

## An Early Detection of Diabetic Retinopathy from Fundus Images using Deep Learning

Deva Kumar S<sup>1\*</sup>, Sushma K<sup>2</sup>, Kudravalli Deepika<sup>3</sup>, R Prathap Kumar<sup>4</sup>

<sup>1,4</sup>Department of CSE, VFSTR deemed to be University, Andhra Pradesh, 522213, India

<sup>2</sup>Department of IT, Vignan's Institute of Information Technology, Andhra Pradesh, 530049, India

<sup>3</sup>Department of CSE, Vignan's Lara Institute of Technology & Science, Andhra Pradesh, 522213, India

\*deva.248@gmail.com

### ABSTRACT

Diabetes is the most common disease that affects people who are suffering with high glucose levels in blood. It has many complications such as Neuropathy, Nephropathy, Retinopathy, etc. Among all these complications Diabetic Retinopathy (DR) is the most founded disease in Diabetic patients. DR occurs when the retinal blood vessels in the eye is damaged. Now-a-days DR is the common disease by which blindness occurs globally. The detection of DR manually is expensive, and time taken. As performing retinal screening examinations is a time taken and unmet need on all diabetic patients, automated medical image analysis helps in identifying the disease severity. In this study the image classification using deep learning is explained. The purpose of this work is to develop an automated diagnostic system for Diabetic Retinopathy screening. Here using deep learning, pre-trained models (ResNet50, VGG-16 and VGG-19) are used for DR classification. Pre-trained models are very helpful in detecting the DR easily. KAGGLE Asia Pacific Tele-Ophthalmology Society (APTOS) dataset which consists of 3662 fundus images of five different classes namely Normal, Mild, Moderate, Severe and PDR (Proliferative DR) is used as training data for pre-trained models. Due to class imbalance after applying pre-processing on the images only 3500 images were fed to the network. The model is processed on 3500 fundus images and the output will produce the severity of DR. VGG-19 gained highest accuracy of 89.00% than other models. In this study KAGGLE dataset is taken to detect DR images, In future, we will try different datasets implementing on this model to increase the accuracy in detecting DR.

### Keywords

CNN, Diabetic Retinopathy, Image Classification, ResNet50, VGG-16, VGG-19

### Introduction

Diabetic patients suffer with many complications among which Diabetic Retinopathy (DR) is one. DR can cause blindness to the patients if it is not recognized and treated time. Most of the patients are unaware of this disease and fails to get the best treatment in time. The classification of DR from the fundus images is very expensive and time consuming even for the experienced experts. Hence, computer aided automated diagnosis came into force by which it became easy to identify the DR severity accurately in less time and been helpful in reducing the number of blindness cases. Multiple automated diagnosis system has been developed from past decades. Among all these techniques deep learning achieved a good response. Their usage has been increased as those techniques are good at providing the accurate results of DR classification. Based on deep learning techniques in this study pre-trained models are used (i.e., DenseNet and U-net) for DR classification. The merit of using pre-trained models is that they are already trained on databases by which we do not need to train them from scratch. So as to accommodate the screening and yearly evaluations obligates a huge number of patients; an automatic screening device is a beneficial assistant to diabetes clinics. Presently, there are numerous approaches that could precisely diagnose definite DR associated injuries. As the features which are mentioned above, to find out DR we need to develop an automated technique for the quick valuation of the retinal images.

## Literature Review

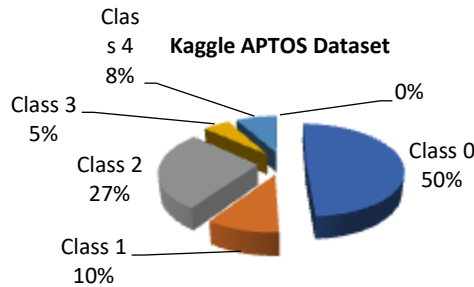
Diabetes is one of the health problems that are found worldwide. Diabetes affects millions of people every year. The disease that affects the fine vessels of blood present in retina is referred as Diabetic Retinopathy (DR) which is found in Diabetic patients. DR is one of the wide spreading disease by which blindness will occur if it's not recognized in its early stage. Only few DR patients are aware of their condition [1]. Although DR is a preventable disease [2], but recognizing it in early stage had become a challenge for ophthalmologists. Most of the DR patients are living in not fully developed areas, the facility to provide specialists and detection is unavailable [3]. DR is categorized into five stages according to its severity scale [4]. DR prevention remains challenging for ophthalmologists because to find out the disease presence in patients directly is time taken and expensive [5]. DR early detection and severity diagnosis also remain subjective, as these were recorded in previous studies [6-7]. As direct examination is somewhat difficult [8] developed a computer-aided system for the fundus images classification. Image classification helps in training the computer with the data by differentiating the image into the prescribed category. [9] proposed a AI based automatic screening for DR and they gained the sensitivity and specificity of DR grading according to ophthalmologists grading. [10] introduced an algorithm to find people with or without DR to access systemic data and DR grading. The above approaches used handcrafted features which will increase the complexity. Thus, deep learning method which can learn features from the fundus images, grabbed the researcher's attention. In [11] fundus-photograph based deep learning algorithms are used for DR detection, as CNN have been taught to recognize the lesions from images, it will be very helpful for detecting DR easily. As DNN brought many breakthroughs in recent years, [12] built a dataset of DR images which are labelled correctly. By this, they trained DNN to grade the severity of DR images and achieved an accuracy of 88.72%. [13] Proposed a Deep CNN for detecting DR lesions. They adopted the CNN structure and detected the DR. They ranked second for good detection result with (FROC) of 0.954 in Kaggle dataset. [14] Proposed a traditional machine learning algorithm with deep CNN. The method achieved 0.94 AUC with 0.93 sensitivity and 0.87 specificity from public datasets.

As computer-aided methods have been used for DR detection, [15] proposed a novel CNN with Siamese like architecture using transfer learning. Different from all other models, this model uses binocular images as inputs and helps in prediction. This model achieved a kappa score of 0.829. CNN models require very less labelled samples for training compared to OCR. In [16] deep learning explanation is given which is useful in processing images, videos, text and speech. In [17] the use of EyePACS is given by which the simplification of images is done. EyePACS acquires retinal images from standard retinal images. It is a license-free web-based DRS system. As the morphological process approaches will increase the complexity for finding the accurate patterns to the images, DL has the ability to process large amount of data and extract meaningful patterns from the fundus images [18]. The detection of objects can be discussed in [19].

## Methodology

As DR is the leading cause of blindness, to overcome this we had introduced an early detection of DR detection using deep neural networks. The objective of this study is to detect DR in its early stage. To detect images easily pre-trained models (VGG-16, VGG-19 and ResNet50) are trained on KAGGLE APTOS dataset which contains 3662 fundus images of five different classes. Class

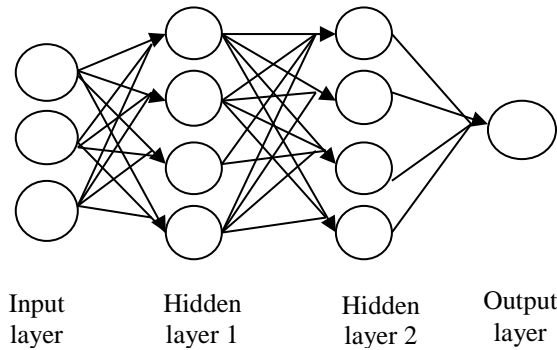
0 is referred as Normal which contains 1805 images, Class 1 is referred as Mild having 370 images, Class 2 is referred as moderate having 999 images, Class 3 is referred as severe having 193 images and Class 4 is referred as PDR having 295 images as shown in the Figure 1. The following sections describes about the methodology that is used in this study. Neural networks, pre-processing, and Methodology is described in below section.



**Figure 1.** Distribution of dataset classes

### Artificial Neural Network

Artificial Neural Networks (ANN) works same as a biological brain, it receives the signal, process it and can signal a neuron connected to it.



**Figure 2.** Basic Neural Network

From the above Figure 2, when an image is given as input to the neural network it will process and produce an output. In between the input and output layers there are some other layers named as hidden layers which are helpful in producing output.

### Pre-processing

The dataset contains total of 3,662 fundus images of five different classes namely Normal, Mild, Moderate, Severe and PDR in which 1805 images are classified as No DR, while 1857 are classified as DR. As the Figure 1 clearly shows that there is a class imbalance problem in the dataset. Due to class imbalance, pre-processing of the image is done. In pre-processing the image

is cropped and resized to  $224 \times 224$ . Later mirroring and rotation of the images is performed. The images which are classified as No DR are only mirrored whereas the images which are classified as DR are mirrored and rotated to 45 and 90 degrees. This is done to make the class balance. After rotations and mirroring, all classes are taken with same number of images. As the dataset contains 3662 images but only 3500 images are taken for processing on the network as each class is divided with 700 images. Out of 3500 images 2800 images for training and 700 images for testing.

### **Pre-trained Models**

Pre-trained models are same as CNNs. In convolution layer image features are extracted and in fully connected layer the features of the images are classified. When we train a CNN on image data, general features are extracted and when we go deep into the network, specific features are extracted. As pre-trained models know the features extraction already, so the training will not start from scratch. This is termed as transfer learning. In the proposed pre-trained models, the transfer learning is applied on pre-trained models except the last layer all layers are frozen and the last layer will be replaced by Dense layer, drop out, global average pooling and applied softmax layer for our own predicting layer. The pre-trained models will further work on the predicting layers which we had added and process the output. By this time usage will be less and can get results accurately.

### **VGG-16**

VGG-16 is a pre-trained model which can process and classify millions of images. VGG-16 has 16 layers. Out of 16 layers 13 layers are convolutional layers. The input of the image size is  $224 \times 224 \times 3$ . After taking the input the set consists of 2 convolutional blocks, it contains 64 channels of  $3 \times 3$  filter size and then applied max pool layer with stride (2,2). Later the image size is reduced to  $112 \times 112$  then applied to two convolutional layers of filter size  $3 \times 3$  and 128 channels followed by max pooling. Next one 3 convolutional layers with image size of  $56 \times 56$ , filter size is  $3 \times 3$  and 256 channels followed by max pooling. After that 2 sets of 3 convolutional layers of  $3 \times 3$  filter size and 512 channels applied followed by the max pooling and then generated three Fully Connected (FC) layers. the transfer learning is applied on pre-trained models except the last layer all layers are frozen and the last layer will be replaced by Dense layer, drop out, global average pooling and applied softmax layer for our own predicting layer. KAGGLE dataset is used as training data for pre-trained In ReLu layer zero is placed in place of negative values. Later pooling is done which is used to reduce the size and then it is passed to the fully connected layer. Up to pooling layer all the layers are frozen. As transfer learning is used the pre-trained models will not start from scratch. The last layer i.e., softmax layer is replaced by our own predicting layers as the softmax layer contains 1000 categories and our model requires only 5 class categories the softmax classification layer is replaced. Now, the pre-trained models will start working on the predicting layers which we had added. The predicting layers which we added are dense, dropout, Global Average Pooling (GAP) and a softmax classifier. A dense layer is used to transfer the data from one neuron to another and then a dropout of 0.3% is done to overcome the overfitting problem. Avg. Pooling is done and then this result will be forwarded to softmax classifier in which the image classification is done and forwarded to fully connected in which the image class is classified and the output is given and the proposed VGG-16 model is as shown in Figure 3. VGG-19 and ResNet50 these two pre-trained models will also classify the images through the same procedure.

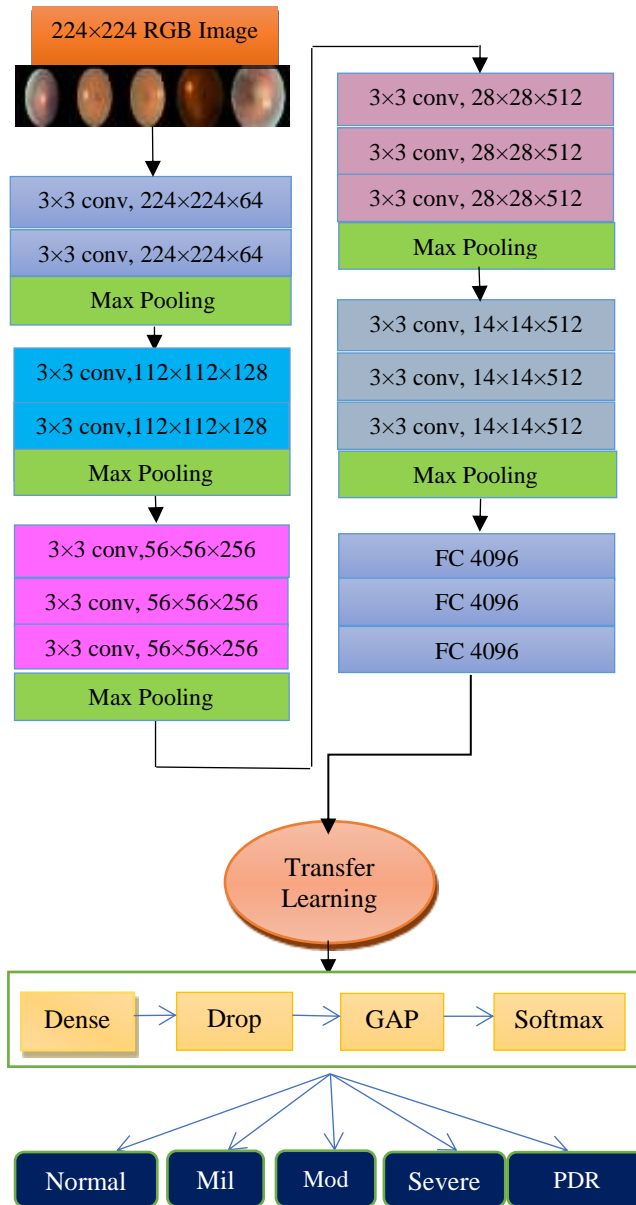


Figure 3. Proposed VGG-16 model

### VGG-19

VGG-19 is a pre-trained model which can process and classify millions of images. VGG-19 has 19 layers. Out of 19 layers 16 layers are convolutional layers. The process is same as the VGG-16. The input of the image size is  $224 \times 224 \times 3$ . After taking the input the set consists of 2 convolutional blocks, it contains 64 channels of  $3 \times 3$  filter size and then applied max pool layer with stride (2,2). Later the image size is reduced to  $112 \times 112$  then applied to two convolutional layers of filter size  $3 \times 3$  and 128 channels followed by max pooling. Next one 4 convolutional layers with image size of  $56 \times 56$ , filter size is  $3 \times 3$  and 256 channels followed by max pooling. After that 2 sets of 4 convolutional layers of  $3 \times 3$  filter size and 512 channels applied followed by the max pooling and then generated three Fully Connected (FC) layers. the transfer learning is applied on pre-trained models except the last layer all layers are frozen and the last layer will be replaced by Dense layer, drop out, global average pooling and applied softmax layer for our own

predicting layer. The proposed VGG-19 model is as shown in Figure 4.

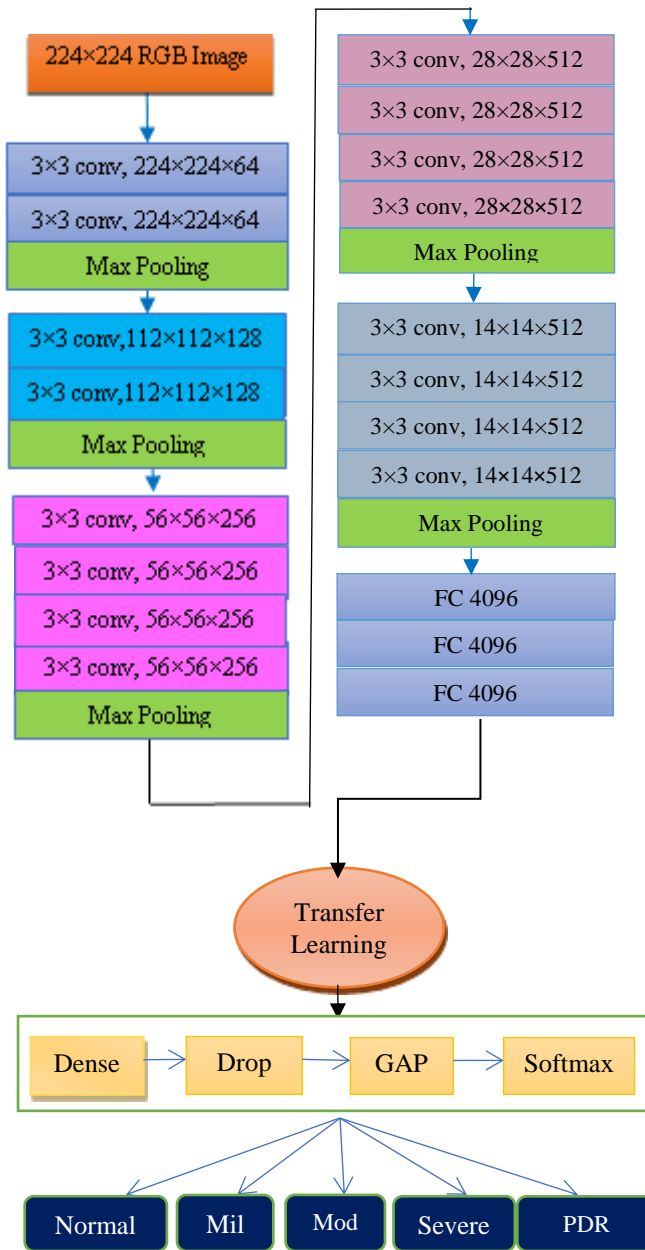


Figure 4. Proposed VGG-19 model

### ResNet50

ResNet-50 stands for Residual Network and it is a convolutional neural network which contains 50 layers. Most of the Computer vision applications can be run through ResNet only and it is backbone of computer vision problems. The ResNet50 architecture having 49 convolutions layers out of 50 layers. These 49 convolutional layers are mainly divided into five blocks and each block having convolutional block and identity block. The input of the image size is  $224 \times 224 \times 3$ . In conv1 the filter size is  $7 \times 7$ , 64 channels applied with stride (2,2). The conv2 have one

convolutional block and two identity blocks. The input parameters for this block has three different filters and channels like  $[[1 \times 1, 64], [3 \times 3, 64], [1 \times 1, 256]]$  applied to the previous conv1 layer output, later it adds the previous conv1 layer output and present conv2 layer output. The conv3 have one convolutional block and 3 identity blocks. The input parameters for this block has three different filters and channels like  $[[1 \times 1, 128], [3 \times 3, 128], [1 \times 1, 512]]$  applied to the previous conv2 layer output, later it adds the previous conv2 layer output and present conv3 layer output. The conv4 have one convolutional block and 5 identity blocks. The input parameters for this block has three different filters and channels like  $[[1 \times 1, 256], [3 \times 3, 256], [1 \times 1, 1024]]$  applied to the previous conv3 layer output, later it adds the previous conv3 layer output and present conv4 layer output. The conv5 have one convolutional block and 2 identity blocks. The input parameters for this block has three different filters and channels like  $[[1 \times 1, 512], [3 \times 3, 512], [1 \times 1, 2048]]$  applied to the previous conv4 layer output, later it adds the previous conv4 layer output and present conv5 layer output. Later applied to the average pool, flatten, and fully connected. The transfer learning is applied on pre-trained models except the last layer all layers are frozen and the last layer will be replaced by Dense layer, drop out, global average pooling and applied softmax layer for our own predicting layer. The modified ResNet50 can be represented in Figure 5.

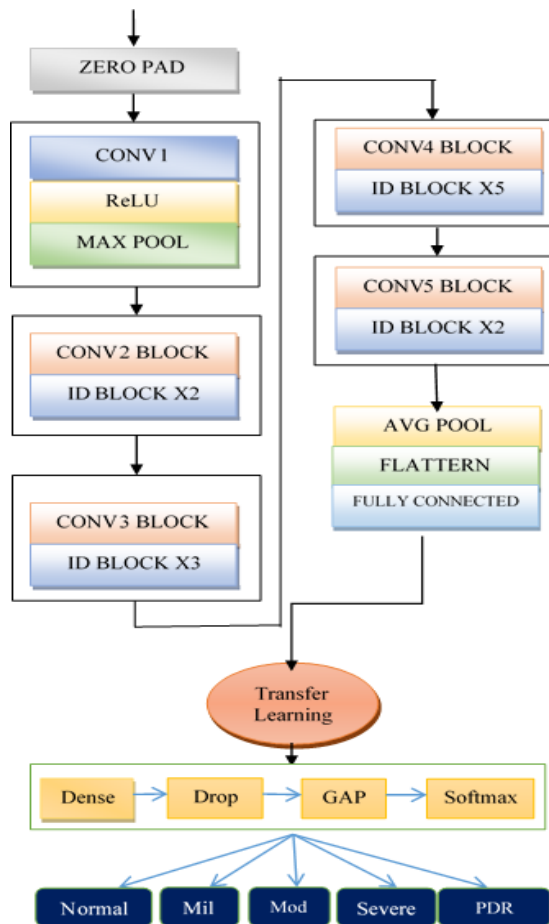


Figure 5. Proposed ResNet50 model  
**Results and Discussions**

The KAGGLE APTOS dataset which contains 3662 images of five different classes is used as training data on pre-trained models. As the dataset contains 3662 images but only 3500 images are taken for processing on the network as each class is divided with 700 images. Out of 3500 images 2800 images for training and 700 images for testing. Anaconda framework is used to implement the network. KAGGLE dataset images are given as inputs and the category of the images is produced as output. Table 1 represents accuracies of three different models.

### Performance Evaluation

The affected cases and the unaffected cases are differentiated through the accuracy test.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

TP (True Positive), FN (False Negative), TN (True Negative), FP (False Positive)

**Table 1.** Proposed models Accuracy

Model	Accuracy	Precision	Recall	F1-score
Proposed VGG-19	89.00%	0.89	0.89	0.89
Proposed VGG-16	88.57%	0.89	0.88	0.88
Proposed RESNET50	81.14%	0.80	0.80	0.80

The dataset accuracies of different pre-trained models are described in this section.

### VGG-19

VGG-19 consists of 19 layers in which convolutional layers, Max. Pooling layers and Fully connected layers are present. It gets an accuracy of 89.00%. The image input size of this model is 224×224. Here, KAGGLE dataset is taken which consists of 3662 images of 5 different classes but only 3500 images are taken for processing on the network as each class is divided with 700 images. Out of 3500 images 2800 images for training and 700 images for testing. Our model contains 5 classes, the pre-trained model is modified by replacing the softmax layer with our own predicting layers as our model requires only 5 class classification. 100 epochs are used for the classification of images and the whole process is done on single CPU system. Figure 6 represents training and testing loss, Figure 7 represents training and testing accuracy, Table 2 represents accuracy levels and Figure 8 represents confusion matrix of VGG-19.

The error that occurred on training data is referred as training loss and the error that occurred after processing the data on neural network is referred as validation loss. Dropout is used when the training loss is less than the validation loss. Here 0.3% of dropout is taken to overcome the overfitting problem of the network as shown in Figure 6.

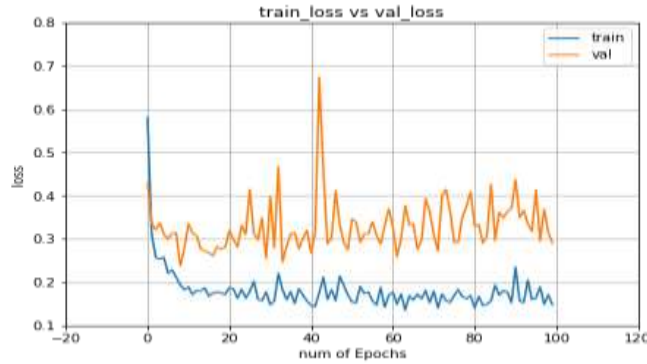


Figure 6. VGG-19 train\_loss vs val\_loss

The accuracy we gained on training data is referred as training accuracy and the accuracy we gained after processing the neural network is referred as validation accuracy as shown in Figure 7.

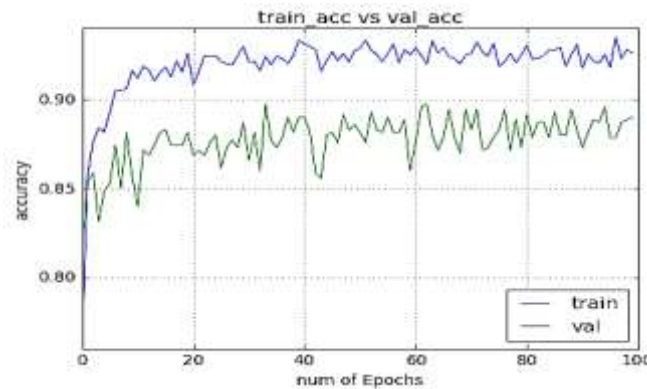


Figure 7. VGG-19 train\_acc vs val\_acc

Here, precision is nothing, but the division is performed between the correct positive values and the positive values that are returned by the classifier. The division is performed between the positive values and the relevant positive samples is termed as Recall. The precision and recall are considered for test's accuracy which is known as f1-score. The positive samples present in the class is termed as Support as shown in Table 2.

Table 2. Accuracy level of proposed VGG-19

	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
<b>0</b>	0.89	0.94	0.92	144
<b>1</b>	0.81	0.80	0.80	136
<b>2</b>	0.97	0.93	0.95	126
<b>3</b>	0.77	0.76	0.77	139
<b>4</b>	0.99	1.00	1.00	155
<b>Accuracy</b>			0.89	700
<b>Macro avg</b>	0.89	0.89	0.89	700
<b>Weighted avg</b>	0.89	0.89	0.89	700

The classification performance is described through confusion matrix as shown in Figure 8.

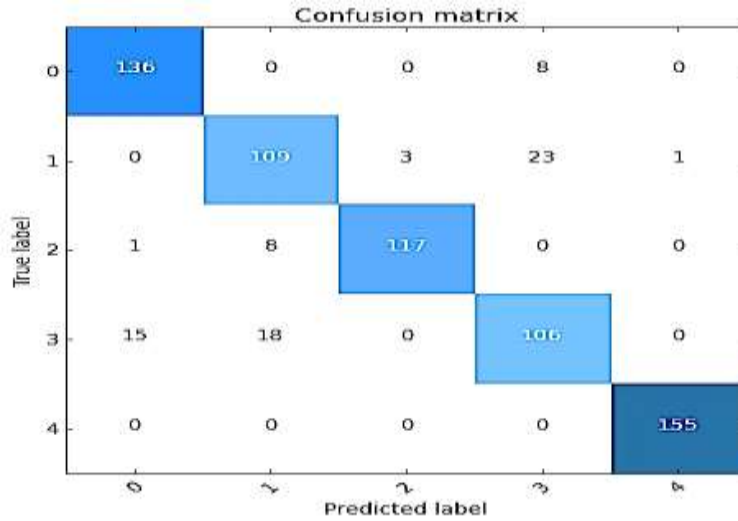


Figure 8. VGG-19 Confusion Matrix

### VGG-16

VGG-16 consists of 16 layers in which convolutional layers, Max. Pooling layers and Fully connected layers are present. It gets an accuracy of 88.57%. The image input size of this model is 224×224. Here, KAGGLE dataset is taken which consists of 3662 images of 5 different classes but only 3500 images are taken for processing on the network as each class is divided with 700 images. Out of 3500 images 2800 images for training and 700 images for testing. Our model contains 5 classes, the pre-trained model is modified by replacing the softmax layer with our own predicting layers as our model requires only 5 class classification. 100 epochs are used for the classification of images and the whole process is done on single CPU system. Figure 9 represents training and testing loss, Figure 10 represents training and testing accuracy, Table 3 represents accuracy levels and Figure 11 represents confusion matrix of VGG-16.

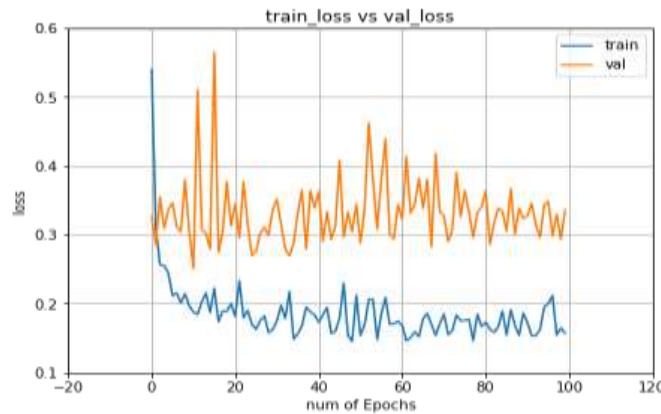


Figure 9. VGG-16 train\_loss vs val\_loss

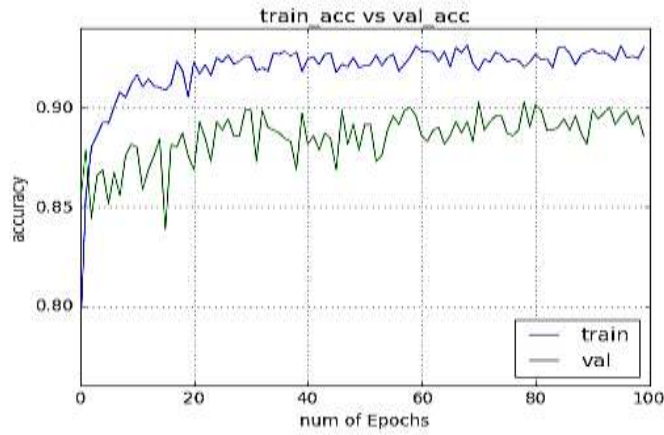


Figure 10. VGG-16 train\_acc vs val\_acc

Table 3. Accuracy level of proposed VGG-16

	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Support</b>
<b>0</b>	0.92	0.92	0.92	144
<b>1</b>	0.73	0.92	0.81	136
<b>2</b>	1.00	0.90	0.95	126
<b>3</b>	0.83	0.68	0.75	139
<b>4</b>	0.99	1.00	1.00	155
<b>Accuracy</b>				0.89
<b>Macro avg</b>	0.89	0.88	0.88	700
<b>Weighted avg</b>	0.89	0.89	0.89	700

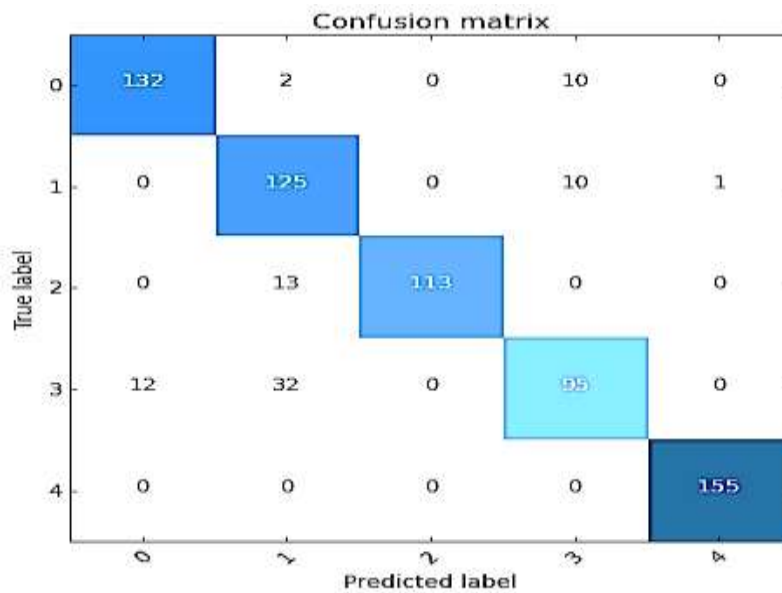


Figure 11. VGG-16 Confusion Matrix

### ResNet50

ResNet50 consists of 50 layers in which the convolution layers, Max. Pool layer and Avg. Pool layers are present. It gets an accuracy of 81.14%. The image input size of this model is  $224 \times 224$ . Here, KAGGLE dataset is taken which consists of 3662 images of 5 different classes but only 3500 images are taken for processing on the network as each class is divided with 700 images. Out of 3500 images 2800 images for training and 700 images for testing. Our model contains 5 classes, the pre-trained model is modified by replacing the softmax layer with our own predicting layers as our model requires only 5 class classification. 100 epochs are used for the classification of images and the whole process is done on single CPU system. Figure 12 represents training and testing loss, Figure 13 represents training and testing accuracy, Table 4 represents accuracy levels and Figure 14 represents confusion matrix of ResNet50.

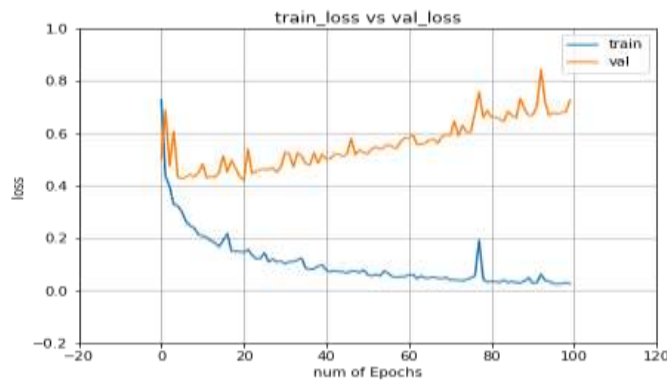


Figure 12. ResNet50 train\_loss vs val\_loss

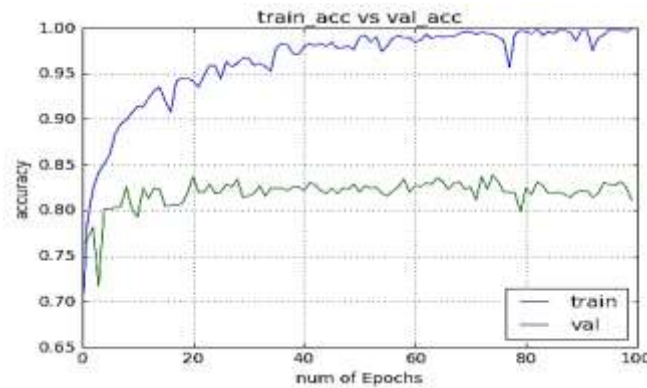


Figure 13. ResNet50 train\_acc vs val\_acc

Table 4. Accuracy level of ResNet50

	Precision	Recall	F1-score	Support
<b>0</b>	0.92	0.91	0.91	144
<b>1</b>	0.58	0.74	0.65	136
<b>2</b>	0.61	0.43	0.50	126
<b>3</b>	0.92	0.91	0.92	139
<b>4</b>	0.99	1.00	1.00	155
<b>Accuracy</b>			0.81	700
<b>Macro avg</b>	0.80	0.80	0.80	700
<b>Weighted avg</b>	0.81	0.81	0.81	700

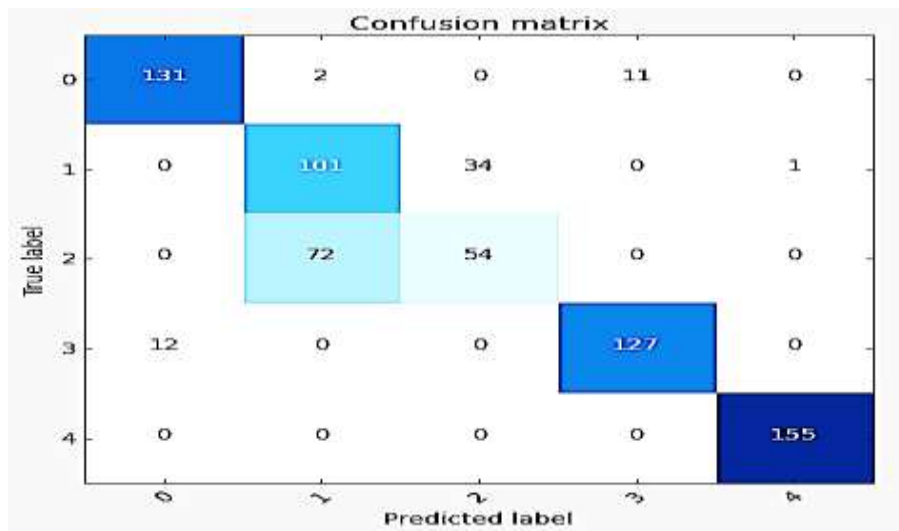


Figure 14. ResNet50 Confusion Matrix

The main objective is to identify DR images which will be helpful for ophthalmologists. As three different pre-trained models were trained on this network among them VGG-19 gets highest accuracy of 89.00% using very less time. The computational time of the other two models VGG-16 and ResNet50 is very high compared to VGG-19.

### Conclusion

As DR has eventually been found in maximum of Diabetic patients, its detection through manually is a time taken process. As the prevention of DR is a challenge for ophthalmologists, to overcome this drawback and to detect DR in a very less time an early detection of DR using deep neural networks is proposed in this study. VGG-19 gained highest accuracy of 89.00% than other models. KAGGLE dataset is used which consists of 3662 fundus images out of which only 3500 images are taken for processing on the network as each class is divided with 700 images. The merit of this model is due to pre-trained models the computational time is very less and can accurately detect DR. This model will be helpful for ophthalmologists in detecting DR within a

less span of time. In future, we will try different datasets implementing on this model to increase the accuracy in detecting DR.

## References

- [1] J. Beagley, L. Guariguata, C. Weil, and A. A. Motala, "Global estimates of undiagnosed diabetes in adults," *Diabetes Res. Clin. Pract.*, vol. 103, no. 2, pp. 150–160, 2014. <http://dx.doi.org/10.1016/j.diabres.2013.11.001>
- [2] C. P. Wilkinson *et al.*, "Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales," *Ophthalmology*, vol. 110, no. 9, pp. 1677–1682, 2003. doi:10.1016/S0161-6420(03)00475-5
- [3] L. Guariguata, D. R. Whiting, I. Hambleton, and J. Beagley, "Global estimates of diabetes prevalence for 2013 and projections for 2035," *Diabetes Res. Clin. Pract.*, vol. 103, no. 2, pp. 137–149, 2013. <http://dx.doi.org/10.1016/j.diabres.2013.11.002>
- [4] V. Gulshan *et al.*, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *JAMA - J. Am. Med. Assoc.*, vol. 316, no. 22, pp. 2402–2410, 2016. doi:10.1001/jama.2016.17216
- [5] M. N. Ozieh, K. G. Bishu, C. E. Dismuke, and L. E. Egede, "Trends in health care expenditure in U.S. adults with Diabetes: 2002-2011," *Diabetes Care*, vol. 38, no. 10, pp. 1844–1851, 2015. DOI: 10.2337/dc15-0369
- [6] L. Sellahewa, C. Simpson, P. Maharajan, J. Duffy, and I. Idris, "Grader agreement, and sensitivity and specificity of digital photography in a community optometry-based diabetic eye screening program," *Clin. Ophthalmol.*, vol. 8, pp. 1345–1349, 2014. <http://dx.doi.org/10.2147/OPHTH.S61483>
- [7] P. Ruamviboonsuk, N. Wongcumchang, P. Surawongsin, E. Panyawatananukul, and M. Tiensuwan, "Screening for diabetic retinopathy in rural area using single-field, digital fundus images," *J. Med. Assoc. Thai.*, vol. 88, no. 2, pp. 176–180, 2005.
- [8] L. Seoud, J. Chelbi, and F. Cheriet, "Automatic Grading of Diabetic Retinopathy on a Public Database," pp. 97–104, 2017. <https://doi.org/10.17077/omia.1032>
- [9] J. He *et al.*, "Artificial intelligence-based screening for diabetic retinopathy at community hospital," *Eye*, 2019. <https://doi.org/10.1038/s41433-019-0562-4>
- [10] R. Casanova, S. Saldana, E. Y. Chew, R. P. Danis, C. M. Greven, and W. T. Ambrosius, "Application of random forests methods to diabetic retinopathy classification analyses," *PLoS One*, vol. 9, no. 6, pp. 1–8, 2014. doi:10.1371/journal.pone.0098587
- [11] R. Raman, S. Srinivasan, S. Virmani, S. Sivaprasad, and R. Rajalakshmi, "Fundus photograph-based deep learning algorithms in detecting diabetic retinopathy," *Eye*, 2018. <https://doi.org/10.1038/s41433-018-0269-y>
- [12] Z. Gao, J. Li, J. Guo, Y. Chen, Z. Yi, and J. Zhong, "Diagnosis of Diabetic Retinopathy Using Deep Neural Networks," *IEEE Access*, vol. 7, pp. 3360–3370, 2019. Digital Object Identifier 10.1109/ACCESS.2018.2888639
- [13] G. Quellec, K. Charrière, Y. Boudi, B. Cochener, and M. Lamard, "Deep image mining for diabetic retinopathy screening," *Med. Image Anal.*, vol. 39, pp. 178–193, 2017. doi: 10.1016/j.media.2017.04.012
- [14] R. Gargeya and T. Leng, "Automated Identification of Diabetic Retinopathy Using Deep Learning," *Ophthalmology*, vol. 124, no. 7, pp. 962–969, 2017.

<http://dx.doi.org/10.1016/j.opthta.2017.02.008>

- [15] X. Zeng, H. Chen, Y. Luo, and W. Ye, “Automated diabetic retinopathy detection based on binocular siamese-like convolutional neural network,” *IEEE Access*, vol. 7, pp. 30744–30753, 2019. Digital Object Identifier 10.1109/ACCESS.2019.2903171
- [16] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. doi:10.1038/nature14539
- [17] J. Cuadros and G. Bresnick, “EyePACS: An adaptable telemedicine system for diabetic retinopathy screening,” *J. Diabetes Sci. Technol.*, vol. 3, no. 3, pp. 509–516, 2009.
- [18] R. A. P. Karthigaikumar, “Multi-retinal disease classification by reduced deep learning features,” *Neural Comput. Appl.*, 2015.
- [19] Salluri, D. K., Bade, K., & Madala, G. Object detection using convolutional neural networks for natural disaster recovery. *International Journal of Safety and Security Engineering*, 10(2), 285–291. <https://doi.org/10.18280/ijssse.100217>, 2020