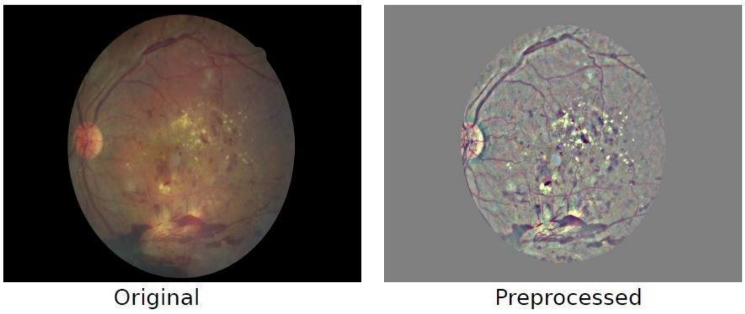

Fig 2: Left is the original image while right illustrates a pre-processed image

3.3 Data Augmentation

Neural networks are capable to learn complex functions very easily and therefore tend to memorize the training data (aka overfitting). Overfit is the situation where the model performs extremely great on already seen data but performs pathetic on unseen new data. Therefore, it has memorized the dataset rather than learning useful patterns within the dataset that can be used for classification in general. Therefore, to overcome the problem of overfitting we have done augmentation of images [13] taking the relative size of each class since it is very skewed as most of the images lie in class MILD and very few lie in class MODERATE. More than 50% of the images belong to class MILD Neural networks are very sensitive to this kind of imbalanced distributions.Fig.3 represents images of post augmentation. The various image augmentation techniques used in pre-processing as given below.

1. We have sheared the images with an angle between 30 and 120.
2. Horizontal and vertical flips were performed.
3. We have squeezed & stretched images randomly between (1/1.4, 1.5).
4. We also cropped the images from the edges up to 95% of their original size to get rid of useless information.
5. We randomly shifted the images between -25 and +25 pixels.
6. Image rotation was performed between 0 to 90 degrees.

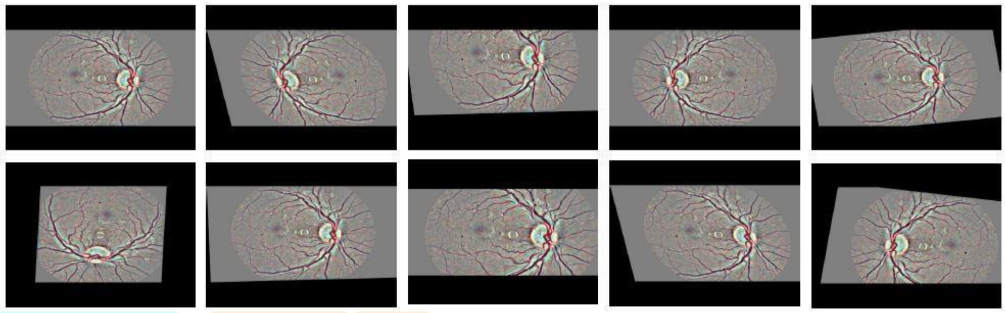

Fig 3: Illustration of post image augmentation.

3.4 Convolution Neural Network architecture

Our Model is implanted by VGGNET architecture as shown in Fig 4.In this paper Kaggle dataset is collected from online source. Then we are gathering into a directory. Once it's done with the Labels. Then move to the training section. In the training section VGG16 architecture model is trained with the available dataset, then model classifiers get created. Then this model will be ready for Predicting the output. The input layer has 224 x 224 neurons with ReLU as the activation function. After trying out with various kernel filters the best performing was 3x3 filter. Therefore, every convolution layer of our network has the same kernel filter size and stride of 1 except in the 1st and 3rd convolution layer with a stride of 2. ReLU activation function was used in all convolutional layers for non-linearity. For Max pooling, we have used a 3x3 filter throughout the network. The latent extracted features are passed into the CNN network with the last 2 layers as a fully connected layer, having 1024 neurons each and dropout of 0.5 to prevent from over-fitting.

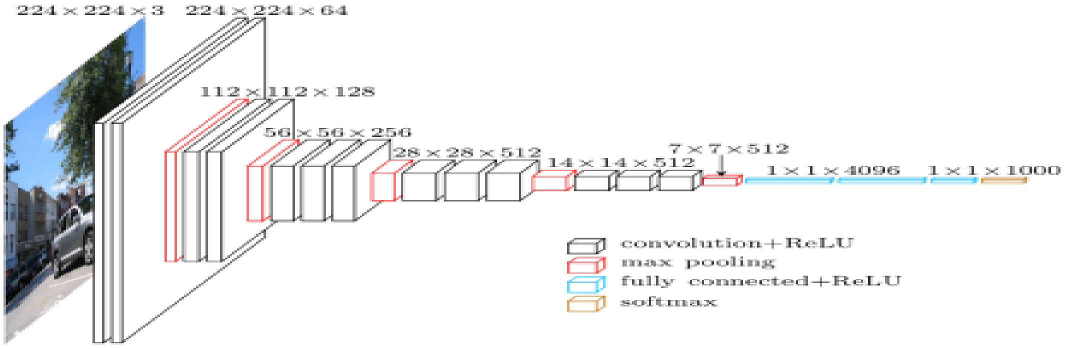

Fig 4: The proposed VGG architecture with a 16-layer depth while the kernel size of 3x3 and max-pooling of size 3x3 is used. After the convolution layers, the 2 FCC layers with 1024 neurons is used and then 1 FCC with 5 neurons and SoftMax is used for output.

3.5 Training

Our proposed neural network has approximately 1.4*10ˆ6 parameters which were initialized using He-et-al weight initialization technique. The network was trained with a Stochastic Gradient Descent optimization solver and augmentation is done as a pre-processing stage. L2 regularization of value 0.05 was applied to all weighted layers, to prevent the overfitting of the model. We also used dropout layers heavily in the last 3 fully connected layers to reduce overfitting. After trying various learning rates, we found 0.0001 to be the right starting point. To prevent overshooting from the global minima we have decreased the learning rate iteratively.

Table 1 depicts the values of the final hyperparameters used while training the CNN model. For training the model, we partition the dataset into training & test set, ideally, we must partition the dataset into training, validation, and test dataset but **due to the limited size of the dataset, we split it into two and the training dataset is used training the model and the test set is** used for testing the test accuracy, as the model performance is more reliable on a completely unseen dataset. To increase the robustness and reliability of our model we have done 5-fold cross-validation during model training. Test accuracy of a model largely depends on the size (No. Of images used for training) and the quality of images. Therefore, all noisy and corrupt data points must be removed from the dataset and one very popular dataset is the lite Kaggle dataset. The images in the lite Kaggle dataset are captured in a Realtime environment, and therefore many of the images are have varying contrast from low to high and noise present.

Table 1: Hyper-Parameter Tuning for VGG16 Model Architecture

HYPER-PARAMETERS	*VALUES*
OPTIMIZER	Adam Optimizer
LEARNING RATE	0.0001
BATCH SIZE	32
EPOCHS	800
ACTIVATION FUNCTION	ReLU and SoftMax
REGULARIZATION	Dropout and L2-regularization

4. Experimental results

4.1 Performance evaluation on Diabetic Retinopathy

The data was divided into 2 sets training set of 80% and validation set of 20%.To train the model,80% of training set is used and validation set is used for testing the test accuracy, as the model performance is more reliable on a completely unseen dataset as Shown in Fig.5, our model has been trained with 95% accuracy , the training loss with 0.1227 and Validation loss of 0.3824. By changing the epoch values, our model gives validation accuracy of 80% as shown in Fig.6. Many iterations of tests were done to come up with the best hyperparameters. VGG 16-layer CNN model performed best in terms of both time and accuracy.

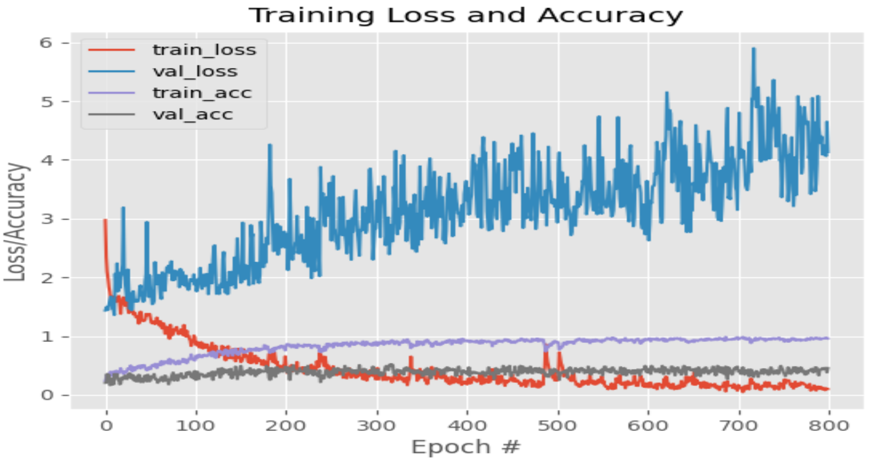

Figure 5 :Training loss and Accuracy

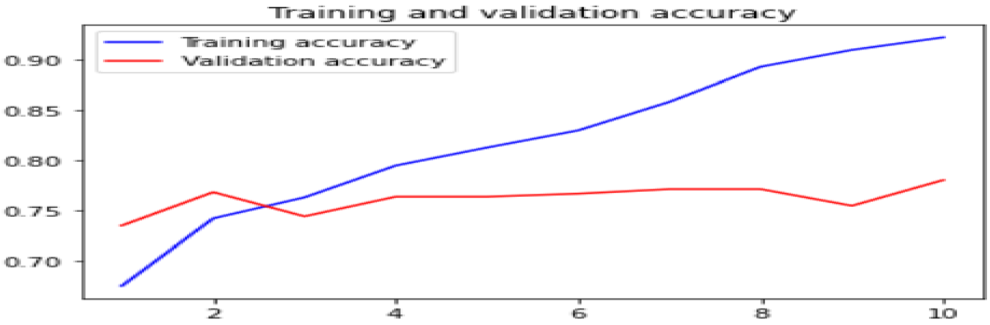

Figure 6 :Training Accuracy and Validation Accuracy

4.2 Various performance metrics are used for evaluating Deep learning models.

Accuracy can itself be false indicator in case of un-balanced dataset known as "accuracy paradox" and hence we need other metrics also to get better judgment of model.

$$\text{Accuracy}=\frac{\text{True positive + True Negative}}{\text{True positive +False positive +True Negative +False Negative}} \qquad (1)$$

Precision in simplest term is defined as ratio of relevant to total instance appeared in output and recall is ratio of total relevant instance to instance that appeared. Precision is more important among the two as we want less false positives or Type II errors. F1-score is the harmonic mean between the precision & recall, it's an indicator of the robustness of our model therefore, how well our model classifies the correct labelled data.

$$\text{F1-Score} = 2^* ((\textit{Precision*Recall}) / (\textit{Precision + Recall})) \qquad (2)$$

F1-Score of five DR stages is discussed in Table 2 and achieved accuracy score of 80% using our model on the lite version of Kaggle dataset which is remarkable for the model as it is trained for only 10 epochs on limited hardware, this could be further be increased by training the model on the entire Kaggle dataset and for epochs close to 800. The performance of our proposed model can be summarized in a table shown below in Table 2.

Table 2:Illustrates the performance of our proposed model in classifying of DR into 5 stages.

DR Grade	Grade Name	Precision	Recall	F1-score
0	Normal	1.00	0.75	0.86
1	Mild NPDR	0.47	1.00	0.64
2	Moderate NPDR	1.00	1.00	1.00
3	Severe NPDR	1.00	0.25	0.40
4	Proliferative DR	1.00	0.88	0.93

5. CONCLUSION

The efficient method of detecting the Diabetes Retinopathy at any stage by using simple proposed solution. Using deep learning CNN model along with AI used to develop automated DR detection at early stage. Neural network-based approaches as it tends to perform lot better than classical ML models. Kaggle DR dataset has become the de-facto dataset today. Eye status has been classified in to one of the among following classes NORMAL, MILD, MODERATE, SEVERE and PDR. These classes are processed by VGG16 model. Our project is aimed to make analysis of Diabetes retinopathy faster, cheaper and helps to detect Diabetes retinopathy at early stage. The experimental results have demonstrated the robustness and reliability of our proposed model to be good enough to be deployed in the real-time diagnosis of DR.

ACKNOWLEDGMENT

I would like to thank Dr. Rajeshwari. J, Associate Professor in Department of ISE, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India

REFERENCES

[1] N. G. Congdon, D. S. Friedman, and T. Lietman, "Important causes of visual impairment in the world today," Jama, vol. 290, no. 15, pp. 2057–2060, 2003.

[2] J. Nayak, P. S. Bhat, R. Acharya, C. M. Lim, and M. Kagathi, "Automated identification of diabetic retinopathy stages using digital fundus images," Journal of medical systems, vol. 32, no. 2, pp. 107–115, 2008.

[3] P. Y. Simard, D. Steinkraus, J. C. Platt, et al., "Best practices for convolutional neural networks applied to visual document analysis.," in Icdar, vol. 3, 2003.

[4] N. Silberman, K. Ahrlich, R. Fergus, and L. Subramanian, "Case for automated detection of diabetic retinopathy," in 2010 AAAI Spring Symposium Series, 2010.

[5] Sopharak, B. Uyyanonvara, and S. Barman, "Automatic exudate detection from non-dilated diabetic retinopathy retinal images using fuzzy c-means clustering," sensors, vol. 9, no. 3, pp. 2148–2161, 2009.

[6] S. Wang, Y. Yin, G. Cao, B. Wei, Y. Zheng, and G. Yang, "Hierarchical retinal blood vessel segmentation based on feature and ensemble learning," Neurocomputing, vol. 149, pp. 708–717, 2015.

[7] Pratt, F. Coenen, D. M. Broadbent, S. P. Harding, and Y. Zheng, "Convolutional neural networks for diabetic retinopathy," Procedia Computer Science, vol. 90, pp. 200–205, 2016.

[8] M. Jahiruzzaman and A. A. Hossain, "Detection and classification of diabetic retinopathy using k-means clustering and fuzzy logic," in 2015 18th

International Conference on Computer and Information Technology (ICCIT), pp. 534–538, IEEE, 2015.

[9] S. Roychowdhury, D. D. Koozekanani, and K. K. Parhi, “Dream: diabetic retinopathy analysis using machine learning,” IEEE journal of biomedical and health informatics, vol. 18, no. 5, pp. 1717–1728, 2013.

[10] Y. Sun, “The neural network of one-dimensional convolution-an example of the diagnosis of diabetic retinopathy,” IEEE Access, vol. 7, pp. 69657–69666, 2019.

[11] D. T. Butterworth, S. Mukherjee, and M. Sharma, “Ensemble learning for detection of diabetic retinopathy,” in 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 2016.

[12] B. Graham, “Kaggle diabetic retinopathy detection competition report,” University of Warwick, 2015.

[13] M. A. Tanner and W. H. Wong, “The calculation of posterior distributions by data augmentation,” Journal of the American statistical Association, vol. 82, no. 398, pp. 528–540, 1987.