

Diabetic Retinopathy Detection Using Machine Learning Techniques and Transfer Learning Approach



Avanti Vartak and Sangeetha Prasanna Ram

Abstract Diabetic retinopathy (DR) is a diseased condition of eyes which arises due to prolonged diabetes. It could result in loss of eyesight if not identified and handled in time. Diabetic retinopathy manifests itself as non-proliferative diabetic retinopathy (NPDR) which is the earlier stage and proliferative diabetic retinopathy (PDR) which is the advanced stage. In this study, a machine learning model has been developed that classifies a given fundus image as normal, NPDR, or PDR. Initially, machine learning algorithms like decision trees, Naive Bayes, random forest, K-nearest neighbor (KNN), and support vector machine (SVM) were applied for binary classification, but the classification accuracy was less. So later, we employed transfer learning techniques such as ResNet-50, VGG16, and EfficientNetB0 for binary classification which gave high validation accuracy. Then the above-mentioned transfer learning techniques were further used for multiclass classification which gave very good validation accuracy in tune with the existing research in this field.

Keywords Diabetic retinopathy · Machine learning · Image processing · Confusion matrix · Transfer learning · Visual Geometry Group (VGG16) · Residual neural network (ResNet-50) · EfficientNetB0

1 Introduction

Diabetic retinopathy (DR) is one of the serious eye diseases. The Union Health Ministry's first National Diabetes and Diabetic Retinopathy Survey (2015–19) revealed that the prevalence of diabetic retinopathy in India is 16.9%, while that of sight-threatening DR is 3.6% [1]. Detection and treatment of DR at an initial stage may avoid harmful consequences in the future. Diabetic retinopathy is classified into

A. Vartak (✉) · S. P. Ram
Vivekanand Education Society's Institute of Technology, Mumbai, India
e-mail: 2019avanti.vartak@ves.ac.in

S. P. Ram
e-mail: sangeeta.prasannaram@ves.ac.in

four stages, Stage 1—mild DR, Stage 2—moderate DR, Stage 3—severe DR, and Stage 4—proliferative DR (PDR) [2]. The first three stages are collectively termed as non-proliferative DR (NPDR). It is essential to treat the disease early to avoid severe complications. Hence, we developed a machine learning model that can classify the image as normal, suffering from NPDR, or suffering from PDR.

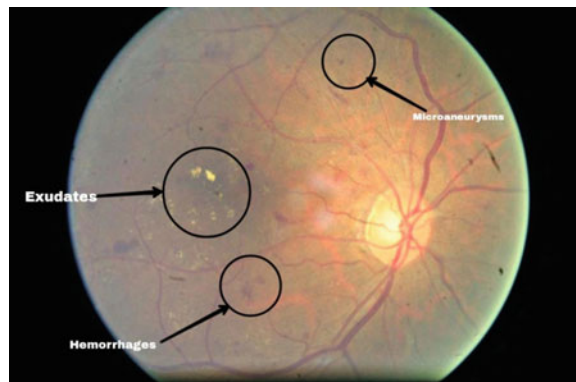
Fundus photography is a procedure of obtaining images of the inner eye through the pupil. A fundus camera is an exclusive low-power microscope connected to a camera. It is used to diagnose internal eye structures such as the optic disc, the retina, and lens. Figure 1 represents fundus photograph of normal fundus eye images. Figure 2 represents diabetic retinopathy fundus eye image indicating microaneurysms, exudates, and hemorrhages.

Mild NPDR: This is the primary stage of DR. It is also termed as background retinopathy. The minute blood vessels in the retina start developing small bulges at this stage. These bulges are also called microaneurysms. They might cause the blood vessels to leak small amounts of blood into retina.

Fig. 1 Normal fundus eye image



Fig. 2 Diabetic retinopathy fundus eye image indicating microaneurysms, exudates, and hemorrhages



Moderate NPDR: This is the second stage of the disease. At this stage, the retinal blood vessels start swelling. It might affect their blood-carrying capacity. Physical changes in the retina and hard exudates can be observed.

Severe NPDR: At this stage, the blockages in the blood vessels increase leading to a reduced blood supply to the retina. The insufficiency of blood activates a signal to the retina to generate new blood vessels. Reaching this stage of the disease indicates a high chance of vision loss. Medical treatment could stop further vision loss. But if some of the vision is already lost, it is impossible to get it back.

Proliferative Diabetic Retinopathy: At this stage, fresh blood vessels start developing in the retina. Since the newly developed blood vessels are fragile and thin, they start bleeding. PDR shall result in vitreous hemorrhage or retinal detachment.

2 Related Work

In a research paper written by Amol et al. [3], the use of multilayer perception neural network (MLPNN) to detect diabetic retinopathy in retinal images was put forth. Swati et al. [4] performed a comparative analysis of KNN and SVM classifier and obtained an accuracy of 85.60% for SVM classifier. Yashal Shakti Kanungo [5] got the best results for DR classification using Inceptionv3 transfer learning model. Researchers such as Kranthi et al. [6] and Vidya et al. [7] have used preprocessed images to train machine learning models using SVM, KNN, and artificial neural network (ANN). Mohamed Chetoui et al. [8] obtained accuracy of 0.904 using support vector machine with a radial basis function kernel. Kaur et al. [9] generated a neural network model and compared its performance with the existing support vector machine classification (SVM) model. The neural network worked better than SVM. Sonali et al. [10] have firstly segmented the optic disc and retinal nerves and then extracted the features using gray-level co-occurrence matrix (GLCM) method. Robiul Islam [11] developed a deep learning model with transfer learning from VGG16 model followed by a novel color version preprocessing technique. Revathy et al. [12] performed image preprocessing using image processing techniques like color space conversion and zero padding. These were followed by median filtering and adaptive histogram equalization. This process was followed by image segmentation and feature extraction. Classification was done using a classifier which was a combination of KNN, random forest, and SVM. The accuracy of this model resulted to be 82%. Out of these three models, the best result was obtained with the SVM model. The accuracy of this model was 87.5%. Satwik et al. [13] used transfer learning methods to detect diabetic retinopathy. Pre-trained models, namely SERes-NeXt32x4d and EfficientNetB3 were used and accuracy obtained was 85.13% and 91.42%, respectively. Ayala et al. [14] implemented a transfer learning model using DenseNet. For this purpose, they used two publically available datasets, APTOS and Messidor. The accuracy obtained for these datasets was 81% and 64%, respectively. Rajkumar et al. [15] also used transfer learning technique, and the pre-trained model used was ResNet-50. The accuracy of the model resulted to be 89.4%.

From the literature survey, it was observed that in the earlier years, the research work in this field was limited only to traditional machine learning algorithms. However, in the recent past, techniques such as neural network and transfer learning have been implemented. We also noticed that some of the researchers have trained the machine learning models without preprocessing the images and some have limited themselves to only one or two transfer learning approaches. Considering these limitations of the existing work, we proceeded with developing a methodology that employed preprocessing of images followed by application of three transfer learning techniques and their comparative study as explained in detail in the proposed work.

3 Proposed Work

3.1 Dataset

The dataset plays a vital role in training a machine learning model. The images we used for training the model originally belonged to the diabetic retinopathy detection dataset provided by EyePACS, a free platform for retinopathy screening, and percent of the images in our dataset belonged to the Aptos 2019 Blindness Detection dataset. These datasets were available on the official Kaggle website.

The former comprised 35,126 images with 708 proliferative, 873 severe, 5,292 moderate, 2443 mild, and 25,810 normal eye images.

The latter comprised 3,662 fundus eye images, where 295 images belonged to the proliferative stage, 193 images to severe, 999 to moderate, 370 images to mild, and 1,805 images to normal eye images.

We observed that the datasets are highly imbalanced, and hence, training the machine learning model using these datasets would result in biased results. In an attempt to avoid this, we decided to create a balanced dataset by choosing equal number of clear images per category from the available dataset. The dataset created for binary classification consisted of 2000 images associated to two separate classes, normal eye images and diabetic retinopathy eye images. The dataset generated for multiclass classification consisted of 3000 images belonging to three distinct classes, normal, NPDR, and PDR. Thus, we successfully generated a balanced dataset for training our model.

3.2 Preprocessing

The original dataset consisted of color fundus eye images. As mentioned in the literature review, many researchers had performed image preprocessing before using the images for training the model. Hence, we used various image processing techniques

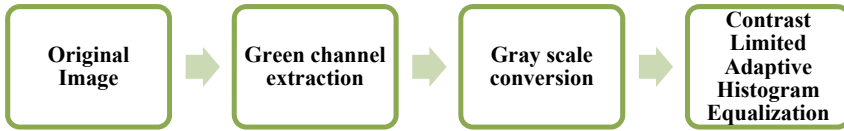


Fig. 3 Machine learning algorithm implementation using Python

and finalized a sequence which suited best for the images in our dataset. Machine learning algorithm implementation using Python is shown in Fig. 3.

An RGB color fundus photograph comprises three channels: red, green, and blue. The original image was split into these components before processing it further. For the sake of visualizing the actual colors, we retained all the channels and set the values of the other channels to zero. Thus, we could obtain each channel separately. In order to convert these images into grayscale, we have used a function ‘split’ which is available in the openCV Python library. We have used green channel grayscale images for further processing as they displayed the best background contrast between the optic disc and retinal tissue. We observed that these images showed a better contrast than the red and blue channel images. Also, they were slightly better than the grayscale images obtained directly from the RGB image. Hence, the green component grayscale images were used for further processing. To enhance the contrast, we used a technique named as contrast limited adaptive histogram equalization (CLAHE). Image processing of eye images is shown in Fig. 4.

3.3 Machine Learning Techniques

To develop a binary classification model, we used different machine learning techniques and followed subsequent steps for implementing those using Python as shown in Fig. 5.

The procedure of image preprocessing has been mentioned above. Later on, we have fitted various machine learning algorithms to our training set.

For evaluating our results, we have observed the test accuracy and plotted a confusion matrix for each algorithm. With the help of the confusion matrix, we could easily understand the number of images that were classified correctly. It was also helpful in understanding biased results.

The labels of confusion matrix were divided into following categories.

True Negative: Model predicted the value as No DR, and the actual value is also No DR.

True Positive: Model predicted DR, and the real value is also DR. **False Negative:** Model predicted No DR, but the actual value was DR.

False Positive: The model predicted DR, but the real value was No DR [16].

Decision tree classification is a predictive modeling tool used in various sectors. We can create decision trees with an algorithmic approach which can split the dataset

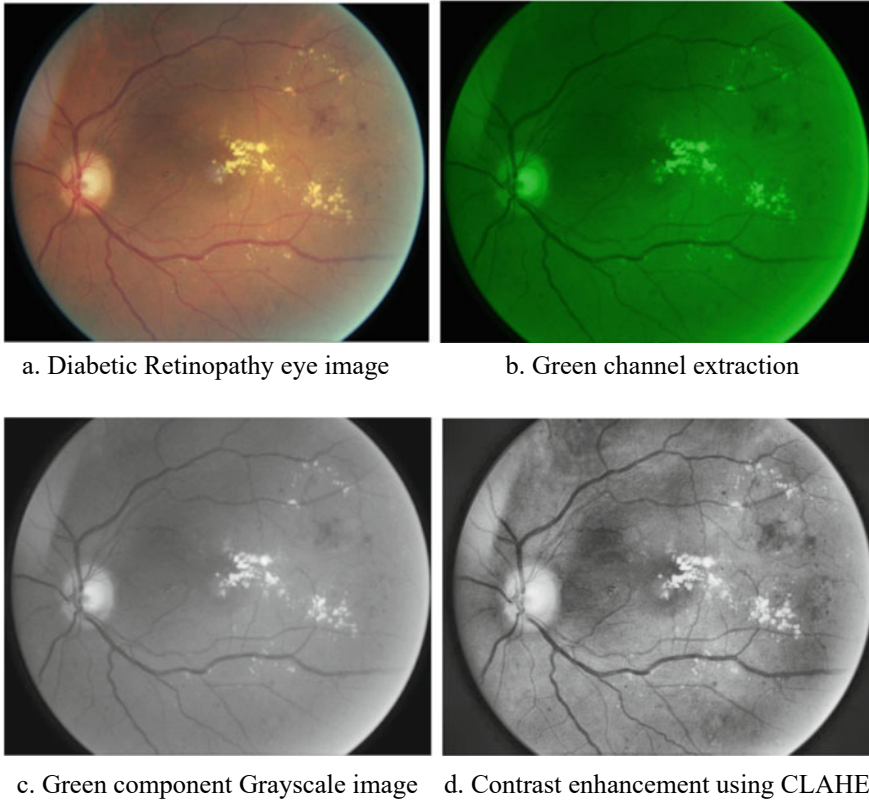


Fig. 4 Image preprocessing

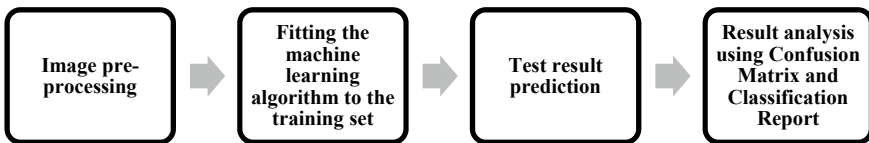


Fig. 5 Machine learning algorithm implementation using Python

in numerous ways based on distinct states. We obtained a testing accuracy of 56.0% for this model. Naïve Bayes classification is a technique derived from Bayes’ theorem where it is considered that all the predictors are self-reliant. Here, the assumption is that the existence of a feature in a category does not rely on any other feature in the same category. The testing accuracy obtained by us for Naïve Bayes’ classification model was 60.5%.

The K-nearest neighbor algorithm considers the correlation between present classes and the new data. It puts the new data in the class that is almost alike to

the present classes. KNN saves the entire data and classifies a new data point on the basis of resemblance. It means whenever a new data arrives, they may be effortlessly classified into a perfectly matched class with the help of KNN classifier. We obtained a testing accuracy of 63.75% for this classification model. Firstly, random forest classifier generates decision trees on data samples. In addition, it receives the prediction from all of them. At the end, it chooses an appropriate answer through voting. Also, it minimizes the overfitting by averaging the result. The testing accuracy obtained by us for this model was 65.25%. An SVM model represents different categories in a hyperplane in multi-dimensional space. The algorithm creates the hyperplane iteratively to reduce the error. This classifier aims to split the dataset in different categories to detect a maximum marginal hyperplane. We obtained a testing accuracy of 67.5% for SVM binary classification model. Confusion matrix of different classifiers is shown in Figs. 6, 7, 8, 9, and 10.

Result analysis of the machine learning models is shown in Fig. 11. Since the highest testing accuracy obtained using machine learning techniques was only 67.5%, we approached the transfer learning techniques.

Fig. 6 Confusion matrix for decision tree classifier

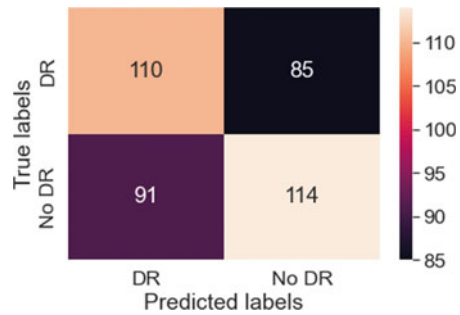


Fig. 7 Confusion matrix for Naive Bayes classifier

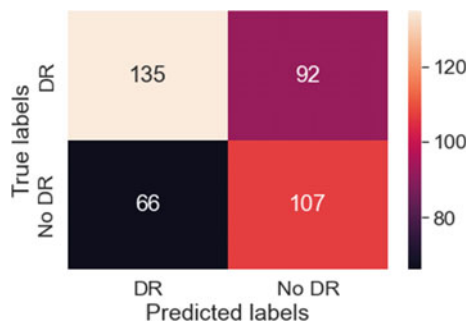


Fig. 8 Confusion matrix for KNN classifier

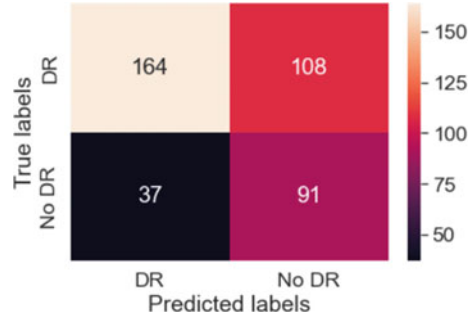


Fig. 9 Confusion matrix for random forest classifier



Fig. 10 Confusion matrix for SVM classifier



3.4 Transfer Learning Techniques

A process where a neural network model is firstly trained on some problem alike to the problem that is to be resolved is called as transfer learning [17]. It is majorly used for problems where the dataset consists of limited data for training a model from scratch. In order to implement the transfer learning technique, we used three pre-trained models namely ResNet-50, VGG16, and EfficientNetB0. We used these pre-trained models for binary as well as multiclass classification.

Following steps were followed for the implementation of transfer learning using Python. Firstly, we imported all the necessary Python libraries and loaded the pre-trained model. The arguments used here were 'input_shape', 'weights', and 'include_

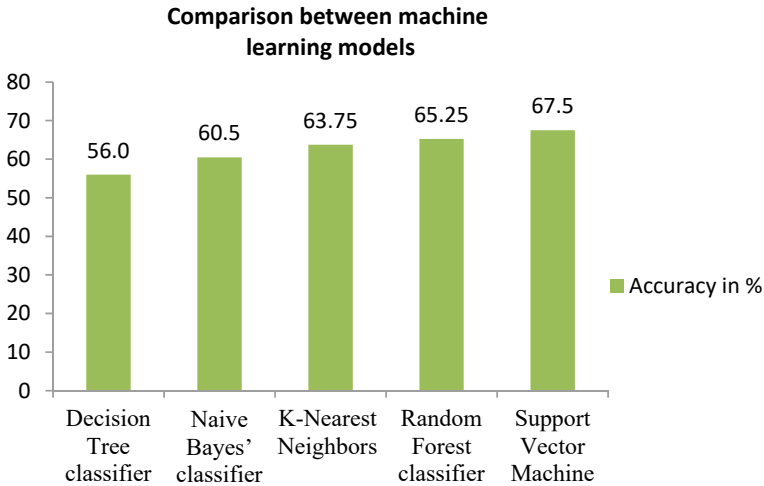


Fig. 11 Result analysis of the machine learning models

top'. Here, we used the weights of the ImageNet database, and the input shape was 224×224 pixels. The parameter 'include_top' was set as false for removing the last layer from the model. This ensured that we could add our own input and output layers according to our custom data. Since the existing layers of models are already trained, we do not have to train them again. Hence, the parameter 'trainable' in the model layers is set as false, and these layers are frozen. If we skip this step, then the model will not be able to give good accuracy because this pre-trained model is already trained on many images. Hence, it is necessary to set the parameter 'layer.trainable' as false as it ensures that the model does not learn the weights once again. This saves space complexity and time. The next step is flattening, where the entire data is converted to a one-dimensional array. We added two dense layers with the activation function 'ReLU'. We used the 'softmax' activation function for the output layer. Here, the number of nodes implies the number of classes. The number of classes was 2 for binary classification, and it was 3 for multiclass classification. Process flow of transfer learning implementation using Python is shown in Fig. 12.

For compiling the model, 'categorical_crossentropy' was used as the loss function, and the optimizer used was 'Adam'. Then we used ImageDataGenerator class for data augmentation. It is an approach to grow the distinctiveness of the training data by applying arbitrary transformations on the images. The next step is fitting the model to our dataset. Here, we trained the model using our training dataset which consisted of fundus eye images. The process took place for ten epochs.

Once the training is completed and the testing accuracy of the model is obtained, analyzing the obtained results is crucial. We did so with the help of a confusion matrix and classification report. As mentioned earlier, we have used the ResNet-50, EfficientNetB0, and VGG16 algorithms. They have been pre-trained on the ImageNet image database.

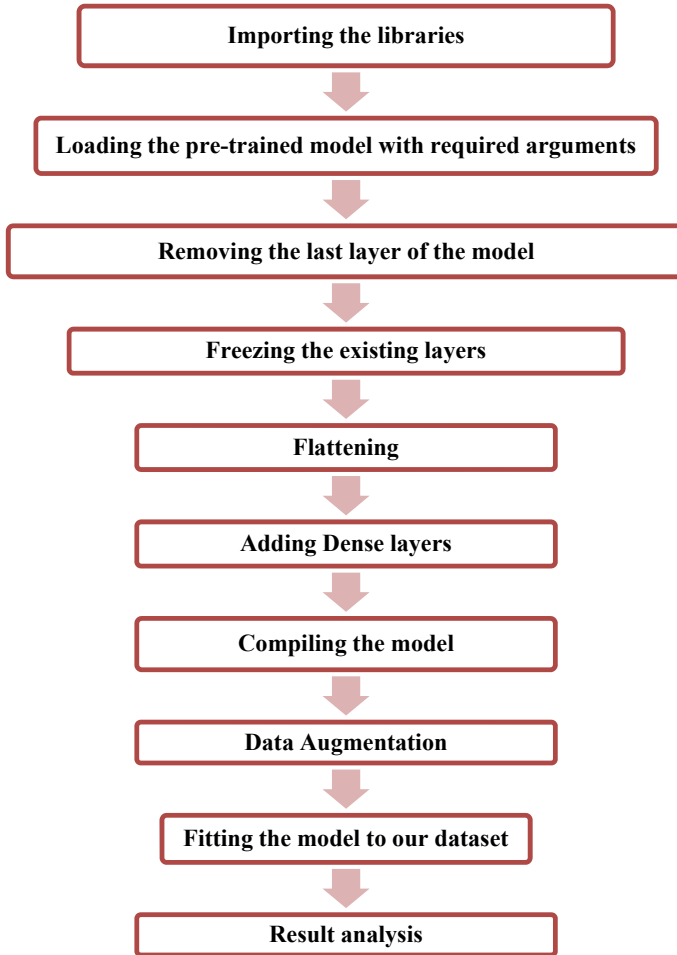


Fig. 12 Transfer learning implementation using Python

ResNet-50 is a convolutional neural network. It is fifty layers deep. ResNet is used to overcome the vanishing gradient problem which was a major disadvantage of convolutional neural networks.

EfficientNet comes up with a family of models (B0–B7) that represents a fine combination of efficiency with accuracy.

VGG16 is a convolutional neural network which has 13 convolutional layers and 3 dense layers. The VGG architecture looks close to the actual convolutional networks. The major thought for VGG was to construct the network deeper by stacking additional convolutional layers. This was implemented by limiting the dimensions of the convolutional windows to 3×3 pixels.

4 Result Analysis

4.1 Binary Classification

Out of the 200 DR images in the testing dataset, the ResNet-50 algorithm could correctly detect 174 DR images. On the other hand, out of 200 normal eye images, 177 images were detected correctly. The accuracy obtained here was 87.75%.

Similarly, the accuracy obtained for EfficientNetB0 was 90.75%, and for VGG16, it was 90%. Confusion matrix of binary classification model is shown in Figs. 13, 14, and 15.

Fig. 13 Confusion matrix for ResNet-50 binary classification model

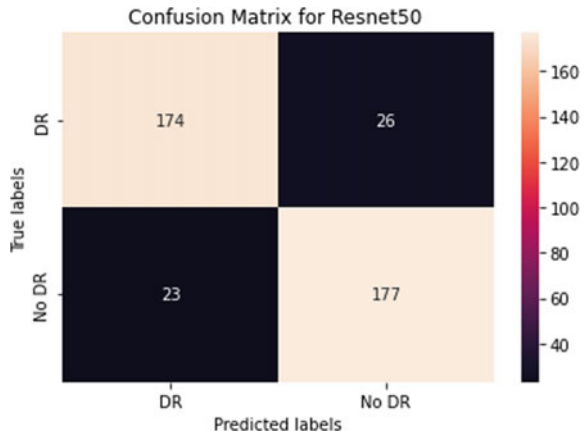


Fig. 14 Confusion matrix for EfficientNetB0 binary classification model

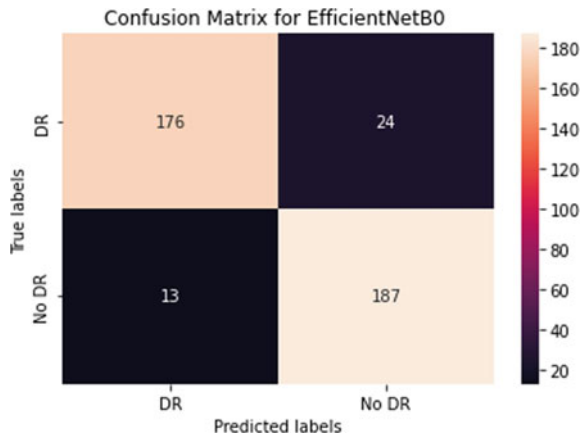
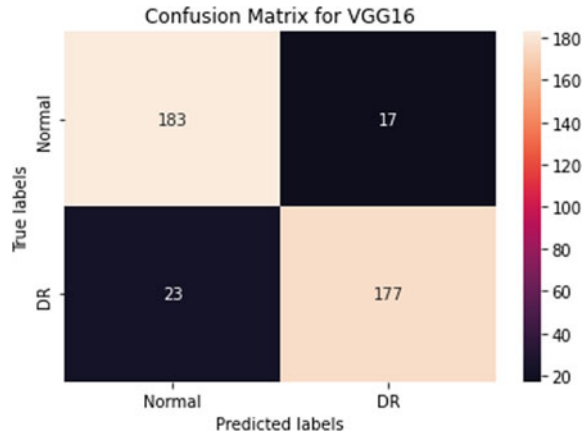


Fig. 15 Confusion matrix for VGG16 binary classification model



4.2 Multiclass Classification

The testing set in our dataset consisted of 600 images for multiclass classification with an equal number of NPDR, PDR, and normal fundus eye images.

Out of the 200 fundus eye images in each class, the ResNet-50 algorithm could correctly detect 180 normal eye images, 139 NPDR images, and 127 PDR images. The accuracy obtained for multiclass classification using ResNet-50 was 74.33%. Similarly, the accuracy obtained using EfficientNetB0 was 77.5%, and the accuracy obtained using VGG16 was 81.3%. Confusion matrix of multiclassification model is shown in Figs. 16, 17, and 18.

Fig. 16 Confusion matrix for ResNet-50 multiclass classification model

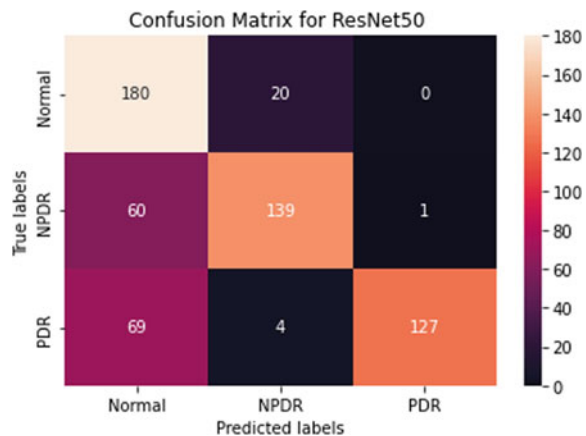


Fig. 17 Confusion matrix for EfficientnetB0 multiclass classification model

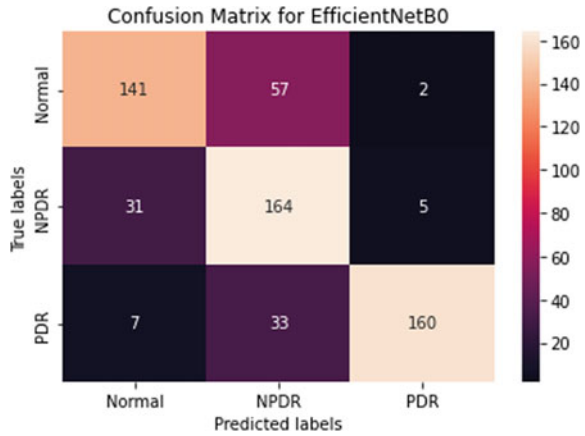
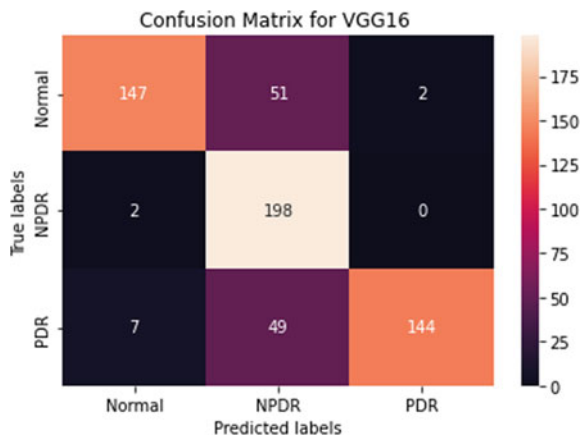


Fig. 18 Confusion matrix for VGG16 multiclass classification model



Testing accuracy of the transfer learning models for binary and multiclass classification is shown in Fig. 19. As discussed in the ‘Related work’ section, most of the researchers have directly used color images for training the model. But we observed that training the model without preprocessing the images did not give satisfactory results. Hence, we firstly preprocessed the images and then trained the model. Also we have utilized various machine learning and transfer learning techniques and created a comparative study of these techniques.

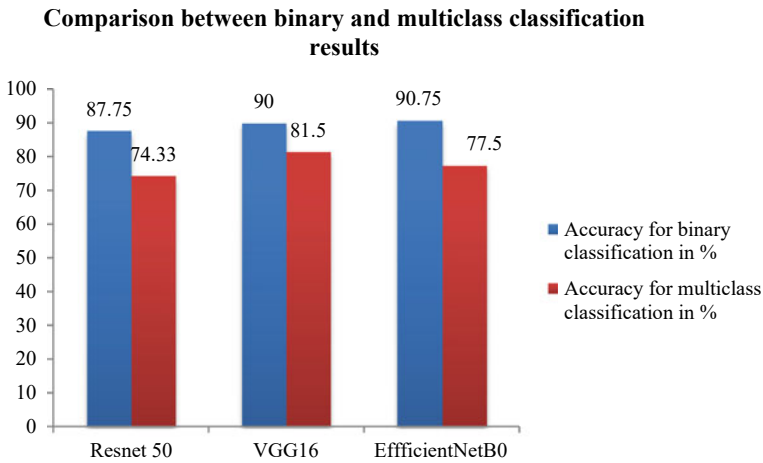


Fig. 19 Testing accuracy of the transfer learning models for binary and multiclass classification

5 Conclusion

In this study of classification of fundus images for diabetic retinopathy, various machine learning models like decision trees, random forest, Naive Bayes, K-nearest neighbors, and support vector machine were employed for binary classification and among all these models. The support vector machine classifier was found to give the highest testing accuracy. The accuracy of this model was 67.5%, which was poor as compared to the expected accuracy. So, to improve the model performance, transfer learning techniques such as ResNet-50, VGG16, and EfficientNetB0 were employed for performing binary classification. VGG16 and EfficientNetB0 were found to give high testing accuracy of 90%, for binary classification. In case of multiclass classification, the testing accuracy was highest for VGG16 at 81.5%. Thus, for both the types of classifications, VGG16 was found to be giving better results than the other transfer learning techniques. It was also observed that the testing accuracy could be further improved if the number of images available for training the model was increased and more fundus images were available for proliferative diabetic retinopathy condition.

This research work can be employed to develop an application which can classify the image as normal, suffering from DR or PDR. This could be achieved by providing real-time fundus images to the application. As a future scope, this work could be extended for applying machine learning algorithm for classifying real-time images for dynamic diagnosis using a web application.

References

1. Kumar A, Vashist P (2020) Indian community eye care in 2020: achievements and challenges. *Indian J Ophthalmol*
2. Majumder S, Kehtarnavaz N (2021) Multitasking deep learning model for detection of five stages of diabetic retinopathy. *IEEE Access* 9
3. Bhatkar AP, Kharat GU (2015) Detection of diabetic retinopathy in retinal images using MLP classifier. In: *IEEE international symposium on nanoelectronic and information systems*
4. Gupta S, Karandikar AM (2015) Diagnosis of diabetic retinopathy using machine learning. *J Res Dev*
5. Kanungo YS, Srinivasan B, Choudhary S (2017) Detecting diabetic retinopathy using deep learning. In: *2nd IEEE international conference on recent trends in electronics information & communication technology (RTEICT)*
6. Palavalasa KK, Sambaturu B (2018) Automatic diabetic retinopathy detection using digital image processing. In: *International conference on communication and signal processing (2018)*
7. Prasannan V, Sathish Kumar C, Deepa V (2018) An automated approach for diagnosing diabetic retinopathy in retinal fundus images. In: *3rd IEEE international conference on recent trends in electronics, information & communication technology*
8. Chetoui M, Akhloufi MA, Kardouchi M (2018) Diabetic retinopathy detection using machine learning and texture features. In: *IEEE Canadian conference on electrical & computer engineering*
9. Kaur P, Chatterjee S, Singh D (2019) Neural network technique for diabetic retinopathy detection. *Int J Eng Adv Technol*
10. Chaudhary S, Ramya HR (2020) Detection of diabetic retinopathy using machine learning algorithm. In: *2020 IEEE international conference for innovation in technology*
11. Robiul Islam M, Al Mehedi Hasan M, Sayeed A (2020) Transfer learning based diabetic retinopathy detection with a novel preprocessed layer. In: *IEEE region 10 symposium*
12. Revathy R, Nithya BS, Reshma JJ, Ragendhu SS, Sumithra MD (2020) Diabetic retinopathy detection using machine learning. *Int J Eng Res Technol* 9(06)
13. Ramchandre S, Patil B, Pharande S, Javali K, Pande H (2020) A deep learning approach for diabetic retinopathy detection using transfer learning. In: *IEEE international conference for innovation in technology*
14. Ayala A, Ortiz Figueroa T, Fernandes B, Cruz F (2021) Diabetic retinopathy improved detection using deep learning. *Appl Sci*
15. Rajkumar RS, Ragul D, Jagathishkumar T, Grace Selvarani A (2021) Transfer learning approach for diabetic retinopathy detection using residual network. In: *Proceedings of the sixth international conference on inventive computation technologies*
16. Khan Z, Khan FG, Khan A, Rehmani ZU, Shah S, Qummar S, Ali F, Pack S (2021) Diabetic retinopathy detection using VGG-NIN a deep learning architecture. <https://doi.org/10.1109/ACCESS.2021.3074422>
17. Ebin PM, Ranjana P (2020) An approach using transfer learning to disclose diabetic retinopathy in early stage. In: *International conference on futuristic technologies in control systems & renewable energy*