Automated Classification of Diabetic Retinopathy Using Deep Learning Architecture

Noel D'Souza, P. C. Siddalingaswamy, and N. V. Subba Reddy

Abstract Diabetic Retinopathy (DR) is an eye disease that can lead to blindness. It is one of the leading causes of vision loss in the world. Early detection of DR is critical to prevent permanent vision loss. DR is diagnosed by a trained ophthalmologist. In many developing countries, access to ophthalmologists for regular screening is often difficult. Using Machine Learning to overcome the need for skilled personnel to diagnose DR is a viable alternative. To automate the diagnosis of DR, we preprocessed our images to obtain 6 channels before training a modified Inception-v3 model on retinal fundus images obtained from Kaggle. We then trained a second model, combining data from both eyes, obtained as outputs from Inception-v3 to diagnose the stage of DR. Using this architecture, we were able to obtain a Kappa score of 0.821 on the dataset provided by Kaggle.

Keywords Diabetic retinopathy · Color model · Deep learning · Inception-v3 · Fundus image

1 Introduction

Diabetic Retinopathy (DR) leads to vision loss in certain diabetic patients. A long history of high blood sugar can lead to DR as it is one of the leading causes of blindness in developed countries and it is fast increasing in developing countries. Globally, DR cases are expected to increase from 126 million in the year 2010 to 191

N. D'Souza
Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA
e-mail: ndsouza@andrew.cmu.edu

P. C. Siddalingaswamy (✉) · N. V. S. Reddy
Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India
e-mail: pcs.swamy@manipal.edu

N. V. S. Reddy
e-mail: nvs.reddy@manipal.edu

S. Borah et al. (eds.), *Advanced Computational and Communication Paradigms*,
Lecture Notes in Networks and Systems 535,
https://doi.org/10.1007/978-981-99-4284-8_26

million by 2030 [1]. There are currently 463 million cases of diabetes worldwide, with India and China alone accounting for 77 and 116 million cases, respectively. Among the top 10 countries with diabetes worldwide, 8 are developing nations. The overall prevalence of DR among diabetic patients was estimated to be 35%. [2]. DR may go undiagnosed in developing countries where access to trained healthcare professionals may be inadequate. In the US, at the time of diagnosis of type-2 diabetes, 20–40% of people already have DR [3].

DR happens due to prolonged high blood sugar levels, blood vessels in the retina can swell and leak, and they can also become blocked; sometimes, this can lead to new abnormal growth of blood vessels on the retina leading to vision loss. Diabetic Retinopathy can be divided into 5 stages according to the International Clinical Diabetic Retinopathy Disease Severity Scale [4]. The first stage is no Diabetic Retinopathy. The next four stages can be divided into two classes, Non-Proliferate Diabetic Retinopathy (NPDR) and Proliferate Diabetic Retinopathy (PDR). The former can further be divided into 3 sub-stages namely mild, moderate, and severe NPDR. Proliferate Diabetic Retinopathy is the advanced stage of DR; it is caused when the retina starts growing new blood vessels. It is categorized by Neovascularization and preretinal hemorrhage. PDR can lead to vision loss. The chances of developing PDR from NPDR increase as the disease progresses; with mild NPDR, the chances of getting PDR within a year are 5%; this increases to 53% with severe NPDR [3].

Diabetic Retinopathy is diagnosed primarily by retinal fundus imaging. Here, an ophthalmologist usually dilates the pupils with drops before imaging. Another technique called fluorescein angiography can reveal changes in the structure of blood vessels on the retina; a special dye is used to highlight these retinal blood vessels. This technique however is a costly and time-consuming procedure. In many developing countries, not everyone has access to trained ophthalmologists; sometimes even when they do, the wait times can be long. An automatic DR grading system could provide a quick diagnosis and course of treatment to a large number of people. It would also provide economically challenged people access to cheap medical care.

A lot of research has been done in the automatic diagnosis of DR. Traditional methods work by employing several feature extractors to identify parts like blood vessels, micro aneurysms, etc. These feature extractors need to be hand-crafted to identify the underlying feature. Deep Learning using Neural Networks has made a lot of progress this decade and has broken several state-of-the-art benchmarks in various fields. They have achieved or exceeded human-level skills in tasks ranging from image recognition and object detection to video games such as Atari. They have even performed remarkably well in NLP tasks with language models GPT-2 generating extremely convincing human-like text [5, 6]. With the power of Deep Learning, a lot of progress has been made in automatically classifying DR images.

In the proposed work we train a deep convolutional neural network to grade DR images from Kaggle [7] into one of five classes. The proposed algorithm is evaluated on the test set provided by the competition, achieving results in the top 10 of the competition leaderboard; in addition, due to our preprocessing technique, we

can make it work with a moderately sized neural network. We achieve much better results compared to recent works in the field.

2 Related Work

Various techniques have been developed to automatically diagnose Diabetic Retinopathy. The earlier techniques used manually designed feature extractors to identify the underlying features. The detected features are then inputted to classifiers like K-Means, Random Forest, etc. to group and classify DR. In recent years, deep learning techniques have been deployed to identify and classify DR. These employ Neural Networks that learn the underlying features. There also exist different datasets that have been used to train DR algorithms with a large variation in the number of images in these datasets, ranging from a few hundred to tens of thousands. Here, we summarize some recent work in automatically detecting DR as well as highlight some traditional techniques.

Traditional techniques rely on domain knowledge and handcrafted feature selection. Designing these feature extractors is often time-consuming. Agurto et al. [8] used a technique that defined and characterized features based on their pixel intensity, size, and geometry. A dataset comprising images from 822 patients was used. An area under the curve of 0.89 was obtained for the detection of retinopathy. M. Usman Akram et al. [9] detect microaneurysms based on their shape, color, etc. These features are extracted using various filters and morphological transforms and are then classified using a hybrid classifier. They achieved 99% sensitivity and 99% specificity on the DIARETDB0 and DIARETDB1 datasets, comprising a total of 219 images. Harini et al. [10] extracted blood vessels, exudates, microaneurysms, etc. using an FCM classifier and morphological transforms, and classified them based on their area using an SVM. They too used the DIARETDB0 and DIARETDB1 datasets and achieved a sensitivity of 100%, 95.83% specificity, and accuracy of 96.67%.

Several recent works using deep convolutional neural networks have achieved remarkable results in identifying DR. Gulshan et al. [11] used a private dataset consisting of more than 128 k images, to train an Inception-v3 [12] model to classify DR images. The algorithm was evaluated at two operating levels, one selected for high sensitivity and another for high specificity.

Gao et al. [13] used a novel dataset consisting of 4476 images labeled into 4 classes. They divide each image into four parts and use a different Inception-v3 network for each. The output features are merged into a single vector and fed to a softmax layer for classification. They achieved accuracy, precision, and recall of 88.72%, 95.77%, and 94.84%, respectively. Graham [14] was the winner of the Kaggle Diabetic Retinopathy competition with a Kappa score 0f 0.849, making use of fractional max pooling and an ensemble of 3 models. The 5-class softmax output for an eye was combined with that of the other and fed to a Random Forest. Arcadu et al. [15] used a deep learning approach to predict future DR progression over the course of 6, 12, and 24 months based on fundus images acquired during a single visit.

They achieved an AUC of 0.79 for a prognosis of 12 months. Yang et al. [16] used the Kaggle dataset along with a two-stage CNN to classify severity. Each image was divided into patches and graded by a network; these gradings were used to generate weighted lesion maps that were combined with the input image and fed to another network for final class DR prediction. They achieved an AUC of 0.959. Ni et al. [17] used an Inception-v3 model trained on the dataset from Kaggle to classify retinal fundus images into 2 classes and 5 classes. They used selective data sampling to overcome data imbalance among the classes in the dataset. On the 2-classs and 5-class problems, they achieved accuracies of 92% and 87.2% respectively, and Kappa scores of 0.744 and 0.806, respectively. Wan et al. [18] used the Kaggle dataset for training. They trained and compared several model architectures, VggNet-s, AlexNet, VggNet-16, GoogleNet, VggNet-19, and ResNet. Transfer learning and hyperparameter tuning were used to overcome overfitting. They used five-fold cross validation to compare the accuracy of the different architectures. The highest accuracy obtained was 95.68% with VggNet.

3 Methodology

3.1 Dataset

The dataset was obtained from Kaggle; it consists of images from EyePacs in collaboration with the California Healthcare Foundation. The images were captured under various lighting conditions and cameras. In addition to this, some images are overexposed, some are underexposed, while others may be out of focus or contain artifacts. Thus, an algorithm that is robust to such variance in the input is needed. Each pair of images has a unique ID along with a label specifying either the left or right eye. The images are graded into 5 categories. Furthermore, the number of samples present in each class is extremely unbalanced, with class 0 containing roughly 73.5% of the 35,126 images. In addition to these images, another dataset consisting of 53,576 was obtained from Kaggle. Our model was evaluated on this dataset; the details are given in Table 1. The retinal dataset was divided into a training set and a validation set. Since the number of images per class was extremely unbalanced, we chose to have a validation set that represented balanced classes; to achieve this, we sampled each class randomly and extracted 150 images from each. This gave us a validation set with a total of 750 images. These images were preprocessed according to techniques we describe later and saved. Figure 1 illustrates the sample images from the dataset.

Table 1 DR grades and their sample counts for train and test sets

Class name	Train set	Test set
0—No DR	25,810	39,533
1—Mild	2,443	3,762
2—Moderate	5,292	7,861
3—Severe	873	1,214
4—Proliferative DR	708	1,206

3.2 Data Preprocessing and Augmentation

Due to high variability in image size, composition, and contrast brought about by different lighting conditions and cameras used, the image needs to be transformed to a uniform format. This way, camera-specific artifacts and lighting conditions can be removed, allowing the to model to generalize to underlying image information rather than camera-specific data. Figure 2 illustrates the various images after the following preprocessing steps:

(1) *Resize to specific radius*: The image is first resized such that the retina is a fixed radius; we experimented with various values, finally settling on 250 pixels.
(2) *Subtract mean local color*: The mean local color was subtracted from the image to correct for any differences in lighting, creating uniform standard images.
(3) *Clip Radius*: The outer 10% of the radius is clipped to remove any external edge effects.
(4) *CLAHE*: It is a technique used to improve contrast in an image. It calculates different histograms over different sections of the image, improving local contrast. It also clips contrast amplification above a certain threshold.
(5) *HSV and LAB transforms*: This image is then converted to HSV (hue, saturation, value) and LAB (CIELAB) formats; the V channel from HSV is extracted; it contains data in the range (0, 1); it is multiplied by 255 to bring it to the standard (0, 255) range. The AB channels from LAB are also extracted; these channels contain values in the range (−128, 127). They are converted to the standard (0, 255) range by adding 128, giving a new VAB image.
(6) *Resizing*: The newly created RGB and VAB images are both cropped to leave behind only the retina; this new image is then padded with a value of 128 to make it square. And finally resized down to 300 × 300.

3.3 Model

Neural Networks are a set of supervised learning algorithms that attempt to map information from an input space to an output space. They are loosely modeled after biological neurons in the brain. They perform tasks without being explicitly programmed to do so. Every ANN is made up of individual neurons, with input, output, and optional

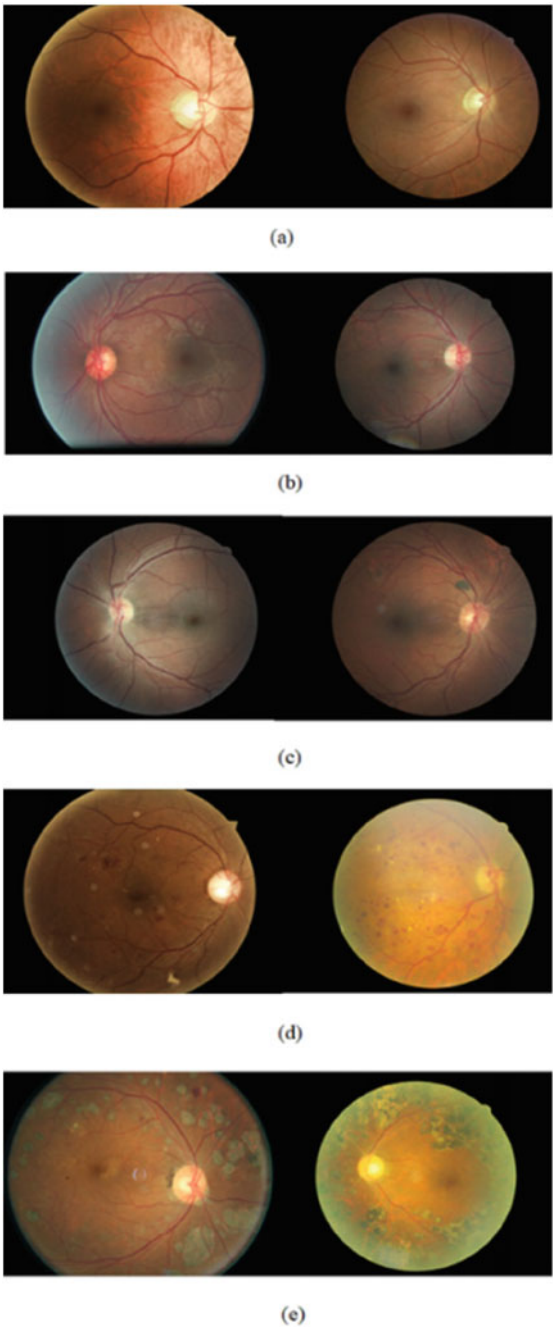

Fig. 1 Image samples from each class: **a** Normal, **b** Mild, **c** Moderate, **d** Severe, and **e** affected

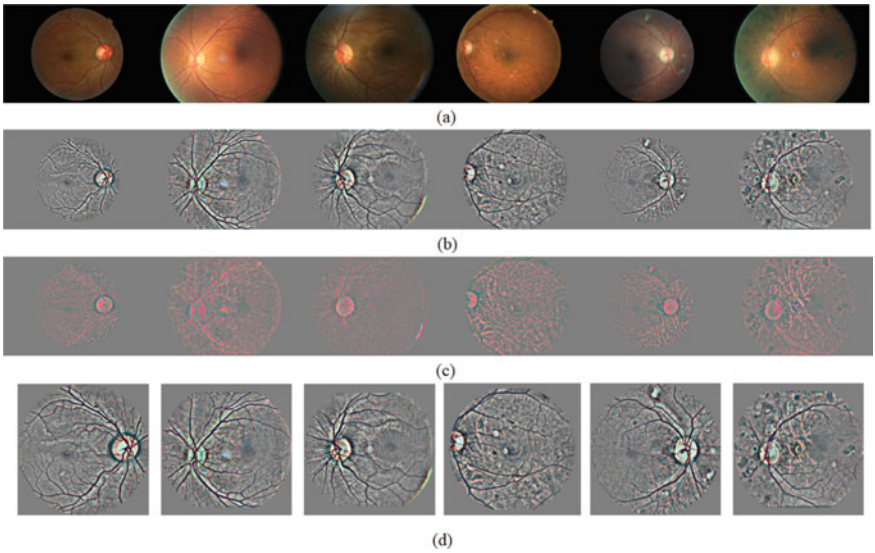

Fig. 2 The various images after preprocessing steps: **a** Original Image, **b** The image after Radius Normalization, color Normalization, and CLAHE, **c** The image after converting to VAB channels, and **d** The image after cropping, padding, and resizing

intermediate hidden layers. Each neuron in one layer is connected to every neuron in the next layer. The input consists of real numbered data that passes through the network, and is multiplied by the neuron's weights; the output is then compared to the true output with the help of a loss function; the weights of the model are then adjusted based on the output of this loss function. The Back-Propagation algorithm is used to calculate how much each weight should be changed based on the output; it does so by calculating the gradient of the loss function with regard to individual network weights. Convolutional Neural Networks are a type of neural network typically employed for visual data. They generally consist of convolution operations, activation functions, and pooling layers.

The convolution layers work by computing the convolution with a fixed-sized filter over the input. The weights of this filter are learnt by the model. The outputs from a filter are passed to an activation function before moving on to further layers. Rectified Linear Unit (RELU) activation function is typically employed in CNNs. Pooling operations down-sample the input reducing the number of computations; they also make the network robust to translational variance in the image. Max pooling is a type of pooling operation that is commonly used in CNNs; it works by sliding a window over the input at selecting the maximum value within that window. Using a combination of these three properties, convolutional networks of arbitrary shapes and sizes can be built. Figure 3 illustrates the CNN architecture of the overall prediction pipeline, from raw image input to output class prediction. Tremendous breakthroughs have been made using deeper and deeper convolutional networks. Starting with LeNet

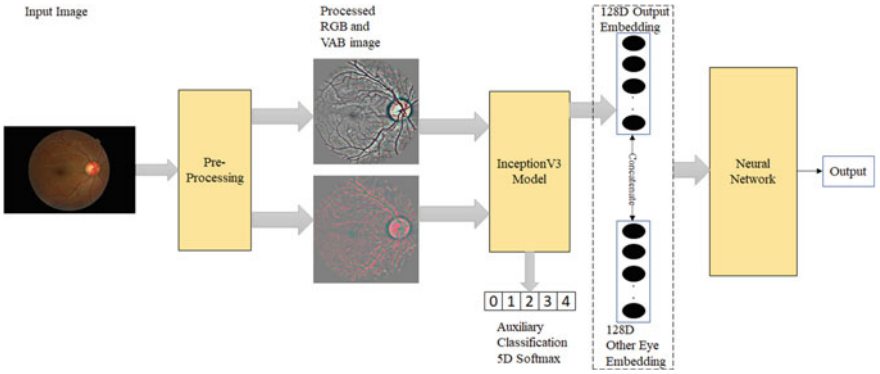

Fig. 3 The overall prediction pipeline, from raw image input to output class prediction

in 1998, this conv net could recognize handwritten digits [19]. However, the networks of the time were difficult to scale and required large amounts of data to work well. Import breakthroughs came in the last decade, starting with AlexNet [20] in 2012, which broke the then benchmark on ImageNet [21], owing its success to being deeper and using GPUs for quick training. VGGNet [22], ResNet [23], and Inception Net [12] are some of the networks that improved on the ImageNet benchmark, with each of these networks having several different iterations based on parameters like neural structure.

Inception Net is a Deep Learning image classifier built by Google. It was built to classify images on the ImageNet [21] classification task where it achieved a new benchmark in terms of accuracy. Since images can have a large variation on where certain objects of the same type are located, choosing appropriate filter sizes becomes difficult, Inception Net solves this by using filters of different sizes at the same level. The network has several such predefined blocks consisting of different filter sizes that are stacked on top of each other. Inception-v3 [12] was the third iteration in the Inception Net architecture; it had several changes in filter sizes to improve both training times and accuracy over previous versions. We used this version of Inception Net to build our model.

We used the inception-v3 model with weights that were pre-trained on ImageNet [21]. Since our input consisted of 6 channels rather than 3 channels that the Inception Net was designed for originally, a few modifications to its structure were required. The new model had two parallel input layers rather than one earlier. The input to these layers were our two $300 \times 300 \times 3$ images. The output from the input layers was concatenated to give an image with 6 channels. Since the original network had only 3 channels, the pre-trained weights from the first convolution layer were appended together to match the dimensions of the new input shape. Thus, it was not necessary to initialize the weights of this layer from scratch.

The last convolution layer of the network was followed by a Global Average Pooling Layer and a dropout of 30% to prevent overfitting. We experimented with various fully connected layer configurations, finally settling on a single 128D fully

Table 2 Network structure of second neural network

Layer name	Output shape	Layer name
Input layer	256	Input layer
Layer 2	256	Layer 2
Layer 3	256	Layer 3
Layer 4	128	Layer 4
Layer 5	128	Layer 5

connected layer, with RELU activation. The output layer has 5 neurons with a softmax activation function. The weights for the last two layers were initialized randomly using the Xavier initializer. The categorical cross-entropy function was used to calculate the loss.

During the evaluation, the final layer of the model was removed and the 128-dimensional output from the second last layer was saved. This output embedding was calculated for all fundus images. The second neural network was trained to predict the class label. The input to this network was the embedding from a pair of eyes, thus requiring a 256-dimensional vector as input. The neural network structure is described in Table 2.

3.4 Training the Model

We trained the main model using the Adam optimizer [24]. Since the samples per class were extremely unbalanced, class weights were used to prevent overfitting to any one class. These class weights were set according to the following formula as given in (1).

Let n_i denote the number of samples in class i, $\forall i \in [0, 4]$ $[0, 4]$.

Then,

$$t = 1 + n_0\left(\frac{1}{n_1} + \frac{1}{n_2} + \frac{1}{n_3} + \frac{1}{n_4}\right) \tag{1}$$

If w_i denotes the weight of class i, $\forall i \in [0, 4]$ then,

$$w_0 = \frac{1}{t}$$

$$w_1 = \frac{n_0/n_1}{t}, w_2 = \frac{n_0/n_2}{t}, w_3 = \frac{n_0/n_3}{t}, w_4 = \frac{n_0/n_4}{t}$$

Thus, for the dataset of 35,126 images, the class weights calculated are given in Table 3.

Table 3 Class weights used during training

Class	Weight
0	0.01212686
1	0.12811882
2	0.05914480
3	0.35852722
4	0.44208230

Initially, only the first convolution layer and the last two fully connected layers were trained, with the rest of the model frozen for training. Following this, the entire model was trained, starting with a learning rate of 6e-4 and ending with a learning rate of 2e-5. The learning rate was manually decreased in steps at certain epochs when the validation loss started to stagnate. The learning rate along with accuracy versus epochs is plotted below.

The weight of class 0 was then increased to 0.025 and the model was further trained; the accuracy on the training set greatly increased, however, the validation accuracy was observed to marginally drop, suggesting that the model was not overfitting on the training data or any particular class. The change in weight is denoted by the red line at epoch 46. Figure 4 illustrates accuracy versus number of epochs on training and validation sets.

The 128-D output embedding from the Inception Net was combined with that from the other eye. This was then passed to our second neural network. We trained the second network as a regressor for the 5-class problem, and final class labels were arrived at after thresholding. For the 2-class problem, we trained the second model as

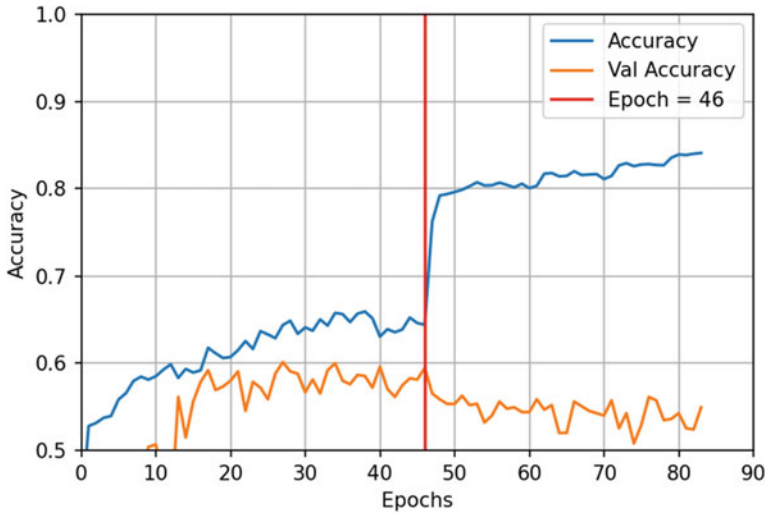

Fig. 4 Accuracy versus number of epochs on training and validation sets

a binary classifier, even though thresholding the regressor network into two classes gave similar results. This was done in order to get the probability scores for the ROC curve.

4 Results and Discussion

We focused on the 5-class and 2-class problems. In the latter, we clubbed classes 0, 1, and classes 2, 3, and 4 as negative and positive, respectively. With our second network as a regressor and binary classifier for the 5-class problem and 2-class problems respectively, predictions for both problems were obtained. We used a technique called Test Time Augmentation (TTA) while evaluating the Inception model on the training and testing sets. TTA works by evaluating the same image several times by randomly augmenting it before each run through the model.

Let m_i be the ith image in the dataset, and let $f(x)$ be a function that returns an augmented image $x\prime$; K number of calls to this function return K images together denoted by $m_{i,k}\prime$; each of these images is fed to our classifier to get an output 128-D embedding vector denoted by $p_{i,k}$. The output class probabilities for each set of augmented images are then averaged as shown below where P_i represents the new probability vector for image m_i as given in (2):

$$P_i = \frac{1}{K} \sum_{k=1}^{K} p_{i,k} \tag{2}$$

Random augmentations were performed on the image prior to prediction, Width/Height shift in a (0, 10)% range and Rotation between (0,180) degrees. The number of samples K generated was set to 7. The above values were arrived at by trying various permutations and combinations of parameters. Using this technique gave us a boost in overall accuracy.

The metrics we use to evaluate our model are as follows:

(1) Accuracy
(2) Quadratic Weighted Cohen Kappa for the 5 classes
(3) Sensitivity and Specificity
(4) ROC curve.

The outputs from the 5-class problem were thresholded to give the final labels. The threshold values we used were 0.47, 1.25, 2.04, and 3.29. Based on this, we calculated the accuracy (Table 4).

Sensitivity and Specificity were obtained on the 2-class problem as 0.752 and 0.976, respectively. The ROC curve for the 2-class problem is shown in Fig. 5.

Table 4 Result metric score

	Accuracy	Kappa
5-class	0.80	0.821
2-class	0.933	0.771

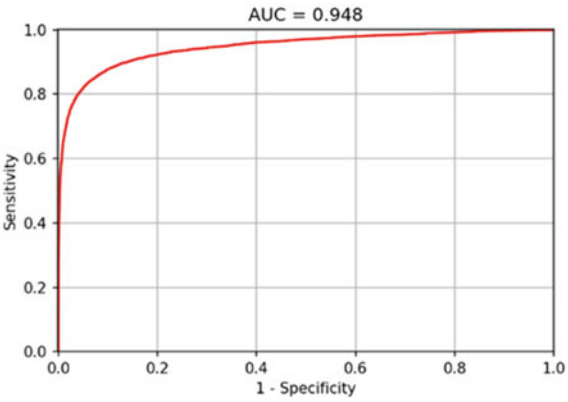

Fig. 5 ROC curve for testing set with an AUC of 0.948

5 Conclusion

In this paper, an Inception-v3 model was trained on retinal fundus images obtained from Kaggle. We focused on the 2-class and 5-class problems where we obtained accuracies of 0.933% and 0.80%, and Kappa scores of 0.821 and 0.771, respectively. By using class weights, we avoid the model overfitting on any one class. The results of the method described are comparatively better than the existing methods, thus proving the efficacy of adopting the proposed method in a clinical setting.

References

1. Zheng Y, He M, Congdon N (2012) The worldwide epidemic of diabetic retinopathy. Indian J Ophthalmol 60(5):428
2. Williams R, Colagiuri S, Chan J, Gregg E, Ke C, Lim LL, Yang X (2019) IDF Atlas 9th
3. American Optometric Association (2014) Evidence-based clinical practice guideline: eye care of the patient with diabetes mellitus. St. Louis (MO), American Optometric Association
4. American Academy of Ophthalmology (2002) International clinical diabetic retinopathy disease severity scale, detailed table
5. Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M (2013) Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.
6. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners. OpenAI blog 1(8):9
7. 'Diabetic Retinopathy Detection' (2015) Kaggle. https://kaggle.com/competitions/diabetic-retinopathy-detection

8. Agurto C, Barriga ES, Murray V, Nemeth S, Crammer R, Bauman W, Zamora G, Pattichis MS, Soliz P (2011) Automatic detection of diabetic retinopathy and age-related macular degeneration in digital fundus images. Invest Ophthalmol Vis Sci 52(8):5862–5871
9. Akram MU, Khalid S, Khan SA (2013) Identification and classification of microaneurysms for early detection of diabetic retinopathy. Pattern Recogn 46(1):107–116
10. Harini R, Sheela N (2016) Feature extraction and classification of retinal images for automated detection of diabetic retinopathy. In: Second International conference on cognitive computing and information processing (CCIP). IEEE, pp 1–4
11. Gulshan V, Peng L, Coram M, Stumpe MC, Wu D, Narayanaswamy A, Venugopalan S, Widner K, Madams T, Cuadros J, Kim R (2016) Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA 316(22):2402–2410
12. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 2818–2826
13. Gao Z, Li J, Guo J, Chen Y, Yi Z, Zhong J (2018) Diagnosis of diabetic retinopathy using deep neural networks. IEEE Access 7:3360–3370
14. Graham B Kaggle. [Online] https://www.kaggle.com/c/diabetic-retinopathy-detection/discussion/15801. Accessed 6 Sept 2020
15. Arcadu F, Benmansour F, Maunz A, Willis J, Haskova Z, Prunotto M (2019) Deep learning algorithm predicts diabetic retinopathy progression in individual patients. NPJ Digit Med 2(1):1–9
16. Yang Y, Li T, Li W, Wu H, Fan W, Zhang W (2017) September. Lesion detection and grading of diabetic retinopathy via two-stages deep convolutional neural networks. In: International conference on medical image computing and computer-assisted intervention. Springer, Cham, pp 533–540
17. Ni J, Chen Q, Liu C, Wang H, Cao Y, Liu B (2019) An effective CNN approach for diabetic retinopathy stage classification with dual inputs and selective data sampling. In: 18th IEEE international conference on machine learning and applications (ICMLA). IEEE, pp 1578–1584
18. Wan S, Liang Y, Zhang Y (2018) Deep convolutional neural networks for diabetic retinopathy detection by image classification. Comput Electr Eng 72:274–282
19. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551
20. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
21. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. IEEE Conf Comput Vis Pattern Recognit
22. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
23. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 770–778
24. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. ICLR. arXiv preprint arXiv:1412.6980, 9
25. Nguyen QH, Muthuraman R, Singh L, Sen G, Tran AC, Nguyen BP, Chua M (2020) Diabetic retinopathy detection using deep learning. In: Proceedings of the 4th international conference on machine learning and soft computing. pp 103–107