PROJECT REPORT
TOPIC: DIABETIC RETINOPATHY

Rashi B
Undergraduate Student
Department of Computer Science & Engineering
IIIT Naya Raipur,India

Rohithram R
Undergraduate Student
Department of Electrical Engineering
IIT Madras, India

Vishnu B
Undergraduate Student
Department of Electrical Engineering
IIT Madras, India

Abstract: This report presents the deep analysis of the detection of the disease called Diabetic Retinopathy. Convolutional Neural Network (ConvNets or CNNs) are neural networks made up of neurons with learnable weights and biases. CNN gives best performance in image and pattern recognition problems[1]. This inspired us to apply CNN for our image classification problem, diabetic retinopathy(DR). Diabetic Retinopathy is a common disease among the patients suffering from diabetes. The automatic recognition will help us to detect the disease at an early stage and help the patient get medication accordingly. We have used images from Kaggle, Methods to Evaluate Segmentation and Indexing techniques in the field of Retinal Ophthalmology(Messidor) and Indian Diabetic Retinopathy Image Dataset (IDRiD) to train the images with our algorithm. We have used both learning from scratch as well as transfer learning to train our images and achieved a top accuracy of 59.45% and 54.3% respectively.

Keywords: Diabetic Retinopathy, Deep Learning, Convolutional Neural Networks

INTRODUCTION

Diabetic Retinopathy(DR):

Diabetes has become one of the rapidly increasing health threats worldwide[2]**.** According to the reports of Jan 28, 2016 diabetes currently affects an estimated **143 million people** worldwide and the number is is growing rapidly. In India, about 5 percent population suffers from diabetes[3].

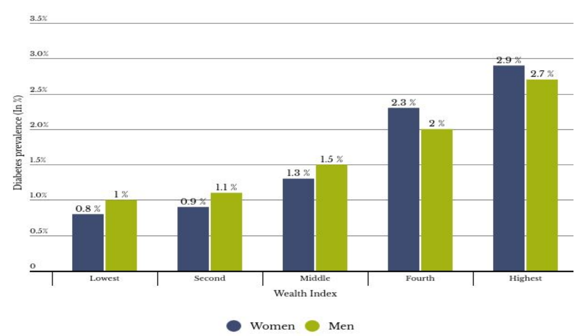

Fig1: Diabetes Prevalence In India, By Wealth Brackets[3]

People with diabetes are often prone to a common eye disease called diabetic retinopathy. This is when high blood sugar levels cause damage to blood vessels in retina. These blood vessels swell and leak or can close which does not allow blood to pass through. Sometimes new abnormal blood vessels grow in the retina and this can be a cause to loss of vision.

Diabetic retinopathy (DR) is classified into two major classes such as non-proliferative DR (NPDR) and proliferative DR (PDR) [15]. The NPDR is further subdivided according to the level of damage in the retina into three different stages such as mild, moderate and severe[16][17].

1. ***Mild nonproliferative retinopathy*:** Small areas of balloon-like swelling in the retina's tiny blood vessels, called microaneurysms, occur at this earliest stage of the disease. These microaneurysms may leak fluid into the retina.

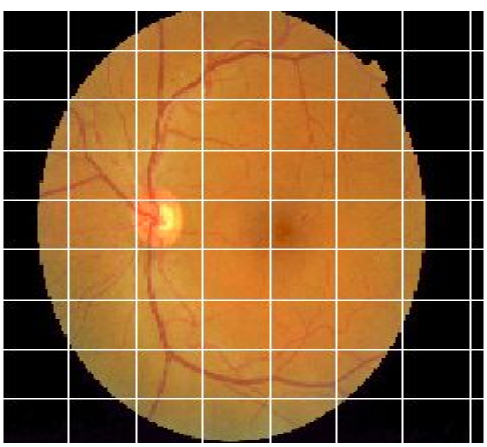

Fig 2: Mild nonproliferative retinopathy[12.a]

2. ***Moderate nonproliferative retinopathy:*** As the disease progresses, blood vessels that nourish the retina may swell and distort. They may also lose their ability to transport blood. Both conditions cause characteristic changes to the appearance of the retina and may contribute to Diabetic Macular Edema (**DME**).

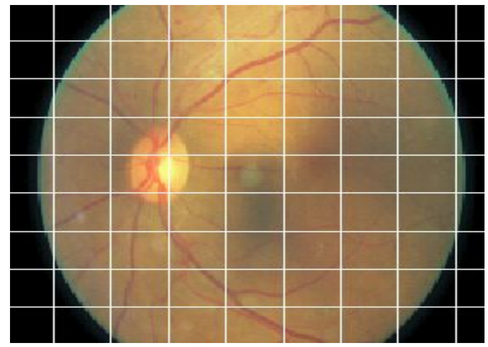

Fig3: Moderate nonproliferative retinopathy[12.a]

3. ***Severe nonproliferative retinopathy:*** Many more blood vessels are blocked, depriving blood supply to areas of the retina. These areas secrete growth factors that signal the retina to grow new blood vessels.

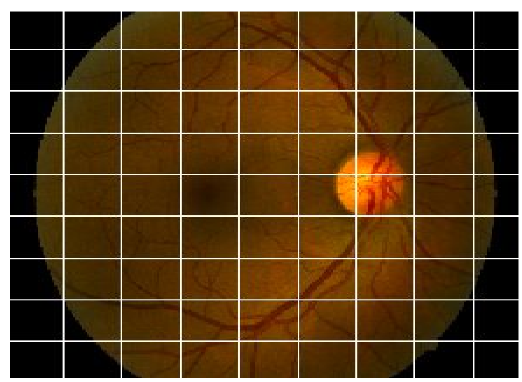

Fig 4: Severe nonproliferative retinopathy[12.a]

4. ***Proliferative diabetic retinopathy (PDR):*** At this advanced stage, growth factors secreted by the retina trigger the proliferation of new blood vessels, which grow along the inside

surface of the retina and into the vitreous gel, the fluid that fills the eye. The new blood vessels are fragile, which makes them more likely to leak and bleed. Accompanying scar tissue can contract and cause retinal detachment—the pulling away of the retina from underlying tissue, like wallpaper peeling away from a wall. Retinal detachment can lead to permanent vision loss.

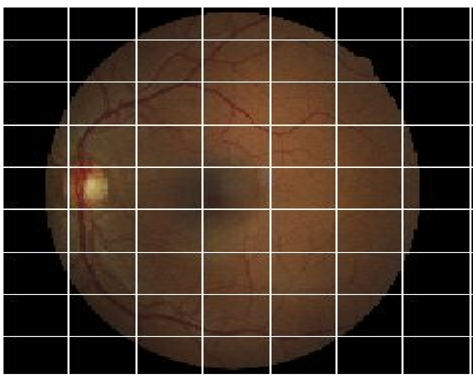

Fig 5: Proliferative diabetic retinopathy[12.a]

5. **No DR:** Below is the image of a person who does not have diabetic retinopathy.

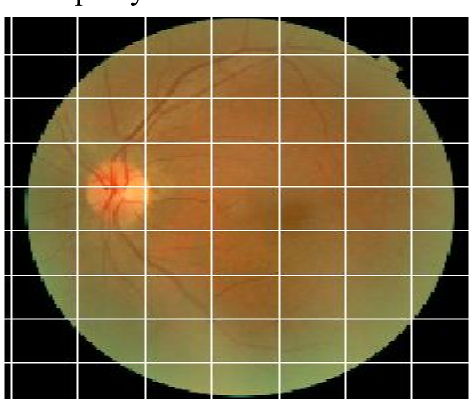

Fig 6 : No DR[12.a]

DATASETS

We have used images from public repositories such as Kaggle, Messidor and IDRiD. The images from Kaggle[12.b] and IDRiD[12.a] have 5 classes whereas the images from Messidor[12.c] dataset has 4 classes.

About Kaggle Dataset: The images in the Kaggle dataset come from different models and types of cameras, that can affect the visual appearance of left vs. right. The appearance of images differ. Some images appear as if one would see the retina anatomically(macula on the left, optic nerve on the right for the right eye). Others are shown as if one would see through a microscope condensing lens(i.e. inverted, as one sees in a typical live eye exam). The kaggle dataset has 35,000 images.

About Messidor Dataset: The 1200 eye fundus color numerical images of the posterior pole for the MESSIDOR database were acquired by 3 ophthalmologic departments using a color video 3CCD camera on a Topcon TRC NW6 non-mydriatic retinograph with a 45 degree field of view[6]. The 1200 images are packaged in 3 sets, one of which is according to ophthalmologic department. Each set is divided into 4 zipped sub sets containing each 100 images in TIFF format and an Excel file with medical diagnoses for each image.

About IDRiD Dataset: It consists of 516 images with image level expert grades for DR and DME. IDRiD, is the first database representative of an Indian population. It is the only dataset constituting typical diabetic retinopathy lesions and normal retinal structures annotated at a pixel level. This dataset provides information on the disease severity of diabetic retinopathy and diabetic macular edema for each image.

Convolutional Neural Networks:

In neural networks, Convolutional neural network are used to perform image recognition, image classification, object detection, face recognition etc. CNN takes a image as an input, processes it and classifies it under certain classes eg., No DR, Severe DR, Mild DR etc. [20]

However, it wasn't until several breakthroughs in neural networks such as the implementation of dropout[4], rectified linear units[5] and the accompanying increase in computing power through graphical processor units (GPUs) that they became viable for more complex image recognition problems. Presently, large CNNs are used to successfully tackle highly complex image recognition tasks with many object classes to an impressive standard.

The main concern working with DR is to get better results for sensitivity(patients correctly identified as having DR) and specificity(patients correctly identified as not having DR). This is significantly harder for a broader criteria which has five classes i.e normal, mild DR, moderate DR, severe DR and proliferative DR classes. Overfitting is a problem which occurs when we train the algorithm with a lot of data. The algorithm then starts to learn from the noise. The model thus trained is not efficient[24]. This condition is prominent while working with skewed dataset.

A typical structure of CNN consists of Convolutional layer, Pooling layer, Dropout layer, Activation Function and the Output Layer[19].

A. Convolutional Layer

The convolutional layer extracts features from the image with the help of the weight matrix. The weight matrix runs across the image such that all the pixels are covered at least once to give a convoluted output. The weight matrix behaves like a filter in an image extracting particular information from the original image matrix. A weight combination might be extracting edges, while another one might extract a particular color, while another one might just blur the unwanted noise.

The weights are learnt such that the loss function is minimized. Therefore weights are learnt to extract features from the original image which help the network in correct prediction. When we have multiple convolutional layers, the initial layer extract more generic features, while as the network gets deeper, the features extracted by the weight matrices are more and more complex and more suited to the problem at hand.

B. The Pooling Layer

When the images are too large, we would need to reduce the number of trainable parameters. It is then desired to periodically introduce pooling layers between subsequent convolution layers. Pooling is done for the sole purpose of reducing the spatial size of the image. Pooling is done independently on each depth dimension, therefore the depth of the image remains unchanged. The most common form of pooling layer generally applied is the max pooling with a filter size of 2x2. Max pooling takes the largest element over the 4 numbers of the little 2x2 region rectified feature map.

C.Dropout Layer

A simple and powerful regularization technique for neural networks and deep learning models is dropout. Dropout is a technique where randomly selected neurons are ignored during training. They are "dropped-out" randomly. The effect is that the network becomes less sensitive to the specific weights of neurons. This in turn results in a network that is capable of better generalization and is less likely to overfit the training data.

D.Activation Function

It is used to determine the output of neural network like yes or no. It maps the resulting values in between 0 to 1 or -1 to 1 etc.(depending upon the function). The Activation Functions can be basically divided into 2 types. Linear Activation Function and non-linear Activation Functions. Sigmoid or logistic activation function, Tanh or hyperbolic tangent function, ReLU activation function, Leaky ReLU comes under non-linear activation function. We have used ReLU as our activation function. ReLU is half rectified from bottom. It ranges from 0 to infinity.

E.Output Layer

To generate the final output we need to apply a fully connected layer to generate an output equal to the number of classes we need. It becomes tough to reach that number with just the convolution layers. Convolution layers generate 3D activation maps while we just need the output as whether or not an image belongs to a particular class. The output layer has a loss function like categorical cross-entropy, to compute the error in prediction since detecting DR is a multi-class classification problem. The activation function used in the output layer is Softmax. Softmax will calculate the probabilities of each target class over all possible target classes. Later the calculated probabilities will be helpful for determining the target class for the given inputs.

In addition to the above layers, another important term often used in a convolutional network is "Strides". A stride is the number of pixels the kernel/filter shifts over the input matrix. When the stride is 1 then we move the filters by 1 pixel at a time. When the stride is 2 then we move the filters by 2 pixels at a time and so on[23].

LITERATURE REVIEW

In literature, several research groups have attempted to develop a model to identify the disease, diabetic retinopathy. Neural Networks have also been used in three-class classification of DR. Some researchers have used features such as the area of exudates and the area of blood vessels together with texture parameters(Nayak et al.[9]). Features are entered into the neural network to classify images into normal, non-proliferative retinopathy and proliferative retinopathy. The neural network used these features as input for classification. The detection results were validated by comparing with grading from expert ophthalmologists. Classification accuracy of 93%, sensitivity of 90% and specificity of 100% was achieved. The limitation with their model was that a dataset of 140 images and feature extraction was required on all images in both training and testing which was time consuming(Nayak et al.[9]). Few researchers (Syahidah izza Rufaida et al.[7]) used Residual Neural Network with L2 as its regularization. A 8 layer filters were used. A shortcut was created from the sixth convolution layer to the eighth layer which helped the authors to achieve a kappa score of 0.51049. Researchers have achieved 0.386 weighted kappa accuracy using Histogram equalization and Min-Max Normalization (Doshi et al.[8]). They used only single channel instead of multiple channels to train their images. Some researchers(Adarsh et al.[10]) have also used image processing techniques to produce an automated diagnosis for DR through the detection of retinal blood vessels, exudate, micro-aneurysms and texture features. The area of lesions and texture features were used to construct the feature vector for the multi-class SVM. This achieved accuracy of 96%. Each of the previous five class methods required feature extraction from the images before being input to an SVM classifier and have only been validated on small test sets of approximately 100 images. These methods are less real-time applicable than a CNN. CNNs are prone to have high accuracy. Therefore CNN's are used for image classification. In 2016, few researchers(Gulshan et al.[11]), reported the results of the Google DL DR validation study. The algorithm was trained using 128,175 macula-centered retinal images obtained from EyePACS[22] in the United States and retinal images from eye hospitals in India. They achieved a highest accuracy of sensitivity i.e 97.5% and specificity i.e 93.4% for DR. This inspired us to choose CNN for our project.

SOLUTION & METHODOLOGY

It has been observed that several people diagnosed with DR are in the later stage of the disease[21]. This indicates that busy Opthamologists might not be able to detect the earliest stage of DR. Therefore we need a computer aided intelligent model which can help us to detect DR so that proper treatment can be taken. This motivated us to develop a deep learning model which can detect Diabetic Retinopathy.

These are the different datasets we used to train the model. [12]

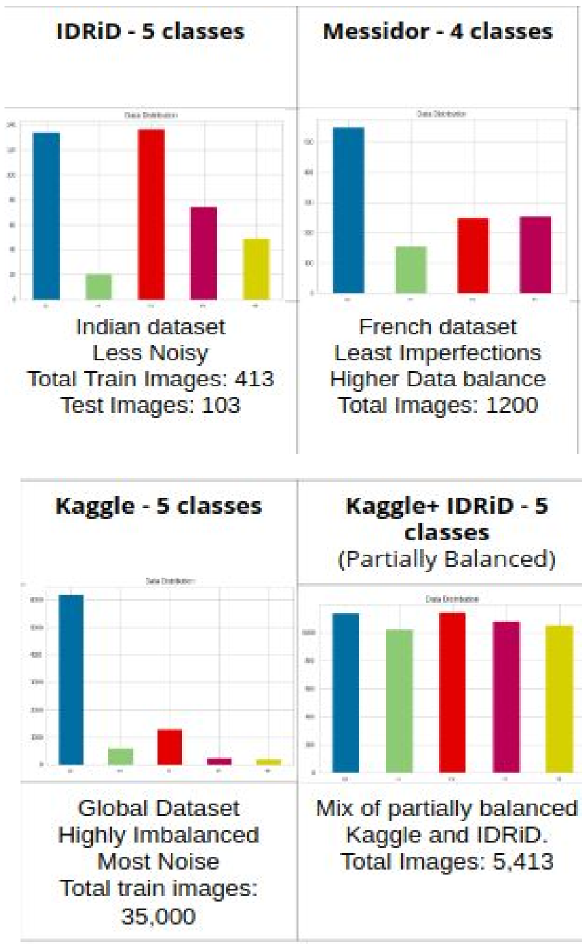

Fig 7: Dataset Distribution

Data Preparation and Class Balancing

Due to the highly imbalanced nature of the Kaggle dataset of total 35,000 labelled samples, we randomly sampled 1000 images from each of the majority class 0, 1 and 2 as we observed from the data distribution given previously and oversampled the minority class 3 and 4 to increase it to 1000 images, so that

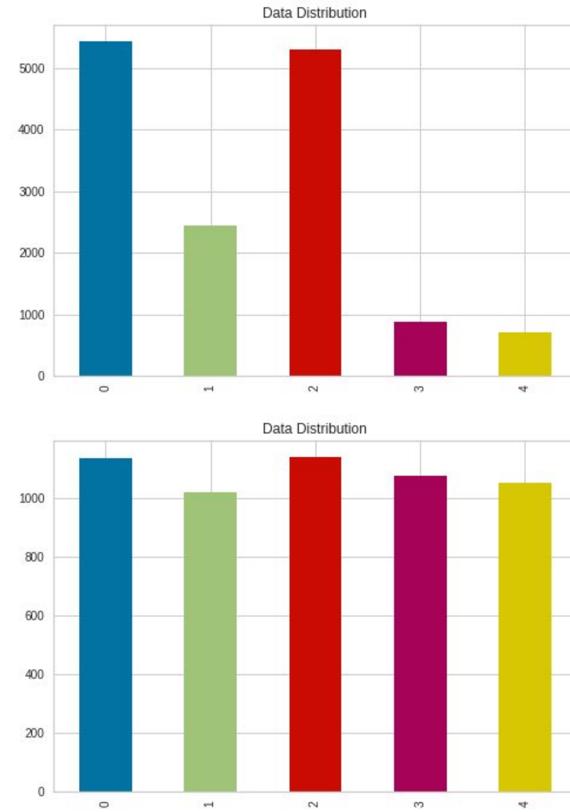

Fig 8 :Top image: Before balancing data
Below image: After Class Balancing.

they add up to 5000 with 1000 from each class.

To prevent the model from learning features biased to foreign dataset, we mixed IDRiD training dataset which consists of 413 labelled samples to the 5000 images already created, thus final dataset has 5413 labelled images.

Data Preprocessing

Due to non-standard image resolutions, the training images could not be utilized directly for training. So the images were scaled down to a fixed resolution size of 224 x 224 x 3 pixels to form a standardized dataset since these are one of the common input dimensions used in many of the popular models like vgg16, vgg19, resnet50, etc and also due to memory constraints with larger input dimensions.

We used a technique called Contrast Limited Adaptive Histogram Equalization (**CLAHE**) on the images to improve the contrast and to enhance the definitions of edges in each region of an image, so that the subtle features like exudates and microaneurysms, etc. are easier to learn by the model.[8]

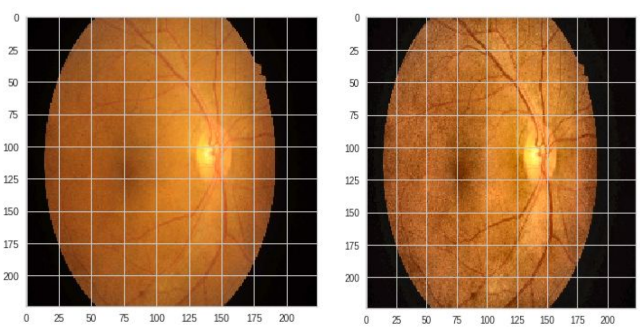

Fig 9: Left image is before applying CLAHE and right one is after preprocessing.

Data Augmentation

Since the dataset has lesser number of labelled samples, we used data augmentation to increase the training data and also to make the classifier robust to different variants of retinal images which in turn helps in generalizing the model.

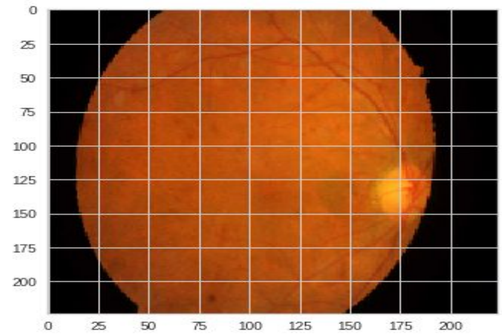

Fig 10: Augmented Image

We used Imagedatagenerator function in Keras library for augmenting images and we rotated images randomly by 5 or 10 degrees, flipped images horizontally and vertically, scaled the pixel values to 0-1 by dividing all pixels by 255.

CNN Architectures

We used 5 different architectures for Diabetic Retinopathy Classification on balanced Kaggle + IDRiD dataset.

First off, Model1 : Normal 5 Layer CNN whose architecture is given as follows.

Layers	Model1
input	224x224x3
Conv1	220x220x32
pool1	110x110x32
conv2	106x106x64
pool2	53x53x64
conv3	49x49x64
pool3	24x24x64

dropout3	24x24x64
conv4	20x20x128
pool4	10x10x128
dropout4	10x10x128
dense1	512
dropout5	512
output	5

Table 1: Between conv & max pool layers, Batch normalization + relu activation is present and Dropout rate of 0.5 is used wherever mentioned.

Model2 being a variant of Vgg 16 with Dilated convolution layers.[13]

VGG16:Block4 Conv Layer

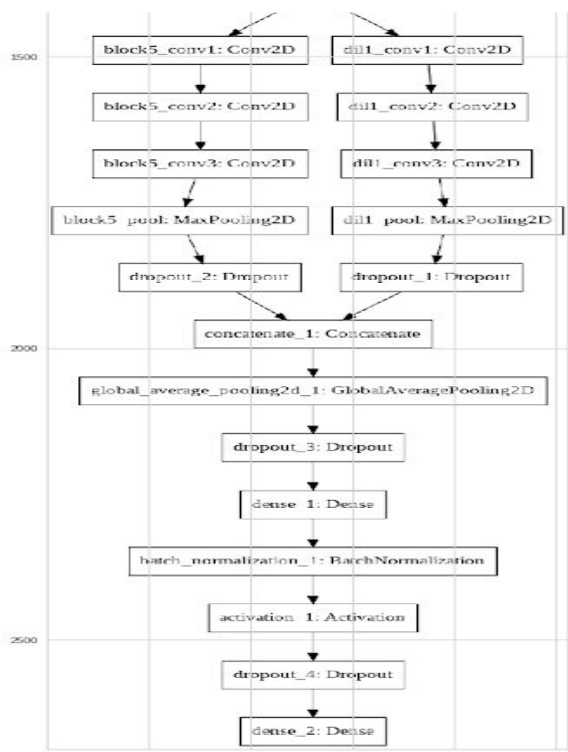

Fig 11:Architecture of Model 2

Rest of the architectures we trained on are Resnet18, Vgg16, Inception V3, etc.

DETAILS OF LEARNING

Dataset is split into train, validation and test images in the ratio of 60:20:20 out of 5413 images.

The train and validation images are fed into ImageDatagenerator function in keras with transformations already mentioned for training data and only rescaling the pixel values for validation data respectively.
To initiate the training process, it was essential to find the ideal batch size that would fit the RAM and GPU. We used Google Colab for the entire learning process. So we found that batch size of 32 suited well.

Since this is a multi-class classification problem, we used softmax as the activation function instead of relu after the output layer. We used the Categorical Cross-Entropy as the loss function. The function below denotes the categorical cross-entropy between the predicted classes and target classes.

$$L_i = \sum_j t_{i,j} log(P_{i,j})$$

To minimize the loss function mentioned above, the models were trained using ADAM optimizer with a batch size as mentioned previously[14]. The number of epochs for the training process was 250.

In the Adam optimizer, the momentum parameter was set to 0.9 throughout the training. In order to remove the fluctuations in final weights caused due to variations in batches, an adaptive learning rate relative to the number of epochs was employed instead of a fixed learning rate. The adaptive learning rate schedule was used, with learning rate of 0.001 initially and reduced by a factor of 0.5 when validation loss doesn't improve for 10 epochs at a stretch, this is implemented using ReduceLROnPlateau function in Keras library.

The weights and biases were initialized using random-uniform initialization, sampled from the uniform distribution.

Overfitting is a common problem faced in deep networks. To reduce overfitting, in addition to data augmentation, dropouts were utilized after some layers with 0.5 as *keeping probability* described in the network previously and in some cases weight regularization of 0.001 was also used to prevent overfitting.

IMPLEMENTATION DETAILS

The various data pre-processing techniques were performed using Keras's ImageDataGenerator function (a tool for augmenting images) and the Python library OpenCV.

To speed up the process of training the mentioned CNN architectures, the entire training process was performed using Tesla k80 series GPU with memory of 10GB and RAM of 12GB hosted by Google Colab.

In order to work with the GPU, Tensorflow backend and Keras as frontend framework were used. To design and train the deep networks, the Python libraries like Scikit-Learn, Numpy, Pandas were used.

RESULTS

In this project, it is not only important to measure the number of correctly and incorrectly classified test images, but also to evaluate by how many classes the images were misclassified and to penalize the accuracy accordingly. Hence just accuracy is not a well defined metric for evaluating the model. So we use other metrics like precision, recall, F1-score to give importance to not only to the True positives and True negatives but also to False positives and False negatives, thus calculating sensitivity and specificity for each model for comparison.

We also found confusion matrix on the predicted classes on test data for each model. For a good model, confusion matrix should have entries densely distributed in and around the main diagonal and the off-diagonal entries should be close to zero.

We trained various models such as Vgg16, Resnet18, InceptionV3, Vgg16 with dilated convolution layers, 5 Layer CNN etc.

Since training deep networks on our small dataset might lead to overfitting, we have used transfer learning for training large networks like InceptionV3, Vgg16 etc. The pretrained models are taken from Keras with imagenet weights and top layers are dropped and custom dense layers are added according to our objective. Then initial layers are locked and trained on last few layers for fine tuning the models.

Relation between Sensitivity, Specificity, Precision, Recall [14]

Sensitivity = Recall

$$Tp = \frac{Tp}{Tp + Tn}$$

Specificity

$$Tn = \frac{Tn}{Tn + Tp}$$

Precision

$$Tp = \frac{Tp}{Tp + Fp}$$

Model Performance on Balanced Kaggle + IDRiD Dataset

Type of Model	Sensitivity	Specificity	F1-Score	Accuracy
Vgg-16	51	52	51.5	52.7%
Resnet-18	53	53.41	52	52.45 %
Vgg-16 plus Dilated conv	53	51`	51.9	54.3%
Normal 5-Layer CNN	**59**	**60**	**59**	**59.45 %**

Table 2 shows : Performance of given models on the data, where sensitivity, specificity, F1-score are the micro averages of 5 classes, since its a multi-class classification problem.

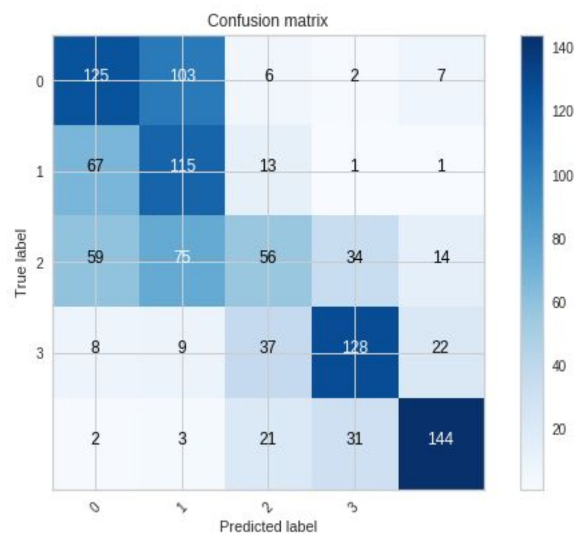

Fig 12:Confusion matrix obtained from Resnet-18 on balanced Kaggle plus IDRiD Data which is trained from scratch.

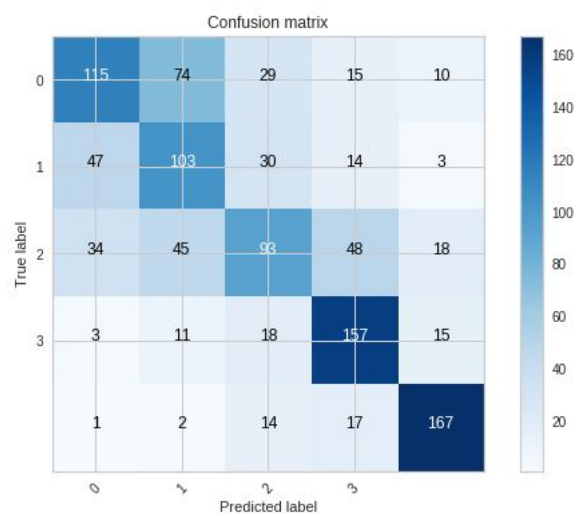

Fig 13: Confusion Matrix obtained from Custom model 5- Layer CNN on balanced Kaggle plus IDRiD data which is trained from scratch.

CONCLUSION

We have experimented with the above mentioned datasets i.e Kaggle, Messidor and IDRiD. We were able to achieve accuracy of 59.45% using the model with 5 layer CNN Network. Transfer learning did not give us better results because retinal fundus images are not related to imagenet dataset and it was a small dataset of 5413 images. So the features learnt by the pretrained models from imagenet are not useful in detecting DR, so transfer learning didn't give good results. Therefore we wish to work with a larger dataset in future with good computational resources and train popular models like resnet50, inceptionv3 from scratch and achieve sensitivity and specificity of more than 90%.

REFERENCES

1. Samer Hijazi, Rishi Kumar, and Chris Rowen, IP Group, Cadence , "Using Convolutional Neural Networks for Image Recognition" , 2015 Cadence Design Systems, Inc. 5152 09/15 SA/DM/PDF
2. Gunvor von Wendt, Screening for diabetic retinopathy: Aspects of photographic methods. PhD thesis, Karolinska Institutet, 2005.
3. https://www.firstpost.com/india/diabetes-is-indias-fastest-growing-disease-72-million-cases-recorded-in-2017-figure-expected-to-nearly-double-by-2025-4435203.html *Accessed on 9th January, 2019*
4. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.. Dropout: A simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15(1):1929–1958
5. Nair, V., Hinton, G.E.. Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10). 2010, p. 807–814
6. Etienne Decenciere et al, "Feedback on a publicly distributed image database: The Messidor Database", Received April 18, 2014; revised June 30, 2014; accepted July 31, 2014
7. Syahidah izza Rufaida, Mohamad Ivan Fanany, "Residual Convolutional Neural Network for Diabetic Retinopathy", Conference Paper · October 2017 DOI: 10.1109/ICACSIS.2017.8355060
8. Darshit Doshi et al, "Diabetic Retinopathy Detection using Deep Convolutional Neural Networks", 2016 International Conference on Computing, Analytics and Security Trends (CAST) College of Engineering Pune, India. Dec 19-21, 2016
9. Nayak, J., Bhat, P.S., Acharya, R., Lim, C., Kagathi, M.. Automated identification of diabetic retinopathy stages using digital fundus images. J Med Syst 2008;32(2):107–115.

10. Adarsh, P., Jeyakumari, D.. Multiclass svm-based automated diagnosis of diabetic retinopathy. In: Communications and Signal Processing (ICCSP), 2013 International Conference on. IEEE; 2013, p. 206–210.
11. Gulshan V, Peng L, Coram M, Stumpe MC, Wu D, Narayanaswamy A, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA. 2016;316:2402–10.
12. (a) Prasanna Porwal, Samiksha Pachade, Ravi Kamble, Manesh Kokare, Girish Deshmukh, Vivek Sahasrabuddhe and Fabrice Meriaudeau, "Indian Diabetic Retinopathy Image Dataset (IDRiD)", IEEE Dataport, 2018. [Online]. Available: http://dx.doi.org/10.21227/H25W98. Accessed: Jan. 09, 2019
(b)https://www.kaggle.com/c/diabetic-retinopathy-detection/data *Accessed on 9th January, 2019*
(c) Decenciere et al.. Feedback on a publicly distributed database: the Messidor database. Image Analysis & Stereology, v.33, n.3, p.231-234, aug. 2014. ISSN 1854-5165. Available at http://www.ias-iss.org/ojs/IAS/article/view/1155 or http://dx.doi.org/10.5566/ias.1155 *Accessed on 9th January, 2019*
13. https://github.com/anisha2102/DiabeticRetinopathyDetection Accessed on 9th January, 2019
14. D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. 2015, arXiv:1412.6980v8 [cs.LG].
15. https://uberpython.wordpress.com/2012/01/01/precision-recall-sensitivity-and-specificity/ *Accessed on 9th January, 2019*
16. Bhaskaranand M et al (2016) Automated diabetic retinopathy screening and monitoring using retinal fundus image analysis. J Diabet Sci Technol 10(2):254–261
17. https://www.preventblindness.org/what-are-stages-diabetic-retinopathy *Accessed on 9th January, 2019*
18. Qaisar Abbas, Irene Fondon, Auxiliadora Sarmiento, Soledad Jiménez, Pedro Alemany, “Automatic recognition of severity level for diagnosis of diabetic retinopathy using deep visual features” Received: 6 September 2016 / Accepted: 13 March 2017 / Published online: 28 March 2017, International Federation for Medical and Biological Engineering 2017
19. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148 *Accessed on 9th January, 2019*
20. Harry Pratta, Frans Coenenb, Deborah M Broadbentc, Simon P Hardinga, Yalin Zheng, “Convolutional Neural Networks for Diabetic Retinopathy” , International Conference On Medical Imaging

Understanding and Analysis 2016, MIUA 2016, 6-8 July 2016, Loughborough, UK

21. https://www.geographyandyou.com/population/health/expansion-eye-health-services-essential-combating-diabetic-retinopathy/ *Accessed on 9th January, 2019*
22. http://www.eyepacs.com/blog/2016/12/2/google-achieves-healthcare-breakthrough-using-eyepacs-retinal-images *Accessed on 9th January, 2019*
23. https://skymind.ai/wiki/convolutional-network *Accessed on 9th January, 2019*
24. https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/ *Accessed on 9th January, 2019*