A Comparative Study on Diabetic Retinopathy Datasets for Data Accuracy Detection

Dr. Kapil Chaturvedi   
SIRT, Bhopal, M.P, India  
[kapil.rgtu@gmail.com](mailto:kapil.rgtu@gmail.com)

Dr. Vijay Bhandari  
SIRT, Bhopal, M.P, India  
[vijaysirt@gmail.com](mailto:vijaysirt@gmail.com)

Rishabh Tiwari  
SIRT, Bhopal, M.P, India  
[tiwaririshabh2023@gmail.com](mailto:tiwaririshabh2023@gmail.com)

*Abstract*—Diabetic retinopathy is a problem that impacts the eyes of diabetic people. It can affect the minute blood vessels in the back of the eye, potentially increasing vision loss. Detecting and understanding diabetic retinopathy is a challenge. Dataset 1[15] had the highest accuracy at 0.74, followed by Dataset 3 [17] at 0.72, and Dataset 2 [16] with the lowest accuracy at 0.68. However, all three datasets share the same model complexity or simplicity level, as indicated by the identical value of 3,989,285. The application of Algorithms in different datasets can give a direction to work on the classification of images. we learn more about eye diseases, these models need to keep learning too, so they stay accurate. Understanding and tackling these challenges are crucial steps towards improving the diagnosis and treatment of diabetic retinopathy.

Keywords—Diabetic Retinopathy, Diabetes, Vision Loss, Retina, Dataset Comparison, Accuracy, Model Simplicity, Comparative Analysis, Dataset Evaluation, Machine Learning, Healthcare.

# Introduction

Diabetic Retinopathy is a problem of diabetes that affects the eyes, increasing damage in blood vessels of the retina, which can cause vision loss. It also damages the light-sensitive tissue at the back of the eye.

Diabetic Retinopathy has no early warning signs and manual diagnosis is time-consuming and needs experimental clinicians to test the retinal picture[1]. Therefore, Lots of computer vision-based techniques have been proposed for the automatic recognition of DR and its different stages from retina images. It’s often late to give effective treatment because of delay. But it seems possible to detect this disease in the last few decades by machine learning model. To detect the disease from the image, we have to pre-process the image. These models/methods are not able to detect some complex features.

The DR is mainly classified into two types: a) Non-Proliferative Diabetic Retinopathy (NPDR) (early stages) and b) Proliferative Diabetic Retinopathy (PDR)(fig-e). NPDR is further classified into Mild(fig-b), Moderate(fig-c), and Severe(fig-d) stages. The mild stage is difficult to detect as compared to the moderate and severe stages because of the initial stage

The mild stage has one micro-aneurysm (MA), a small circular reddish dot at the end of blood vessels. In the Moderate stage, the MA captures into deeper layers and forms a flame-shaped hemorrhage in the retina[2]

DR is detected by the appearance of different types of lesions on a retina image. These lesions are microaneurysms (MA), hemorrhages (HM), and soft, and hard exudates (EX)[6,7].

• MA is the initial sign of DR that appears as tiny red circular dots on the retina due to the weakness of the vessel’s walls. its size is less than 125 μ m.

• HM appears as larger spots on the retina, their size is greater than 125 μ m. it is classified into two types, their name is flame [superficial HM] and blot [deeper HM]

• Hard exudates appear as bright-yellow spots on the retina caused by leakage of blood plasma. It is occurring in the retina’s outer layers.

Soft exudates appear as white spots on the retina caused by nerve fiber swelling. It is rounded in shape.

The red lesion is MA and HM, and the light/brighter lesion is soft and hard exudates (EX). Depending on the presence of these lesions, DR has five stages as the stage increases difficulty also increases in the eye retina, namely normal, mild, moderate, severe, and PDR (Proliferative Diabetic Retinopathy). DL applications are made in these processes’ classification, segmentation, recognition, fetching, and registration of the images. There are some below research which helps to accomplish this paper.

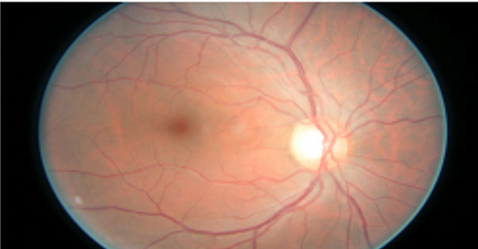
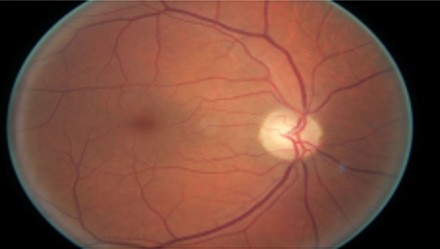
 

Fig. 1 Healthy image Fig. 2 Mild image

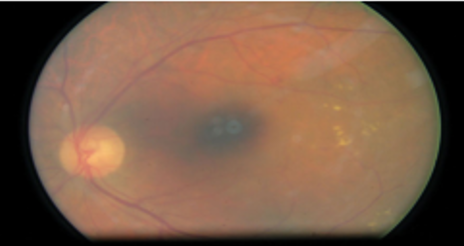
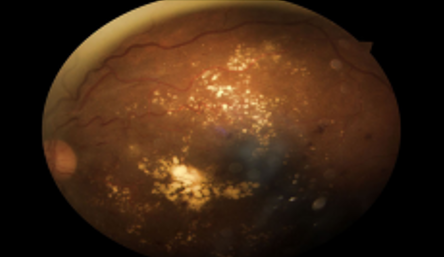
 

Fig. 3: Moderate image Fig. 4: Severe image

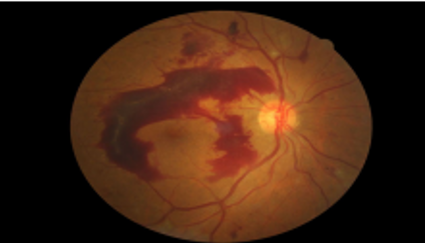


Fig. 5: Proliferate

# Literature Review

**H. Pratt** et al. [4] proposed a CNN-based method to classify images in the Kaggle dataset [26] into five DR stages. In the preprocessing stage, color normalization is done and the image is resized to 512×512 pixels. Custom CNN architectures consist of 10 CONV layers, 8 maximum pooling layers, and 3 FC layers. The SoftMax function was used to classify 80,000 test images. The specificity of the results was 95%, accuracy 75%, and sensitivity 30%. Unfortunately, CNN. failed to detect lesions in the image and only one data was used to evaluate their CNN**. T. Shanthi and R. Sabeenian** [9] detected DR levels in the Messidor dataset [10] using the pre-trained Alex net architecture [11]. This article uses convolutional neural networks to classify DR fundus images according to disease severity and uses appropriate Pooling, SoftMax, and Smoothed Linear Activation Unit (ReLU) layers to achieve accuracy. The classification accuracy of healthy images reached 96.6% and 96.2%, 95.6% and 96.6% of diabetic retinopathy stage 1, stage 2, and stage 3 images, respectively. **M. Abramoff** et al. [12] proposed CNN with an IDX-DR device to identify and classify DR images. They used data augmentation of the Messidor-2 [10] dataset containing 1748 images.

They reported the area under the curve as 0.980, sensitivity as 96.8%, and specificity as 87.0%. they use a Random Forest classifier to detect DR lesions. **S. Dutta** et al. took datasets from Kaggle [13][14] and classified them into five DR stages. Using 2000 images, they examined the performance of three networks: back propagation neural network (BNN), deep neural network (DNN), and CNN. The image was converted to 300×300 pixels and converted to grayscale and the data was extracted from the RGB image. In addition, Filter banks are used, including edge detection, median filter, morphological operation, and binary transformation before feeding into the network. Pre-training VGG16 [8] is used as a CNN architecture with 16 CONV layers 4 max-pooling layers and three FC layers, while DNN includes three FC layers. Their results showed that the DNN class outperformed well as compared to CNN.

# Challenges

## Data Quality and Quantity

Datasets are limited in size due to the difficulty of collecting high-quality labelled samples making it a challenge to models. Ensure accurate and proper labeling of images.

## Class Imbalance

* Most pictures are of healthy eyes, not sick ones. This can make the computer think everyone's eyes are fine, even when they're not. so we have to train our model.

## Preprocessing Challenges:

Eye pictures often have messy, blur in them, like shadows. We need to clean up these pictures so the computer can understand them better.

## Model Interpretability

It's tough to know why the computer says what it does. We want to make sure doctors can trust it and understand why it makes certain decisions.

## Transfer Learning and Fine-Tuning:

Choosing an appropriate pre-trained model architecture. It may require more expertise to select the right architecture and take it to the medical domain.

## Continual Learning:

As medical knowledge evolves, models need to be continually updated and retrained to stay accurate and up-to-date. We have to keep checking if the computer model is still good and accurate as new information comes in.

# Proposed Work

Designing a model for diabetic retinopathy classification involves several steps.

## Data Collection and Preprocessing:

#### First, collect many pictures of eyes, especially those with diabetic retinopathy. Make sure each picture has a label telling us how severe the problem is.

## Data Splitting:

Split the pictures into three groups: one to teach the model (training), one to help it get better (validation), and one to test how good it is (testing).

## Data Augmentation

Sometimes, we need to change the pictures a little bit to help the computer understand them better. We might flip them, rotate them, or make them brighter.

## Model Selection and Adaptation:

Pick a special computer program that's already good at understanding pictures. Think of it as a smart assistant. Teach the computer program about eyes and how to tell if they're sick or not.

## Optimizer, Loss Function & Training:

Create a scorecard to see if the computer program is good at telling which eyes are sick and which are healthy.

Choose an appropriate loss function for multi-class classification, such as categorical cross-entropy

•Select an optimizer (e.g., Adam, SGD) to update the model's parameters during training.

•Train the model on the training data.

## Security:

Make sure people's eye pictures and health information are kept private and follow the rules for using medical data. Figure out why the computer thinks an eye is sick or not. Look at the parts of the picture it pays attention to. If the computer program works well, use it in clinics or hospitals to help doctors. Keep an eye on it to make sure it's always correct. Think about being fair to everyone and making sure the computer doesn't make mistakes that could hurt people.

Deep learning uses convolutional neural networks (CNNs) to perform multi-class tasks. CNNs automatically learn hierarchical features from data, effectively eliminating multiple levels of abstraction. This allows them to analyze complex data such as images or sequences across multiple categories, making them robust for a variety of challenges.

Algorithm.

# **Algorithm: DRDA**

**(Diabetic Retinopathy Dataset Accuracy) Algorithm**

Step 1: **Import DS**

// Import the Diabetic Retinopathy Dataset.

Step 2: **Count\_Class\_Img=DS**

//Count the number of classified images in the Dataset.

Step 3: Prepare Dictionary and indexing of retina Scan images respectively.

images\_dict =

{

    'h'= list('No\_DR’) //Healthy images

    'mi'= list('Mild') //Mild images

    'mo'=list ('Moderate’) //Moderate images

    's'= list('Severe’) //Severe images

    'p'=list('Proliferate DR’) //Proliferate images

}

Index\_dict = // Labelling the classified data

{

'h'= 0,

'mi'= 1,

'mo'= 2,

'p'= 4,

's'=3,

}

Step 4: Normalize the scanned images by resizing

Define array: x, y= [], []

for (Each directory)

{

for (Each image)

{

     img= ImageRead(str(image))

Resize\_img = Resize\_img(img,(180,180))

x.append(resize\_img)

y.append(i\_labels\_dict[i\_name])

x=Array(x)

y=Array(y)

}

}

Step 5: Split the data into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, random\_state=0);

Step 6: Features extraction of scanned images stored in the dataset, from the hidden layer

layers.Conv2D(16, 3, padding='same', activation='relu'),

layers.MaxPooling2D(),

 layers.Flatten(), // Dimension reduction, transformation from 2D to 1D

Step 7: Find the Accuracy of different clusters.

Step 8: plot the results to analyze the accuracy of different clusters.

Step 9: Stop

# Experiment & Result Analysis

The output of the prediction is displayed in the form of an array. Each element of the array represents the model's confidence score for a specific class. In this case, the five classes' output is [ -689.26556, 133.92921, 1341.1217, -2194.9722, 772.93713 ].

Confusion Matrix: data of Sachin

[[108 4 7 1]

[ 3 24 19 5]

[ 3 9 90 10]

[ 0 6 14 4]]

Accuracy: 0.74

Sensitivity (Recall) per Class:

Class 0: 0.90

Class 1: 0.47

Class 2: 0.80

Class 3: 0.17

Simplicity (Model Complexity): 3989285

Table 1: Accuracy and simplicity of different dataset

|  |  |  |  |
| --- | --- | --- | --- |
|  | Dataset1[15] | Dataset2[16] | Dataset3[17] |
| Accuracy | 0.74 | 0.68 | 0.72 |
| Simplicity | 3989285 | 3989285 | 3989285 |

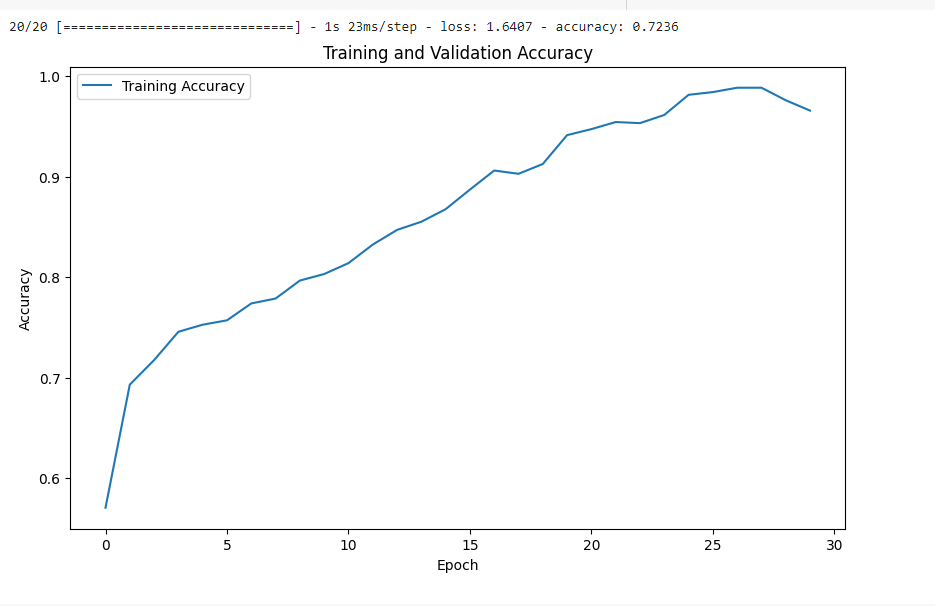


Fig. 1: Accuracy score of dataset1

15/15 [==============================] - 0s 6ms/step

Confusion Matrix:

[[210 8 5 1 3]

[ 2 23 15 2 2]

[ 4 28 65 5 14]

[ 1 2 14 4 5]

[ 2 8 22 2 11]]

Accuracy: 0.68

Sensitivity (Recall) per Class:

Class 0: 0.93

Class 1: 0.52

Class 2: 0.56

Class 3: 0.15

F1-Score per Class:

Class 0: 0.04

Class 1: 0.12

Class 2: 0.08

Class 3: 0.21

Simplicity (Model Complexity): 3989285

Accuracy: 68.34%

Sensitivity (Recall): 15.38%

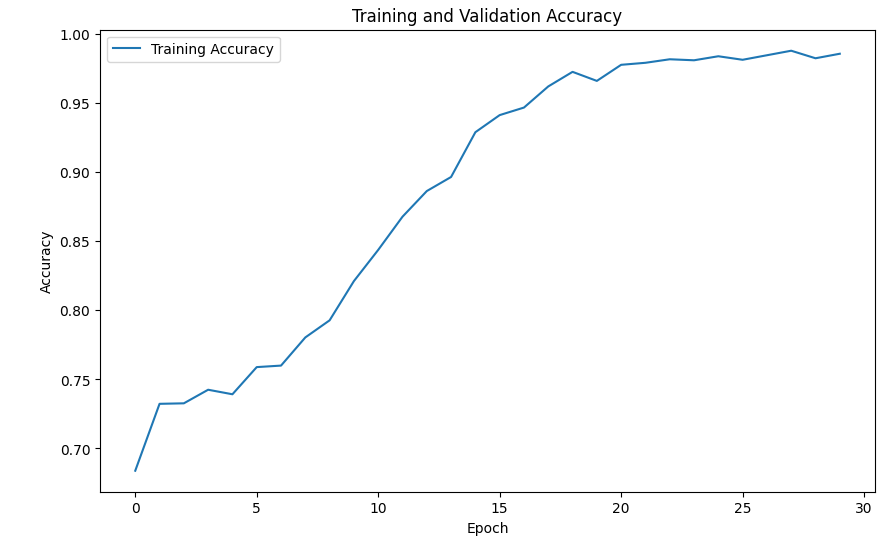


Fig. 2: Accuracy score of dataset2

Confusion Matrix:

[[218 2 6 0 1]

[ 7 17 14 4 2]

[ 10 10 80 5 11]

[ 6 4 8 7 1]

[ 7 9 23 0 6]]

Accuracy: 0.72

Sensitivity (Recall) per Class:

Class 0: 0.96

Class 1: 0.39

Class 2: 0.69

Class 3: 0.27

F1-Score per Class:

Class 0: 0.06

Class 1: 0.27

Class 2: 0.12

Class 3: 0.33

Simplicity (Model Complexity): 3989285

Accuracy: 71.62%

Sensitivity (Recall): 26.92%

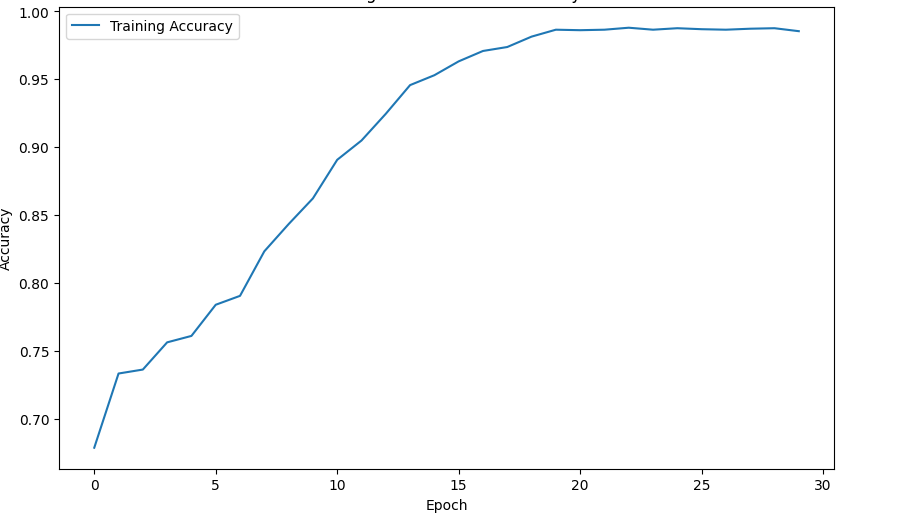


Fig. 3: Accuracy score of dataset3

Fig. 4: Classification of dataset1

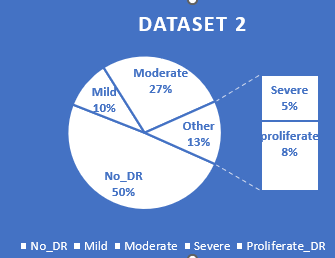


Fig. 5: Classification of dataset2

Table 2. Number of images in each class of a different dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Dataset name | NO\_DR | Mild | Moderate | Severe | Proliferate |
| Diabetic Retinopathy 224x224 Gaussian Filtered[17] | 1000 | 370 | 900 | 190 | 290 |
| Diabetic Retinopathy 224x224 [16] | 1000 | 370 | 900 | 190 | 290 |
| Diabetic Retinopathy Dataset[15] | 1805 | 370 | 999 | 193 | 295 |

# Conclusions

In this paper, we worked on different datasets and got different results while applying algorithms. we got different accuracy, simplicity, and sensitivity. At the initial stage, we have taken a small dataset and made an algorithm, as our research progresses, we will increase the dataset size, and as we increase the size of the dataset we will also get high accuracy.

Without the use of pre-trained data, feature engineering becomes a crucial aspect of model development. So we continuously work on a pretrained model to try to implement of big dataset.

The model's low accuracy on a small dataset without pre-trained data is a common challenge. However, it presents an opportunity for improvement through data augmentation, pre-trained models, and iterative model refinement. expanding dataset and transfer learning can enhance the model's predictive capability. Automated screening systems make it much faster for eye doctors to figure out what's wrong with a patient's eyes. This saves them time and money and helps patients get the right treatment quickly.

# Acknowledgements

This research work is under a state Govt. Research Project grant, this grant is sanctioned by the M.P. Council of Science & Technology, Bhopal, MP, India.

##### References

1. Zhe Wang , Yanxin Yin, Jianping Shi, Wei Fang, Hongsheng Li1, and Xiaogang Wang-- Zoom-in-Net: Deep Mining Lesions for Diabetic Retinopathy Detection the progression of diabetic retinopathy DOI: 10.1007/978-3-319-66179-7 31
2. W. R. Memon, B. Lal, and A. A. Sahto, ‘‘Diabetic retinopathy,’’ The Prof. Med. J., vol. 24, no. 2, pp. 234–238, 2017.
3. Ehrish Qummar , Fiaz gul khan, Sajid shah, Ahmad,Waqis jadoon,Zia ur Rahman DOI 0.1109/ACCESS.2019.2947484
4. Pratt H, Coenen F, Broadbent DM, Harding SP, Zheng Y. Convolutional neural networks for diabetic retinopathy. Procedia Comput Sci 2016;90:200–5.
5. Harper CA, Keeffe JE. Diabetic retinopathy management guidelines. Expet Rev Ophthalmol 2012;7(5):417–39.
6. E. T. D. R. S. R. GROUP. Grading diabetic retinopa thy from stereoscopic color fundus photographs- an extension of the modified Airlie House classification. Ophthalmology 1991;98(5):786–806.
7. Scanlon PH, Wilkinson CP, Aldington SJ, Matthews DR. A Practical manual of diabetic retinopathy management. first ed. Wiley-Blackwell; 2009.
8. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: International Conference on learning representations; 2015.
9. Shanthi T, Sabeenian RS. Modified Alexnet architecture for classification of diabetic retinopathy images. Comput Electr Eng 2019;76:56–64.
10. Decenciere E, et al. Feedback on a publicly distributed image database: the messidor database. Image Anal Stereol 2014;33(3):231–4
11. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM 2017;60(6).
12. Abramoff MD, et al. Improved automated detection of diabetic retinopathy on a publicly available dataset through integration of deep learning. Investig Ophthalmol Vis Sci 2016;57(13):5200–6.
13. Dutta S, Manideep BC, Basha SM, Caytiles RD, Iyengar NCSN. Classification of diabetic retinopathy images by using deep learning models. Int J Grid Distr Comput 2018;11(1):99–106.
14. Kaggle dataset [Online]. Available, https://kaggle.com/c/diabetic-retinopat hy-detection
15. https://www.kaggle.com/datasets/sachinkumar413/diabetic-retinopathy-dataset
16. https://www.kaggle.com/datasets/sovitrath/diabetic-retinopathy-224x224-2019-data
17. https://www.kaggle.com/datasets/sovitrath/diabetic-retinopathy-224x224-gaussian-filtered