Diabetic Retinopathy Detection and Classification

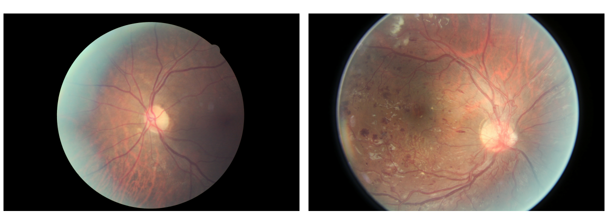
A P Vaideeswaran[1]

***Abstract*— Diabetic retinopathy (DR) is the leading cause of blindness in the working-age population. The aim is to detect and classify DR into four stages if the disease exists thereby, estimating the severity even before visiting a doctor. This is crucial as the disease barely presents any symptoms in the early stages. Early medical attention can prevent loss of eyesight or slow down the disease. The patient's condition can be monitored remotely using such models. DR detection is time-consuming and requires sizable expertise and infrastructure. Therefore, such models can help doctors and patients alike, giving the health sector time to improve the infrastructure required to cater to the needs of an increasing diabetic population, especially in rural areas. In this paper, we compare the performance of various image classification deep-learning models like VGG16, ResNet50, EfficientNetB3, and EfficientNetB4. The best model shall be identified and further be enhanced.**

I. INTRODUCTION

The World Health Organization estimates that 347 million people have the disease worldwide. Diabetic Retinopathy (DR) is an eye disease associated with long-standing diabetes. Present DR detection is costly regarding time and manual effort, where a clinician examines and evaluates digital color fundus photographs of the retina. When reviews are submitted a few days later, the delayed results lead to lost follow-up, miscommunication, and delayed treatment. Clinicians identify DR by observing vascular abnormalities that manifest in the form of lesions. While this approach is effective, its resource demands are high.

Therefore, there is a need for an effective automated method of DR screening based on ML image classification algorithms. Colored fundus have sufficient resolution, which, when trained upon, might result in the models even having clinical potential.



(a) (b)

Fig. 1: Fundus photographs of the retina: (a) Healthy Eye; and (b) Proliferative Diabetic Retinopathy

VGG16, ResNet50, EfficientNetB3, and EfficientNetB4 are all Convolution Neural Network (CNN) based Deep Learning Models. All of them have been used previously for classification purposes. This sets the expectation that they can handle this problem efficiently as well after training. For the ideation phase, we pursue Transfer Learning on lower-resolution images to handle the computational cost for all the models and then look at their performance on the test dataset. The Python Multiprocess Library helped to run several processes (like preprocessing and testing) parallely utilizing the complete computing power of the system. All the pre-trained methods were obtained from Keras-Applications. In this paper, we present the results of the ideation phase.

[1]Department of Electrical Engineering, Indian Institute of Technology Hyderabad, India.

II. MATERIALS AND METHODS

A Kaggle annotated public dataset comprising of retinal fundus images was used to perform transfer learning on the above mentioned four models. The images are from EyePACS - a free retinopathy screening platform.

A. Data Description

The dataset includes many high-resolution retina images taken under various imaging conditions. A left and right field is provided for every subject. Images are labeled with a subject ID and either left or right. For instance, 1_left.jpeg is the left eye of patient ID 1. A clinician has rated the presence of the disease in each image on a scale of 0 to 4, according to the following scale: 0 - No DR 1 - Mild 2 - Moderate 3 - Severe 4 - Proliferative DR.

B. Preprocessing Methods

Adaptive cropping with thresholding has been done on the images. In other words, if a column has more than 95 % (threshold) black pixels, that column can be removed. Each image was then resized down to 200x200. Resizing without cropping was done before the testing phase as well.

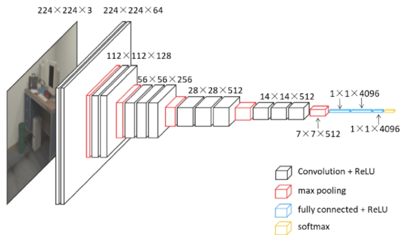
C. Transfer Learning

The Transfer Learning was done keeping 16 nodes in the penultimate layer with the ReLU Activation Function. The output layer had 5 nodes corresponding to the 5 classes with the Soft Max Activation Function for 10 Epochs. The loss function employed was the Categorical Cross Entropy, with the Optimization being done using the Adam Optimizer. The chosen batch size was 32.

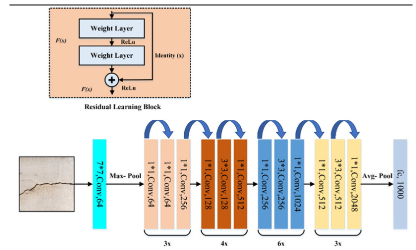
Softmax Function for a given class s_i: f(s_i) = $\left(\frac{e^{s_p}}{\Sigma_j^C e^{s_j}}\right)$ Where s_j are the scores inferred for each class in C.

ReLU Function: f(x) = max(0,x)

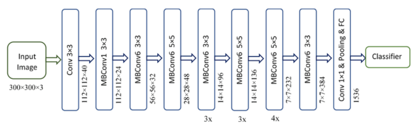
Accuracy = $\frac{\Sigma|y_i-\hat{y}_i|}{N}$ where y is the ground truth, $\hat{y}$ is the predicted label, and N is the size of the test over which



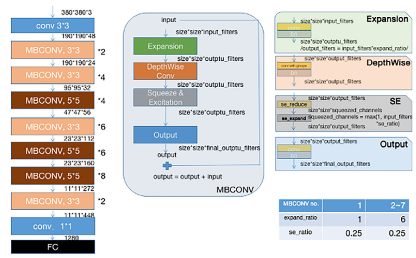
(a)



(b)



(c)



(d)

Fig. 2: ML models: (a) VGG16; and (b) ResNet50 (c) EfficientNetB3; and (d) EfficientNetB4

accuracy is being found.

Categorical-Cross Entropy $= -\log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right)$ where s_p is the CNN score for the positive class.

D. Outcome Prediction

Each of the models predicts a class label with a certain probability. We choose the class that corresponds to the maximum probability. The absolute difference between the predicted and true labels was taken across the entire set and averaged out. This metric shall then be used to compare between the models.

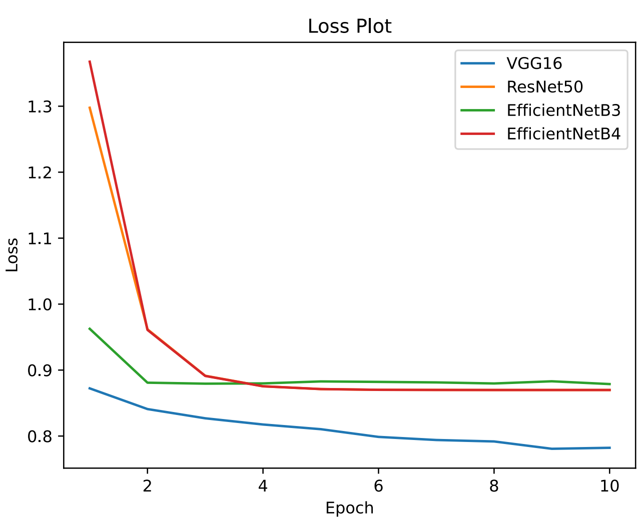


Fig. 3: Loss Plot

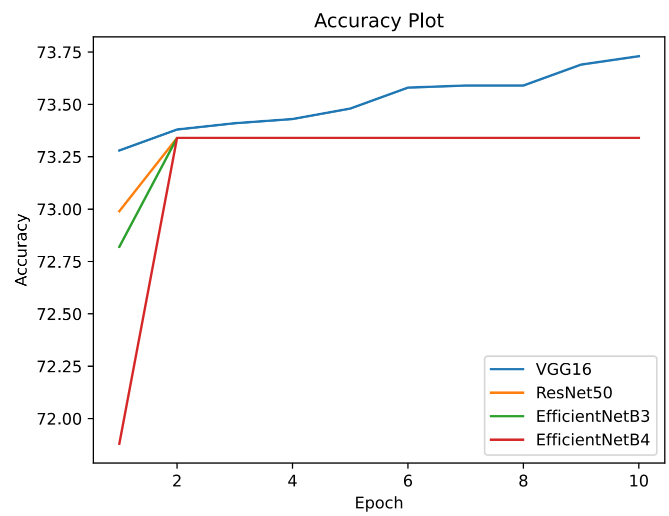


Fig. 4: Accuracy Plot

III. Experimental Results

The performance of all the models on the Test Images was evaluated before and after transfer learning. The following results were obtained:

Models - Before Training			
VGG16	ResNet50	EfficientNetB3	EfficientNetB4
2.4172	1.7261	2.8894	2.4217

Models - After Transfer Learning			
VGG16	ResNet50	EfficientNetB3	EfficientNetB4
1.0606	3.4211	1.6133	0.5789

Performance Change			
VGG16	ResNet50	EfficientNetB3	EfficientNetB4
1.3567	-1.695	1.2761	1.8428

IV. DISCUSSION

The number of epochs was not increased as the accuracy and loss barely improved for the models. The number of penultimate layer nodes was not increased for VGG16, ResNet50, and EfficientNetB3, owing to their poorer test performance improvement than EfficientNetB4. The train accuracies are not changing significantly for all the models - a possible limitation of Transfer Learning (only so much can be learned by training the last set of layers). EfficientNetB4 is still performing better despite having comparable accuracies to other models, as it might provide a ”less” incorrect answer.

To improve the performance of EfficientNetB4, all the model layers must be trained to suit the exact purpose. The weights of all the models mentioned above were based on classifying the ImageNet Dataset. While we are still dealing with a classification problem, nuances specific to this task can only be learned by training on this specific dataset. This exercise might be more fruitful than simply adding more layers in the front. Since we are confident about the learning capacity of EfficienetNetB4, we can train the entire Network on 500x500 Images. These images would also provide the necessary resolution to classify the rarer and serious cases of the disease.

V. REFERENCES

- Reference Problem Statement
- Link to the Dataset
- VGG16 architecture image
- ResNet50 architecture image
- EfficientNetB3 architecture image
- EfficientNetB4 architecture image

VI. CODE SUBMISSION

Code Files