Diabetic Retinopathy Detection Using Convolutional Neural Networks

[1]Sherin Thomas, [2]Reshmi S, [3]Nandana V, [4]Navaneeth Babu, [5]Neeraj Nandan

[1]Under Graduate, [2]Assistant Professor, [3,4,5]Under Graduate
[1,2,3,4,5]Department of Electronics and Communication,
[1,2,3,4,5]NSS College of Engineering, Palakkad, Kerala, India

***Abstract*— Diabetes is one of the most common health issues that we come across in our lives. Most of the people are unaware of the consequences of not taking care of the same. Diabetes leads to a lot of health dilemmas, one of which being Diabetic Retinopathy. Diabetes leads to rise in blood sugar level which in turn becomes the source for the development of Diabetic Retinopathy and having too much sugar can damage the retina over time along with damage of optical nerve in the long run. Diabetic Retinopathy can even lead to blindness if not treated properly in the initial stages. Convolutional Neural Networks can be used efficiently for the detection of Diabetic Retinopathy. The fundus images of retina can be tested using a well-trained network and the disease as well as its current stage can be determined. Some of the existing networks namely- VGG16, Resnet50 and Inception V3 have been trained with the datasets from kaggle. The most efficient network being Inception V3 is selected and has been trained and tested with the external dataset collected from Government Medical College, Thrissur. Hyperparameter tuning has also been studied with the parameters learning rate, epoch, decay and momentum, giving a wider picture of how effective CNN can be in diagnosing Diabetic Retinopathy.**

***Index Terms*— Classification, CNN, Diabetic retinopathy, Fundus images.**

I. INTRODUCTION

Diabetes is one of the world's most serious health problems. Diabetic retinopathy (DR) is a subset of diabetes that leads to blindness. It can be divided into various types based on the intensity and can be properly treated if diagnosed early enough. DR infects the retinal region of the eye, which oversees converting light into an electric signal that can be explicated to form a picture of an object. The retina has a fretwork of blood vessels that supply it with the vital nourishment required. Diabetes affects this grid of blood vessels, resulting in a lack of blood supply to the retina. This can affect the retina's health and, as a result, distorts a person's vision. Background retinopathy is the first stage of diabetic retinopathy. Diabetes does not affect vision at this stage, but it does affect blood vessels.

Fig. 1. DR Fundus Image

As shown in Fig. 1, the vessels can bulge slightly(microaneurysms), leak fluid and proteinsexudates), or leak blood (retinal hemorrhages). At a later stage, DR develops into proliferative retinopathy, which causes more damage to the retina than background retinopathy. As the retina is denied proper blood supply, it affects its nutrient cycle as a result it increases the risk of vision loss and poses a threat of severe myopia or blindness.

Manual detection of DR by ophthalmologists or expert graders is costly. Repetitive examining of a bunch of diabetic patients in order to inspect for the presence of DR places a burden on the experts, reducing their productivity and impeding the time for treatment. Different graders' findings differ. As a result, there is a prominent need for an intuiting self-regulating system for assessing patients for DR presence, which can assist graders in exposing DR with conviction and referring the person to an expert for additional diagnosis and medication.

In the last decade, many computer-based systems for DR diagnosis have been developed. Deep learning (DL) is currently being used for DR diagnosis from fundus images, with promising results. The Convolutional Neural Network (CNN) is a common DL architecture that involves a huge number of masterable frameworks and requires a copious proportion of data for training, which

is primarily inaccessible in the case of DR, resulting in the phenomenon of overfitting. For DR grading, CNN models that have been fine-tuned and pre- trained are used. Legitimate images have different composition than fundus images created by the eye's retina, so pre-training and tuning of these models are necessary for them to adapt to fundus image structures.

For fine-tuning pre-trained CNN models, most transfer learning techniques use fundus images. In this method, a retinal fundus image is used as input, which is then processed with an adeptly calibrated CNN model and graded into discrete clinically predesigned DR levels.

II. METHODOLOGY

This project is implemented in five phases. The first phase is about analyzing different CNN models available and selecting a few that show prominence with respect to the problem in hand. Second phase includes training the CNN models and finalizing the model to be used based on the results obtained. Third phase is about collecting legitimate dataset from respective hospitals and labeling them. Fourth phase focuses on testing the finalized model and obtaining the accuracy. Fifth phase is hyperparameter tuning.

Analyzing CNN models

This step includes gazing upon the works of different CNN models through research papers and scientific documents and obtaining a precise idea about their performance and properties. And based on these properties selecting few appropriate models among the list that can perform exceptionally when it comes to the case of diabetic retinopathy detection.

Training of CNN models

Images comprising of the different stages of diabetic retinopathy are collected from the site named kaggle in bulk as training datasets and is used to train the selected CNN models. Here the models are tested to discover their accuracy and time of delivery. Based on the results obtained the pre- eminent model is chosen from the bunch and is finalized as the foremost model for performing the task in hand.

Dataset collection

Images consisting of real-life cases of diabetic retinopathy are collected from various hospitals. Further the images are examined by an experienced ophthalmologist and are specifically classified into different stages of diabetic retinopathy.

Testing of CNN model

The labeled images are assigned as testing images and used for the testing process of the trained CNN model. The accuracy of the model in determining the stages of the disease precisely is noted as results.

Hyperparameter Tuning

Parameters like learning rate, momentum, decay, and epochs are taken and they are given various values. The results are then recorded and analyzed on the which the efficiency is also determined for various values.

III. SELECTED NETWORKS

Based on the study of different network layers available, we have selected 3 networks which can give most accurate results. They are VGG16, Inception V3 and Resnet50.

Resnet50

Residual Network or Resnet was first introduced by Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun in 2015. Resnet can train up to hundreds or thousands of layers and can still achieve better performance. After AlexNet, other networks started to introduce more convolutional layers. But increasing the number of layers can lead to vanishing gradient problem. This vanishing gradient problem can affect the performance of the network. But the main advantage of Resnet is that it can eliminate the vanishing gradient problem. More variations of this network have introduced lately. Out of all these variants, we selected the widely used RESNET50 for our project. Resnet50 has 48 Convolutional layers. Along with that, it has a MaxPool layer and an Average Pool layer. The image input size of the network is 224 × 224.

VGG16

VGG or Visual Geometry Group is a standard deep Convolutional Neural Network (CNN) architecture with multiple layers. "Deep" refers to the number of layers with VGG-16 or VGG-19 consisting of 16 and 19 convolutional layers. As mentioned above, the VGGNet-16 supports 16 layers and can classify images into 1000 object categories. The model has an image input size of 224 x 224. The network is constructed with very small convolutional filters consisting of 13 convolutional layers and 3 fully connected layers.

Inception V3

Inception V3 is the 3rd consecutive version of a pre-trained Convolutional Neural Network model. It is 48 layers deep. The model has an image input size of 299 299. It is a version of the network already trained on using more than a million images from the ImageNet database. It was developed by google. The Inception V3 model was released in the year 2015, has a total of 48 layers and a lower error rate than its predecessors. The major modifications done on the Inception V3 model are:

1) Factorization into smaller convolutions
2) Spatial factorization into asymmetric convolutions
3) Utility of auxiliary classifiers
4) Efficient grid size reduction

1) *Factorized convolutions:* This decreases the number of parameters used in a network while keeping check on the network efficiency, which helps to lower the computational efficiency.

2) *Smaller convolutions:* Replacing bigger convolutions with smaller convolutions leads to faster training. Say a 5 × 5 filter has 25 parameters; replacing it with two 3 × 3 filters results in only 18 (3 × 3 + 3 × 3) parameters.
3) *Asymmetric convolutions:* A 1 × 3 convolution followed by a 3 × 1 convolution could be used in place of a 3 × 3 convolution. The number of parameters would be a little bit larger than the suggested asymmetric convolution if a 3 × 3 convolution were swapped out for a 2 × 2 convolution.
4) *Auxiliary classifier:* A small CNN called an auxiliary classifier is placed between layers during training, and the loss it incurs is added to the loss of the primary network. A deeper network in GoogleNet was built using auxiliary classifiers. An auxiliary classifier function as a regularizer in Inception V3.
5) *Grid size reduction:* Grid size reduction is done by pooling operations to tackle the bottleneck of computational costs.

All the above concepts are consolidated into the final architecture:

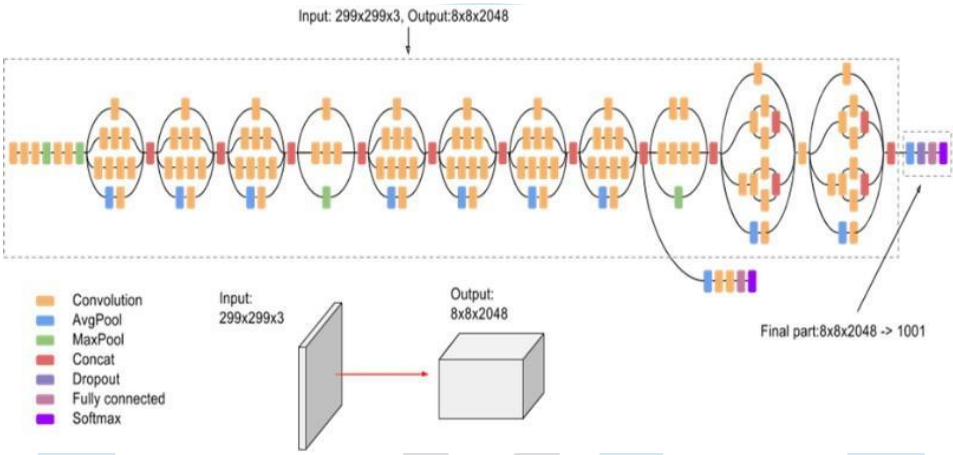

Fig. 2. Inception V3 Architecture

IV. DATASET

The dataset for training the model was collected from kaggle. Over 3,667 images for training were included in that dataset. The model was trained using them. The image data for testing the model was collected from Govt Medical College (GMC), Thrissur. 81 images, with various DR stages were present in it. The images were then labelled by Ophthalmologist.

V. TRAINING RESULT

VGG16, Resnet50 and Inception V3 have been trained with the datasets from kaggle and ideal model architecture is obtained using hyperparameter tuning.

Table I Training Summary

Selected Networks	Time Taken for Training 10 Epoch	Training Accuracy
Inception V3	2 hrs.	88.39%
Resnet50	15 mins	71%
VGG16	3hr 12 mins	48.41%

After training the three networks we concluded that Inception V3 has better efficiency (88.39%). Accuracy of 88.92% was achieved for training after hyperparameter tuning where Learning rate = 0.01, Decay= 0.01 and Momentum = 0.9. So, Inception V3 network was taken for testing input data set collected from Government Medical College, Thrissur.

Training result of Inception V3

Training of selected networks were done by Google Colab.
Number of epochs = 10
Batch size = 90

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:8: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future versio

Epoch 1/10
34/34 [==============================] - 631s 18s/step - loss: 0.8648 - accuracy: 0.6864 - val_loss: 1.4058 - val_accuracy: 0.6093 - lr: 0.0100
Epoch 2/10
34/34 [==============================] - 577s 17s/step - loss: 0.6037 - accuracy: 0.7665 - val_loss: 1.1346 - val_accuracy: 0.6852 - lr: 0.0100
Epoch 3/10
34/34 [==============================] - 571s 17s/step - loss: 0.5017 - accuracy: 0.8121 - val_loss: 0.6613 - val_accuracy: 0.7296 - lr: 0.0100
Epoch 4/10
34/34 [==============================] - 577s 17s/step - loss: 0.4295 - accuracy: 0.8389 - val_loss: 0.6572 - val_accuracy: 0.7556 - lr: 0.0100
Epoch 5/10
34/34 [==============================] - 574s 17s/step - loss: 0.4048 - accuracy: 0.8442 - val_loss: 0.6528 - val_accuracy: 0.7815 - lr: 0.0100
Epoch 6/10
34/34 [==============================] - 514s 15s/step - loss: 0.3763 - accuracy: 0.8674 - val_loss: 0.6808 - val_accuracy: 0.7648 - lr: 0.0100
Epoch 7/10
34/34 [==============================] - 568s 17s/step - loss: 0.3307 - accuracy: 0.8779 - val_loss: 0.5722 - val_accuracy: 0.7926 - lr: 0.0100
Epoch 8/10
34/34 [==============================] - 572s 17s/step - loss: 0.3178 - accuracy: 0.8743 - val_loss: 0.5867 - val_accuracy: 0.8019 - lr: 0.0100
Epoch 9/10
34/34 [==============================] - ETA: 0s - loss: 0.2874 - accuracy: 0.8882
Epoch 9: ReduceLROnPlateau reducing learning rate to 0.0009999999776482583.
34/34 [==============================] - 569s 17s/step - loss: 0.2874 - accuracy: 0.8882 - val_loss: 0.6623 - val_accuracy: 0.7722 - lr: 0.0100
Epoch 10/10
34/34 [==============================] - 570s 17s/step - loss: 0.2928 - accuracy: 0.8839 - val_loss: 0.5599 - val_accuracy: 0.7907 - lr: 1.0000e-03
4
```

Fig. 3. Inception V3 training result

Accuracy plot

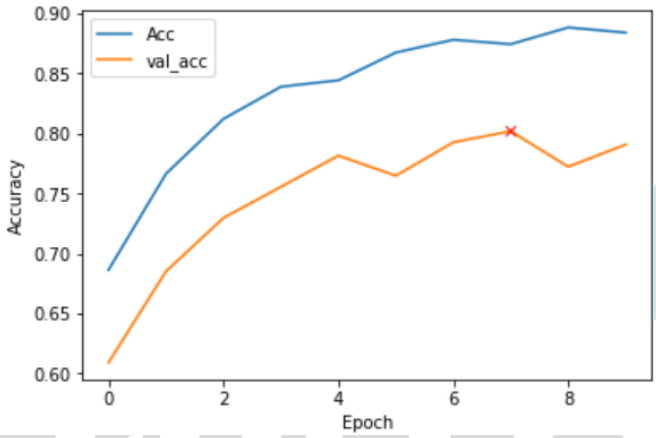

Fig. 4. Accuracy graph for Inception V3

VI. TESTING RESULT

The most efficient network, Inception V3, is selected and has been trained and tested with the external dataset collected from Government Medical College, Thrissur, and an accuracy of 64.19% is obtained.

```
[ ] def compute_accuracy(y_true, y_pred):
        correct_predictions = 0
        # iterate over each label and check
        for true, predicted in zip(y_true, y_pred):
            if true == predicted:
                correct_predictions += 1
        # compute the accuracy
        accuracy = correct_predictions/len(y_true)
        return accuracy

    y_true = [2,4,4,4,4,4,4,2,4,2,2,1,2,0,4,4,4,4,0,4,4,0,0,0,2,4,4,4,0,2,2,2,4,4,2,3,4,1,3,3,4,4,4,4,0,0,4,2,1,3,3,3,3,4,3,4,3,2,0,0,2,4,4,4,0,4,1,0,
    y_pred = [2,4,4,4,4,0,4,4,4,4,2,2,1,2,0,4,4,4,4,1,4,1,4,0,0,1,1,4,4,4,0,3,2,2,1,1,2,1,4,3,3,4,4,4,4,0,0,4,2,1,2,3,4,3,4,3,2,4,1,0,0,0,1,4,4,4,0,4,1,0,
    # get the accuracy
    compute_accuracy(y_true, y_pred)

    0.6419753086419753
```

Fig. 5. Testing Result

VII. CONCLUSION

It is important to acknowledge the fact that "what appears minimum, today grows out to be uncontrollable in the future." DR as a disease is quite misunderstood and ignored by the current medical community. It is one of the diseases that can be avoided in a person if they are properly tended to at an early stage. But it is important to concede the fact that an untimely approach towards the matter can lead to several consequences which may even include a scenario of permanent blindness.

Detection of a disease is as important as its diagnosis. So, a proper method that can duly detect the presence of any ailment in a person is a major factor that contributes towards that person's conventional health management. CNN as a detection system has shown major advances in the medical field. It has widely been accepted throughout the world for its accurate and optimized outcome production. Using CNN to detect or analyze the presence of DR in a person can widely improve the scope of recognizing the disease within the person at an early stage and also in assisting the experts to decide whether the person is in an immediate requirement of treatment or not. The convenience in incorporating other devices with CNN to provide a much more efficient output also plays a major role in the long run.

The future for this project lies in the creation of hardware mechanisms that can duly conspire with the results provided by CNN and apply proper diagnosis to the patient based on that result. It can subjectively reduce the stress placed on the examiners and create a much beneficial treatment domain for the respective sufferer.

ACKNOWLEDGEMENT

We take this opportunity to acknowledge all the people who have helped us kind-heartedly in every stage of this project. First, we thank the management of NSS College of Engineering, Palakkad and Dr. P. R. Suresh, Principal, NSS College of Engineering for the encouragement and support for the completion of the project. We would also like to express our profound gratitude to Dr. Binu G S, Head of the Department of ECE and all the staffs of department for their valuable feedback. We express our sincere gratitude to Asst. Prof. Reshmi S, Department of Electronics and Communication for her valuable support towards the completion of the project. Last, but not the least, we take pleasant privilege in expressing our heartfelt thanks to my friends who were of precious help in completing this project successfully.

REFERENCES

[1] Bhardwaj, Charu, Shruti Jain, and Meenakshi Sood." Diabetic retinopathy severity grading employing quadrant-based Inception-V3 convolution neural network architecture." International Journal of Imaging Systems and Technology, pp. 592-608, 2021.
[2] Deshpande, Abhishek, and Jatin Pardhi." Automated detection of Diabetic Retinopathy using VGG-16 architecture", 2021
[3] H. Pratt, F. Coenen, D. M. Broadbent, S. P. Harding, and Y. Zheng," Convolutional neural networks for diabetic retinopathy," Procedia Comput. Sci., vol. 90, pp. 200-205, Jan. 2016.
[4] Alexandr Pak, Atabay Ziyaden, Kuanysh Tukeshev, Assel Jaxylykova & Dana Abdullina (2020) Comparative analysis of deep learning methods of detection of diabetic retinopathy, Cogent Engineering, 2020.
[5] N. Cheung, P. Mitchell, and T. Y. Wong," Diabetic retinopathy," Lancet vol. 376, no.9735, pp. 124- 136, 2010.
[6] Hemanth, O. Deperlioglu, and U. Kose," An enhanced diabetic retinopathy detection and classification approach using deep convolutional neural network," Neural Comput.Appl., vol. 32, no. 3, pp. 707–721, 2020.