Mild DR Detection from CLAHE Fundus Images using Experimental Minimal CNN Model with Batch Normalization

Ebin PM[1*], P Ranjana[2]

[1*]Computer Science and Engineering, Hindustan University,

Chennai, India, 603103, 0000-0001-8302-796X

[2]Computer Science and Engineering, Hindustan University,

Chennai, India, 603103, 0000-0003-4680-4998

Corresponding Author E-mail: pmebin74@gmail.com

Orcid ID: 0000-0001-8302-796X

Abstract

Vision impairment is one of the major problems affecting middle-aged individuals due to uncontrolled blood sugar levels, commonly known as Diabetic Retinopathy (DR). The small abnormalities in the retinal capillaries, called microaneurysms and intra retinal bleeding, are the initial symptoms of Diabetic Retinopathy. Clinical identification of Diabetic Retinopathy is a time-consuming and difficult process due to limitations in resources and experienced doctors. Early detection is crucial in avoiding the progression of Diabetic Retinopathy, highlighting the importance of an automated DR detection method to identify symptoms in its early stages. In this paper, researchers developed an Enhanced Minimal Convolutional Neural Network (EMCNN) model to classify Mild-DR and No-DR fundus images using a binary classification process. The fundus images were pre-processed using Contrast Limited Adaptive Histogram Equalization (CLAHE) method before passed through the network. EMCNN is an experimental model that enjoys a minimum number of layers and batch normalization to minimize the training effort. Finally, the EMCNN model is compared to existing models in terms of accuracy and efficiency

Keywords: CLAHE, Deep Learning, Diabetic Retinopathy, EMCNN

Introduction

There exist many reasons for visual impairment and one of the major reason is diabetes. The longer diabetes and poor control of blood sugar may develop the eye complication called Diabetic Retinopathy (DR). DR makes damage in the blood vessels of the retina and dark spots, blurred vision, fluctuations in the vision are the main symptoms and finally the loss of vision. DR affects both eyes and may lead to the permanent blindness. The treatments of the Diabetic Retinopathy can prevent the further vision loss only. Diabetic Retinopathy stages can be divided into two main categories: Non-Proliferative Diabetic Retinopathy (NPDR) and Proliferative Diabetic Retinopathy (PDR). The main symptoms at the Non-Proliferative stage are exudates and microaneurysms, while at the Proliferative stage; the blood vessel walls leak fluid and blood. Proliferative is the severe type and leads to grow new and abnormal blood vessels, leak protein and lipids in the vitreous. Physical tests are nowadays available like Visual Acuity Test (VAT), Pupil Dilation and Optical Coherence Tomography (OCT) are time consuming and affects patients. The interior surface of the eye can be examined by Ophthalmoscopy, which cover retina, optic disc, macula, fovea and blood vessels. The fundus images of the eye can give important information related to the Diabetic Retinopathy

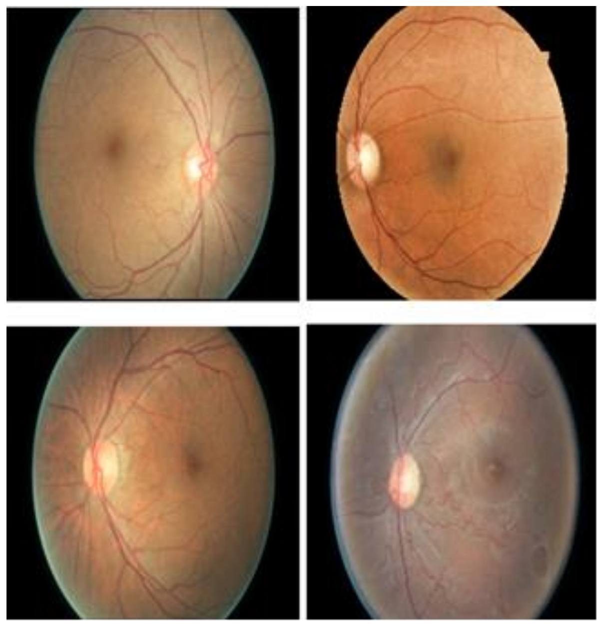

Fig.1. No-DR (Top), Mild-DR (Bottom)

Many approaches are raised by the big data and Artificial Intelligence (AI) to address the emergence and spread of DR. Artificial Intelligence was majorly used in detecting many health issues and plays an important role in tackling DR disease to an extent. The field of AI helps a lot to cluster the disease areas and to have a proper monitoring of the cases. It also helps to predict future outbreaks based on existing cases and to analyses the mortality risks. Disease management studies the trend of the disease by allocating resources, doing proper training, maintaining proper records and finally doing pattern recognition. The field of computer science can contribute efficient methods and models to detect Diabetic Retinopathy in its early stage through Deep Learning mechanism. Artificial Intelligence (AI) can helps medical diagnosis in a large extent and can give accurate prediction in an automated way. Deep Neural networks inspired from brain can learn the patterns from the images, which can be used for disease identification in medical field. When considering Deep Learning, there is a mechanism called Transfer Learning, which can provide mode accurate models in less epochs. The top layer of the pretrained model can be fine-tuned according to our need and the model can be developed. In two-class classification, the model can recognize the features like exudates and hemorrhages in the fundus images. The images can be downloaded from Kaggle, Messidor that are publically available.

This work involves the following processes.

1. We proposed an Experimental Minimal Convolution Neural Network (EMCNN) to detect the mild diabetic retinopathy symptoms and classify mild DR and No DR categories from the fundus images of the eye.
2. We utilized a mixture of Messidor and EyePACS (Kaggle) publically available dataset and applied

CLAHE algorithm before passing the dataset to the proposed EMCNN

4. To calculate the mean and variance of features with a batch size of 30, we offered a batch normalization (mini batch) approach.

3. The EMCNN is trained with CLAHE dataset and the results are compared in terms of accuracy, Precision, Recall and F1 score with other existing models. We also created a ROC curve to analyze the performance of the proposed classification model.

Literature Survery

Recently different research papers are dealing Diabetic Retinopathy detection using deep learning, transfer-learning methods, and produced some interesting results. Some authors are dealing with transfer learning methods and they all are used publically available datasets like kaggle and Messidor. Some of the recent works with fruitful results are given. Table 1 shows the summary of the existing works.

S. S. Karki et al [1] Diabetic Retinopathy severity experiment was conducted using an ensemble method with Efficient Nets B1 (256×256), B2 (224×224), B3 (256×256) and B5 (256×256). The model was combined with Different EfficientNet models and finally it achieved 92% quadratic kappa score. The datasets were used is EyePACS plus APTOS. In the preprocessing, radius reduction method was used to reduce the irregularities along the circular border. The model was trained on 20 epoch using Tesla K80 GPU. The EfficientNet B3 performed better on the test set as compared to others.

R. S. Rajkumar et al [2] prepared the model using the advantage of Transfer Learning with ResNet-50 to reduce the vanishing gradient problem. The model was used with 35,000 images resized to 512×512, and an image thresholding method was applied to separate the object from the foreground pixels and background pixels. The model was run on Tesla V100 GPU and an accuracy of 89.4%, specificity of 97% and sensitivity of 57% was achieved. The model was compared with other models, and it was found that the proposed model performed well. The researchers [3] are proposed a hardware friendly Binary Convolutional Neural Network (BCNN) model to reduce the memory consumption and speed up the execution. The convolution and dense layers are binarized. The binarized activations and biases reduces the number of parameters, which reduces the memory occupancy. The BCNN act as a regularizer to control overfitting without the usage of Dropout layers. The dataset were used is EYEPACS and the experiments were conducted in NVDIA GPU system. The model is compared with others and gain 37.50% of reduction in memory usage, 49.34% gain in run time. The researchers [4] proposed a method using the combination of VGG network and auto encoder, which prevents the overfitting while training the model. VGG network is used for transfer learning purpose. Auto encoder encode the input image and VGG nets inspire it. Data augmentation and batch normalization was applied in the pre-processing stage. ReLU activation function was used in all the layers of auto encoder and classifier. The training done in 100 epochs. The optimizer is used as Adam with learning rate 0.001 and parameters 0.9 and 0.999. The auto encoder is trained for the first 50 epochs and for the next 30 epochs; the auto encoder and classifier are jointly trained on the reconstruction loss and classification loss. Finally, in the last 20 epochs the encoder weights are freeze and train only the classifier weights on the classification loss. For the training, the environment used as GPU based virtual machine on Google Colab.

To classify the DR stages, the researchers [5] used a deep learning approach using more than 3700 images. The images were collected from AECS and classified into five categories: normal images, mild Nonproliferative, moderate Nonproliferative, severe Nonproliferative, and proliferative. The dataset is imbalanced so that 85% data is used for training and 15% is used for testing. The dataset were unstable because of various resolutions. OpenCV converted the images to Gray scale using weighted method. Laplacian method was used to drop the blurred images. The image is then resized to 256×256 and cropped it circularly o remove the background. Augmentation is applied to increase the diversity in data sets. The researchers used Dense Net as their CNN model in which each layer receives the collective knowledge from the previous layer. The model used Adam optimizer, Binary cross entropy as the loss function and Accuracy as the metric. The model obtained the training accuracy 96.98% and validation accuracy 96.69%. Early stopping method is provided to prevent overfitting which is happened after 11th epoch. The researchers also performed fivefold cross validation got accuracy 83.6% and kappa score of 88.4%. The training setup was in Linux environment with ASPEED Technology. The researchers [6] proposed a CNN ensemble based framework to classify and detect Diabetic Retinopathy stages using color fundus images. The ensemble method combines several machine learning models and the researchers combines Xception, Inceptionv3, Resnet50, Dense121 and Dense169. Kaggle dataset were used and some pre-processing techniques like resizing, up sampling flipping and down sampling. 64 percent data is used for training, 20% for testing and 16% for validation were used. The model was trained on NVIDIA Tesla k40 and accuracy 80.8% recall 51.5% specificity 86.72% precision 63.85% and F1 score 53.74% was achieved.

R. N. Lazuardi et al [7] used EfficientNet-B4 and EfficientNet-B5 to train and detect the Diabetic Retinopathy on Kaggle data sets. The images are pre-processed using Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm to improve the contrast in images and image central cropping. The images were standardized in to256×256 and 512×512 pixels. The 80% images were used for training and 20% images were used for testing purposes. For the initial training 256×256 pixels images were used and for progressive training 512×512 pixels images were used. In total, one model is trained for 100 epochs. Dataset is highly imbalanced so that the evaluation matric used in this research is kappa score. For EfficientNet-B4 the kappa score 0.7922, F1 score 0.8269 and Accuracy 83.87% obtained. For Efficient-B5 the kappa score 0.7931, F1 score 0.8265 and Accuracy 83.89% obtained.

The researchers [8] did DR classification with eleven pretrained deep learning models using OCT images, selected best one and optimized. The optimization process decreases the training time and maintain high classification accuracy. The database contains four CNV, DME, DRUSEN and NORMAL images. Pre-processing likes removing white pixels from the image, image segmentation into 3 levels, converting segmented image into color image and subtracted the green channel, image conversion into binary, removal of small objects and detection of ROI, extraction of ROI. During the training time, validation was performed on every five iteration to avoid over fitting problem. Out of eleven pre trained deep learning model, DenseNet-201 obtained 97%accuracy, 99% specificity and 97% precision. The experiment were conducted in MATLAB. The optimized DenseNet-201 is used as the feature extractor and trained an ANN to perform the diagnosis of Diabetic Retinopathy. The proposed model (DenseNet-201 + ANN) obtained 98% accuracy, 99% specificity and 98% precision.

Chetoui M et al [9] used EfficientNET-B7 CNN model and fine-tuned to detect Diabetic Retinopathy. The two publically available datasets EyePACS and APTOS2019 were used. The researchers added Global Average Pooling (GAP) to improve the accuracy and reduce overfitting. After that a dense layer of 1024 and Dropout of 25% was added. Softmax gave the probability prediction score later. In the pre-processing stage, images are resized in to 224×224 pixels, the experiment were carried out for 200 epochs with batch size 64. For Referable Diabetic Retinopathy (RDR), the model obtained 91% sensitivity, 98% specificity and 98% AUC on EyePACS dataset. For

APTOS datasets, 97% specificity, 91% sensitivity and 96% AUC was scored. For VTDR, the model obtained 99%AUC, 98%sensitivity and 93% specificity on EyePACS dataset. For APTOS datasets, 92% specificity, 99% sensitivity and 99% AUC was scored. The researchers [10] are focused to reduce the number of learnable parameters to improve the accuracy for better classification. EyePACS dataset is the main source of dataset, which is downloaded from the kaggle, and the pre-processing such as image resizing, min-max normalization and augmentation are performed. To produce a fixed size output vector and to avoid the problem of resolution reduction, a Spatial Pyramid Pooling Layer (SSP) is fixed between the last convolution layer and first fully connected layer. The data is nonlinear and to deal the non-linear behavior the researchers add NiN, which is a collection of micro networks, called mlpconv layer on the top of the SSP layer. The parametric Relu (PRelu) function helps to overcome the overfitting problem. The architecture was trained on the NVIDIA Tesla K40 and obtained an average recall of 55.6, precision of 67, specificity of 91, and F1 score of 59.6. Additionally, it achieved a micro AUC of 95.0 and a macro AUC of 84.0. To speed up the disease detection, the researchers [11] proposed a model to segment the capillaries abnormalities present in the image and to localize and detect the object. The retinal images were taken from Kaggle competition APTOS 2019 and 2015 with total 23302 images ranging from 0 to 4 classes like No DR, Mild, Moderate, Severe and Proliferative. In the data-augmentation the researchers used flipping method, improves the brightness and contrast. The image size is reduced to 256×256 and Gaussian blur subtraction method was applied to enhance the image to get more fine details from the image. For training, the researchers used ResNet50, DenseNet12, DenseNet169 and DenseNet256. The model is trained with 10 epochs and the optimization technique used is Adam. The best model DenseNet121 got kappa score of 80.02.The researchers implemented a website and the user can upload retina image and predict the result.

K, Harihanth et al [12] created the model, which is trained with green channel of the image with highly imbalanced dataset and the ensemble method with five CNN. The image channels (Red, Green and Blue) are separately passed through the model and the result is analyzed. The EyePACS images that are resized to 512 ×512 were used for training purpose. To maintain the shape of 512x512x3, the channels of the images are separated and then stacked. Data augmentation such as vertical flipping and horizontal flipping was introduced to reduce overfitting. The performance metrics was precision 70%, Recall 51% and F1 score 56% was obtained and the model was compared with another work shows better performance in current model. Red channel, blue channel and green channel confusion matrix are compared and it is noticed that the accuracy of the model using red channel was 0.7489, Blue channel was 0.7386 and green channel was 0.8185.

Table 1: Summary of Existing Models

Research	Model	Work Done	Training Environment	Results
S. S. Karki et al[1]	Ensemble method with Efficient Nets B1, B2, B3, and B5	Radius reduction method to reduce irregularities along the circular border	Tesla K80 GPU	92% quadratic kappa score, EfficientNet B3 performed best
R. S. Rajkumar et al[2]	Transfer Learning with ResNet-50	Image thresholding method to separate object from foreground and background pixels	Tesla V100 GPU	89.4% accuracy, 97% specificity, and 57% sensitivity

Kolla,M.&T,V.[3]	Hardware-friendly Binary Convolutional Neural Network (BCNN) model	Binarized convolution and dense layers, binarized activations and biases to reduce number of parameters	NVIDIA GPU system	37.50% reduction in memory usage, 49.34% gain in run time
N.Barhate, S.Bhave et al[4]	Combination of VGG network and auto encoder	Data augmentation and batch normalization applied, ReLU activation function used in all layers	GPU-based virtual machine on Google Colab	Training accuracy of 96.98% and validation accuracy of 96.69%
A.Singh and W.Kim[5]	Dense Net	OpenCV to convert images to grayscale, Laplacian method to drop blurred images, augmentation applied to increase diversity	Linux environment with ASPEED Technology	Accuracy of 83.6% and kappa score of 88.4% with fivefold cross validation
S.Qummar et al.[6]	CNN ensemble based framework with Xception, Inceptionv3, Resnet50, Dense121 and Dense169	Resizing, up-sampling, flipping, and down-sampling	NVIDIA Tesla k40	Accuracy of 80.8%, recall of 51.5%, specificity of 86.72%, precision of 63.85%, and F1 score of 53.74%
R. N. Lazuardi et al[7]	EfficientNet-B4 and EfficientNet-B5	Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm to improve contrast and image central cropping	Not specified	Kappa score of 0.7922, F1 score of 0.8269, and accuracy of 83.87% for EfficientNet-B4

Proposed Method

A. Dataset Preparation and Preprocessing

Data sets are available openly and for this research purpose, the data set is downloaded from Kaggle and Messidor in to the local system. The data set is cleaned to remove the dark images, blur images, and combine both images to make a proper number of deserts. Datasets and its proper ratio is very much important while training and testing a model. Sample fundus images from kaggle is given in figure 2. In the pre-processing stage, the data set is passed to an algorithm called Contrast Limited Adaptive Histogram Equalization (CLAHE) which takes care the over amplification of the image contrast. The working principle behind the CLAHE is that it operate on very small regions (tiles) in the images. CLAHE is the advanced version of AHE, which is commonly used to improve the image contrast but over amplify the noise. To resolve this problem CLAHE exist in path. Tile generation, histogram equalization, and bilinear interpolation are the basic three parts in CLAHE. 1500 CLAHE No DR and 1500 CLAHE Mild DR images are used for the model creation purpose and these images are a mixture of Kaggle and Messidor datasets. The CLAHE preprocessing stage and the output obtained is shown in figure 3.

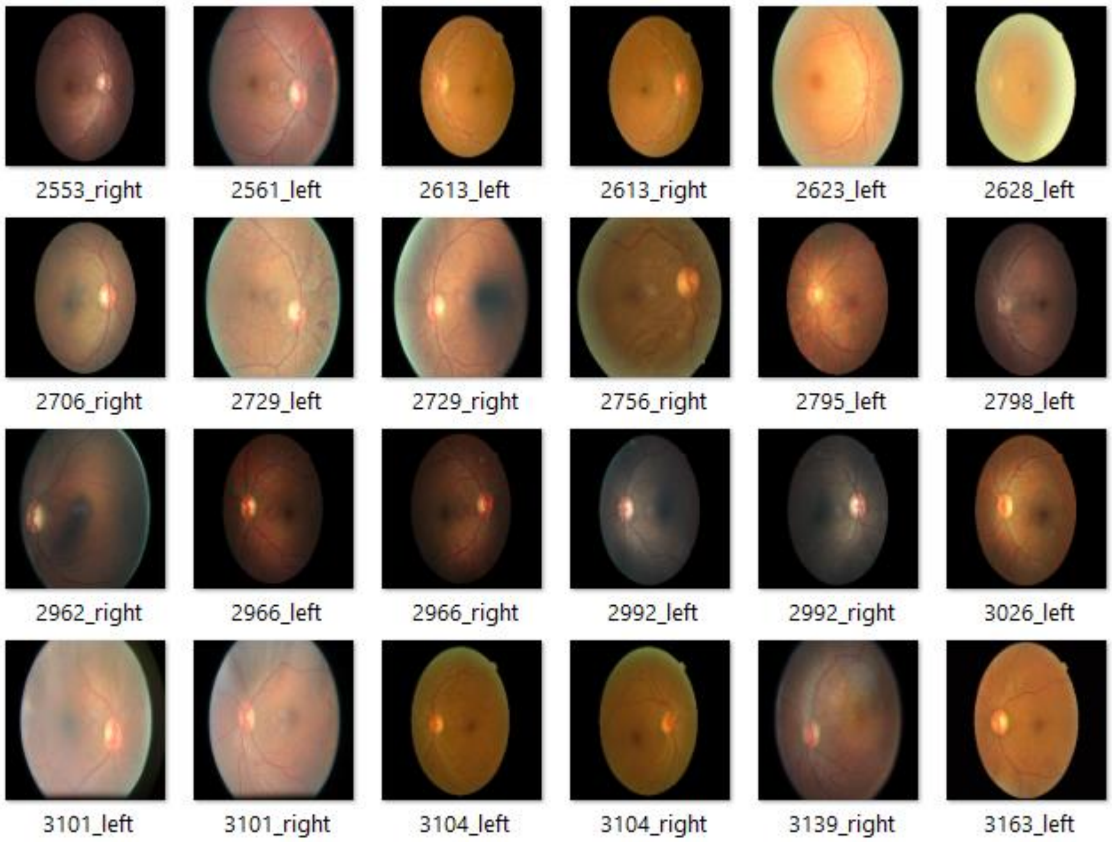

Fig.2. Sample fundus images from Kaggle

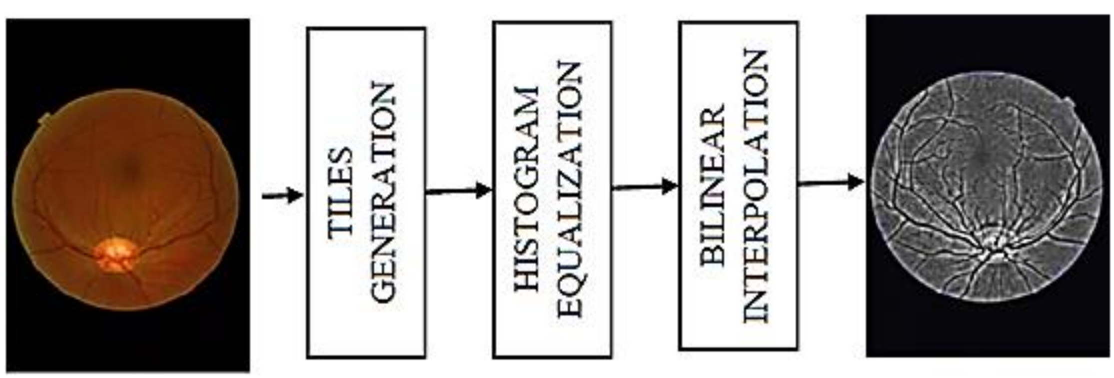

Fig.3. CLAHE Image Processing Steps

B. Proposed Architecture

The Enhanced Minimum Convolutional Neural Network (EMCNN) enjoys the sequential architecture. The model initially have a convolution layer of 64 neurons and it accepted the input image shape of 224×224×1 , then ReLu activation function ,maxpooling with pool size (2,2) and drop out layer is used. Three stack are added like Convolution, Maxpool and Dropout. The final dense layer (Fully connected layer) having two neurons and softmax is the activation function. The model uses 'categorical cross entropy' as the loss function; 'rmsprop' as the optimizer and 'accuracy' is the metrics. The model architecture is shown in figure 4 and have total 603330 trainable parameters.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 64, 224, 224)      640
_________________________________________________________________
activation (Activation)      (None, 64, 224, 224)      0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 32, 112, 224)      0
_________________________________________________________________
dropout (Dropout)            (None, 32, 112, 224)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 10, 37, 128)       258176
_________________________________________________________________
activation_1 (Activation)    (None, 10, 37, 128)       0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 5, 19, 128)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 5, 19, 128)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 1, 6, 256)         295168
_________________________________________________________________
activation_2 (Activation)    (None, 1, 6, 256)         0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 1, 3, 256)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 1, 3, 256)         0
_________________________________________________________________
flatten (Flatten)            (None, 768)               0
_________________________________________________________________
dense (Dense)                (None, 64)                49216
_________________________________________________________________
activation_3 (Activation)    (None, 64)                0
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 130
_________________________________________________________________
activation_4 (Activation)    (None, 2)                 0
=================================================================
Total params: 603,330
Trainable params: 603,330
Non-trainable params: 0
```

Fig.4. EMCNN Model Architecture at Implementation Level

The EMCNN is trained with 3100images. The block diagram is shown in figure 5. The input image size was 224×224×1. The model is implemented with the help of the GPU Nvidia Geforce GTX graphics processor, Jupyter notebook and Tensor flow. Early stopping mechanism is used while training time and batch size 32. The training is done with 20 epochs and acquired good result when compared to the earlier implementation. The training, validation loss and accuracy is shown in the figure 6. In the graph, the X-axis shows the number of epochs and Y-axis shows accuracy. The blue line indicates training phase accuracy and the green line indicates validation time accuracy. Similarly in the case of training loss and validation loss. After 5 epochs training accuracy increases and the training loss decreases gradually. On 20 epochs, the model gave the training accuracy 0.9799 and validation accuracy 0.92.

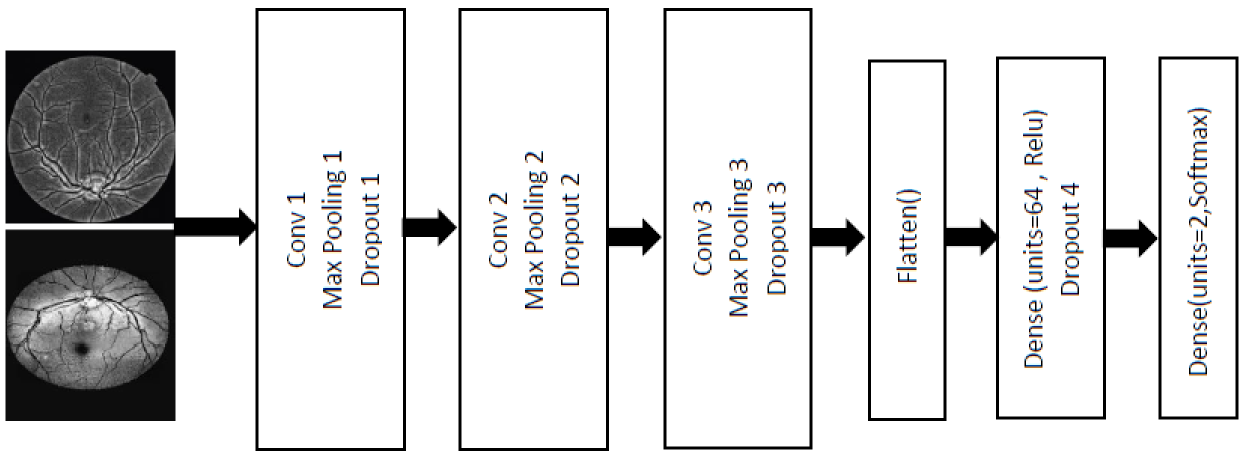

Fig.5. EMCNN Architectural Block Diagram (Flow of Layers)

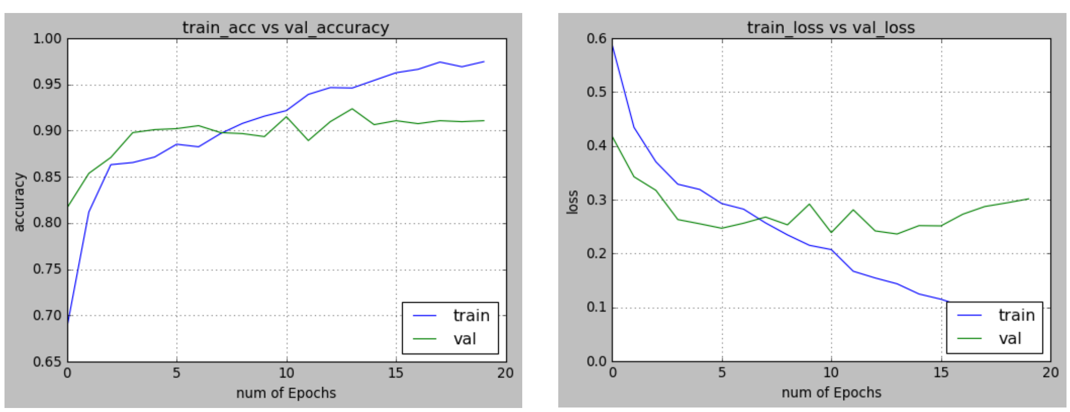

Fig.6. EMCNN Model Training, validation accuracy and loss

Result and Discussions

The model is trained with 3100 fundus CLAHE images out of which 1550 are No-DR images and 1500 are Mild-DR images. The researchers followed 70-30 ratio for choosing training and testing images. The proposed model EMCNN is tested with 930 fundus CLAHE images, out of 930 images, 406 images are recognized as True Positive and 441 images are recognized as True Negative. The results are shown in the form of Confusion Matrix, a table with four different combination of actual and predicted values, figure7. The model shows other parameters like precision, recall and F1 score. The precision is the ratio of true positive to the false positive plus true positive. It can be given as Precision = TP/ (FP+TP). Recall is the ratio of true positive to the total number of actual positive samples. It can be given as Recall= TP/ (FN+TP). F1 score is the weighted average of the given model's precision and recall. The F1 score can be represented by F1 = two*(precision*recall)/ (precision+ recall). The EMCNN gave the values for mild DR, the precision 0.92; recall 0.89 and F1 score 0.91. The value for No-DR, the precision 0.90, recall 0.93 and F1 score 0.91. The values obtained is shown in the figure 8.

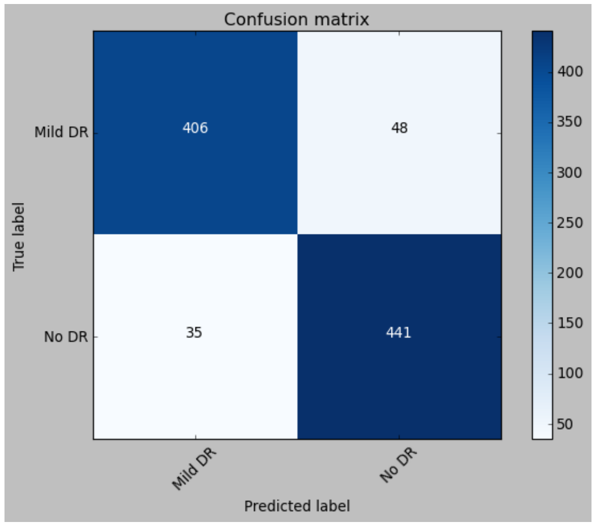

Fig.7. Confusion Matrix obtained from EMCNN Model

```
              precision    recall  f1-score   support

     Mild DR       0.92      0.89      0.91       454
       No DR       0.90      0.93      0.91       476

    accuracy                           0.91       930
   macro avg       0.91      0.91      0.91       930
weighted avg       0.91      0.91      0.91       930

[[406  48]
 [ 35 441]]
Confusion matrix, without normalization
[[406  48]
 [ 35 441]]
```

Fig.8. Precision Recall and F1 score obtained from EMCNN

Receiver Operating Characteristics (ROC) curve is another parameter to define the model performance. ROC curve plots using two parameters True Positive (TP) and False Positive (FP). It can be drawn TP against FP. The graph is shown in figure 9 where, the area under the curve is 0.97 for Mild and No DR classification means the proposed classifier is a strong and excellent one. The table 1 shows the model performance comparison with recently developed models. The Ensemble Approach [6] method has accuracy of 0.80, precision 0.63, recall 0.51 and F1 score 0.53 using Kaggle dataset. The EfficientNet [7] has obtained the accuracy of 0.83 and F1 score 0.82 using the dataset Kaggle. The VGG-NIN [10] obtained the accuracy 95% and Individual channel training [12] model obtained 81%. The table analysis shows that the proposed model is performed better than other existing model with 0.97 accuracy 0.92 precision 0.93 recall and 0.91 F1 score obtained with a mixture of Kaggle and Messidor datasets.

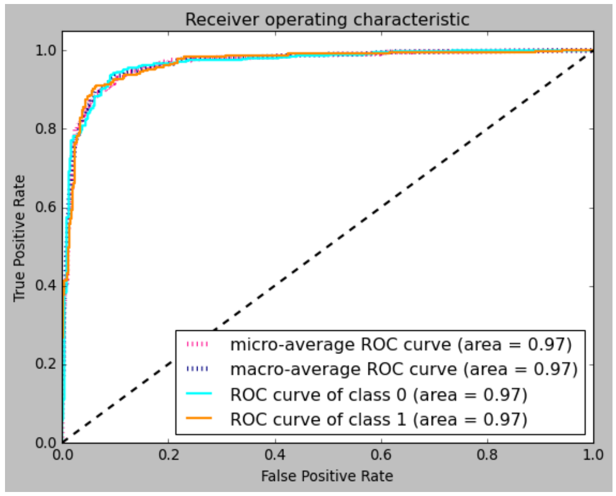

Fig.9. ROC Curve obtained from EMCNN

Table 1: Model Comparison with EMCNN and Existing Models

Model	Accuracy	Precision	Recall	F1 Score	Dataset Used
Ensemble Approach [6]	0.80	0.63	0.51	0.53	Kaggle
EfficientNet [7]	0.83	-	-	0.82	Kaggle
VGG-NIN [10]	0.95	0.67	0.55	0.59	Kaggle
Individual channel training [12]	0.81	0.70	0.51	0.56	Kaggle
EMCNN	**0.97**	**0.92**	**0.93**	**0.91**	**Kaggle+ Messidor**

Conclusion

The work is focused on creating an Experimental Minimum CNN (EMCNN) model to classify Diabetic Retinopathy (DR) stages and detect mild symptoms of DR. The model is created with minimum number of convolution, pooling layers and an experiment is conducted to detect mild DR symptoms and classify Mild DR, No-DR images with minimum effort. Proposed model is trained with fundus images, which are preprocessed with CLAHE algorithm to improve the contrast and reduce the noise. The image sets are resized in to 224×224 before passing to the EMCNN. The proposed EMCNN model gave the accuracy 97% and F1 score of 0.91. The model is compared with existing models and the difference gives excitement. The model employs 3100 photos, which are restricted in number, so that the model may be trained with additional images in the future to enhance accuracy and other parameters.

Reference

1. S. S. Karki and P. Kulkarni, "Diabetic Retinopathy Classification using a Combination of EfficientNets," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 2021, pp. 68-72, doi: 10.1109/ESCI50559.2021.9397035.
2. R. S. Rajkumar, T. Jagathishkumar, D. Ragul and A. G. Selvarani, "Transfer Learning Approach for Diabetic Retinopathy Detection using Residual Network," 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 1189-1193, doi: 10.1109/ICICT50816.2021.9358468.

3. Kolla, M., & T, V. (2021). Efficient Classification of Diabetic Retinopathy using Binary CNN. 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), 244-247.
4. N. Barhate, S. Bhave, R. Bhise, R. G. Sutar and D. C. Karia, "Reducing Overfitting in Diabetic Retinopathy Detection using Transfer Learning," 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), 2020, pp. 298-301, doi: 10.1109/ICCCA49541.2020.9250772.
5. A. Singh and W. Kim, "Detection of diabetic blindness with Deep-Learning," 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2020, pp. 2440-2447, doi: 10.1109/BIBM49941.2020.9313392.
6. S. Qummar et al., "A Deep Learning Ensemble Approach for Diabetic Retinopathy Detection," in IEEE Access, vol. 7, pp. 150530-150539, 2019, doi: 10.1109/ACCESS.2019.2947484.
7. R. N. Lazuardi, N. Abiwinanda, T. H. Suryawan, M. Hanif and A. Handayani, "Automatic Diabetic Retinopathy Classification with EfficientNet," 2020 IEEE REGION 10 CONFERENCE (TENCON), 2020, pp. 756-760, doi: 10.1109/TENCON50793.2020.9293941.
8. K. T. Islam, S. Wijewickrema and S. O'Leary, "Identifying Diabetic Retinopathy from OCT Images using Deep Transfer Learning with Artificial Neural Networks," 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS), 2019, pp. 281-286, doi: 10.1109/CBMS.2019.00066.
9. Chetoui M, Akhloufi MA. Explainable Diabetic Retinopathy using EfficientNET. Annu Int Conf IEEE Eng Med Biol Soc. 2020 Jul;2020:1966-1969. doi: 10.1109/EMBC44109.2020.9175664. PMID: 33018388
10. Khan, Z., Khan, F. G., Khan, A., Rehman, Z. U., Shah, S., Qummar, S., Ali, F., & Pack, S. (2021). Diabetic Retinopathy Detection Using VGG-NIN a Deep Learning Architecture. IEEE Access, 9, 61408-61416. [9409084]. https://doi.org/10.1109/ACCESS.2021.3074422
11. Lands, A., Kottarathil, A.J., Biju, A., Jacob, E.M., & Thomas, S. (2020). Implementation of deep learning based algorithms for diabetic retinopathy classification from fundus images. 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 1028-1032.
12. K, Harihanth and Karthikeyan B. “Diabetic Retinopathy Detection using ensemble deep Learning and Individual Channel Training.” 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS) (2020): 1042-1049.