

Screening of Diabetic Retinopathy and its stages using Deep Neural Networks on Retinal optical Images

Ankit Kumar¹, Piyush Goenka², Dr. Jharna Majumdar³

¹Student, Department of Information Science & Engineering, ankit.tikna92@gmail.com

²Student, Department of Electronics & Communication Engineering, goenkapiyush@yahoo.co.in

³Dean R&D, Prof. and Head Dept. of M.Tech Computer Science & Engineering, Head Center for Space Research, Head Center for Robotics Research, jharna.majumdar@gmail.com

Nitte Meenakshi Institute of Technology, Bangalore – 560 064, India

ABSTRACT

Vision loss is also caused by Diabetic Retinopathy, being a consequence of Diabetes from which a vast portion of the world's population is suffering. To automate the diagnosis of Diabetic Retinopathy we have presented two-stage Detection and grading system using Deep Neural Networks. For our analysis, we have made use of the publicly available dataset. On performing hyper-parametric variations in the architecture, we have been able to reach an accuracy of 97.46% to classify if it is a Diabetic Retinal Image or not. We have obtained an accuracy of 73% for classifying the different stages of Diabetic Retinopathy.

Key words: AlexNet, Custom neural networks, Diabetic Retinopathy, GoogLeNet, hyper parameters, ResNet

1. INTRODUCTION

The changes in lifestyles of working-age adults have also led to a rise in diabetic patients. Among more than 422 million Diabetic individuals Diabetic Retinopathy (DR) is prevalent in the United States, India and China [1]. Of these patients, more than 30% are expected to have DR [2]. DR is the leading cause of vision impairment [3] that can progress to blindness.

This disease can be prevented by Blood Glucose control and timely treatment. Furthermore, due to the complexities of DR [4] and many diabetic patients, manual diagnosis is a time-consuming process [5]. Adding with to technological advances, there have been attempts at the automation of the DR detection [2][5][6][7][8].

As we know, DR being a diabetic complication characterized by Damaging of blood vessels in the eye. The image features from the retinal image responsible for detection are 1. Microaneurysms (MA): These are the tiny blood vessel protrusions in the back of the eye. They do not affect vision and are generally reversible with treatment of high blood pressure, diabetes or other disorder causing them. 2. Haemorrhages (HE): It is the leakage, of blood into the areas in and around vitreous humor of the eye. Most common

causes are MA-related macular degeneration. They can be Intraretinal HE, vitreous HE, Sub-hyaloid HE. Vision is unaffected in most cases. 3. Exudates (EX): These are intraretinal deposits largely made up of extracellular lipids leaked from abnormal retinal capillaries and can vary from specks to larger patches. They are distinct yellow-white and evolve into rings known as circinate. These are of two types: hard exudates, soft exudates. 4. Cotton Wool Spots (CWS): These are nerve fibre layer having indistinct (fluffy) edges which often become greyish-white due to discoloration. They are the result of local ischemia resulting in axoplasmic flow disruption. Figure 1 shows symptoms in sample images of Retina.

These relevant features of DR, which are observably recognized based on seeing the Image of Retina Fundus. Moreover, Deep Learning (DL) techniques have proven to be excellent for operating with images hence, we have applied the DL for screening and prediction of severity of DR based on Patient's Retinal fundus Imagery.

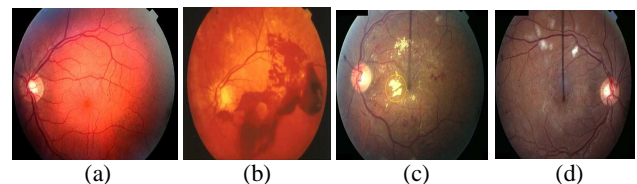


Figure 1: (a) Microaneurysms, (b) haemorrhages, (c) Exudates, (d) Cotton wool spots

Our work aims to build a screening neural network model and compare the efficiencies of Convolutional Neural Network (CNN) as described in the Methodology section using the datasets as stated in the Data composition section. We have also introduced a custom NN architecture based DR grading system which is capable of image classification. It is based on Disease pathologies from its four severity levels explained in the Dataset composition section. And then an analysis is discussed in the results section followed by conclusions.

2. LITERATURE SURVEY

An early statistical attempt was made by Bernhard M. Ege, developing a screening tool providing routine monitoring using a Bayesian, a KNN and Mahala Nobis classifiers [6] focusing on 4 major abnormalities: Hemorrhages (HR), Cotton wool spots (CWS) [4], Exudates (EX) and Microaneurysms (MA). Using machine learning technologies and making Neural Network (NN) model has given a jump to efficiency and intricacy of detection by generating mass dataset by augmentation and preprocessing to precisely predict classification This was used by Shao Hua in [5] for DR image classification. Suvajit Dutta trained the model with back-propagation NN and CNN by first filtering by the median filter and made use of edge detection for processing images digitally then feeding to NN [5].

Reference [2] showed that using contrast limited adaptive histogram equalization (CLAHE) improved recognition, pre-trained networks such as GoogLeNet, AlexNet [5] models been accurate to 74.5% on binary classification models [2] and shown that training converged much faster for transfer learning.

Varun Gulshan, [1] observed that to achieve high sensitivity and specificity, lesion-based features are not necessarily to be specified. Frank D. Verbraak, high diagnostic accuracy was achieved with the use of hybrid lesion based device with deep learning enhancements, which is beneficial at primary care sites in areas rich in diabetic patients and will help screen the patients efficiently [9].

3. DATASET COMPOSITION

We have used two kinds of Datasets viz. **A.** Numerical Dataset (PIMA Indians Dataset [10]): Input: Numerical data (insulin, glucose levels and different details), Target Classes are 2 *i.e.* diabetes or not (0/1-YES/NO); **B.** Pictorial Dataset (i). Asia Pacific Tele-Ophthalmology Society (APTOS) 2019 Dataset [11] Input: Retinal Image data (Unaugmented Data), Target Classes are 5 (Diabetic stages)

A. the Numerical Dataset provides the information on the diagnosis of Diabetes which can be especially useful in early prevention of DR as DR is a complication of Diabetes characterized by Damaging of blood vessels in the eye. This is the main reason why; we can use Retinal fundoscopic images to detect and grade the DR. 7 features are helping in the prediction of diabetes *i.e.* Age , Blood pressure, BMI, Diabetes pedigree function, Insulin levels, Glucose, Skin thickness. Based on which Diabetes Classification is done into two classes as shown in Table 1.

B. the Pictorial Dataset provides information on disease severity of DR. this dataset contains 3662 images for training and 1928 images for testing. The dataset has been divided into 5 classes as shown in Table 2. The relationship between these numerical data and image-based data is shown by the Venn diagram in Figure 2. As seen clearly, only a subset of

people who have diabetes (referred as target class 1 *i.e.* blue color) turns out to have DR (Target class 1 *i.e.* Red color) which is further divided into four-stages (*i.e.* disease severity). Also, it clearly shows our methodology described in Figure 3 followed is appropriate. The Datasets contained Data of patients from varying ethnicity, age groups etc. this makes it perfect for the analysis and detection of DR and early prevention.

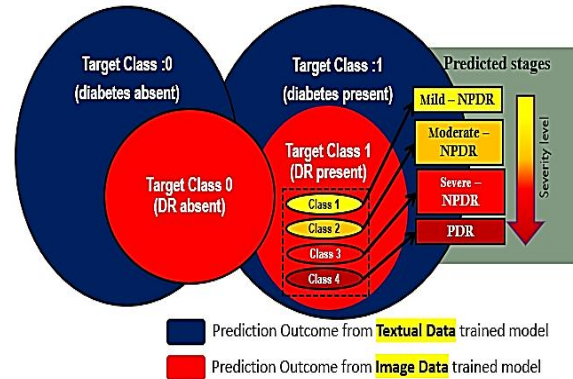


Figure 2: Relationship between outcomes based on Numerical Dataset and Image Dataset

Table 1: Class distribution in Numerical Dataset

Class label	Description	Number of instances
0	Diabetes Absent	500
1	Diabetes Present	268

Table 2: Class distribution in Pictorial Dataset

Class label	Disease Severity Level	Signs	Number of images in class
0	0 Grade: DR Not apparent	No visible sign of abnormality	1805
1	1 st Grade: NPDR - Mild	Presence of MA only	370
2	2 nd Grade: NPDR - Moderate	Less than severe NPDR but More than just MA	999
3	3 rd Grade: NPDR - Severe	More than 20 HR Intraretinal microvascular abnormalities Venous beading No signs of PDR	193
4	4 th Grade: PDR	Vitreous/pre-retinal HR s-Neovascularization	295

4. METHODOLOGY

After studying previous literature in the domain of DR applications and related image classification tasks, we have used the three types of Convolutional Neural Network (CNN) i.e. AlexNet [12], GoogLeNet [13], ResNet [14], and three NN of custom architectures.

PREPROCESSING: It is a crucial step before we start any operation using dataset, as the accuracy and efficiency of the system created, depends on the kind of data we are sending as input to the system. There are many tasks in preprocessing the data depending on the datasets. Some of the entries are termed as noises because they create unnecessary variation which is not related to classification, hence are required to be operated on or removed. As we have two different kinds of Datasets.

For the Numerical dataset very first step observing the distribution of the frequencies over various attributes/factors e.g. glucose. Insulin levels etc. over the dataset. As we observed that there were various entries with BMI, insulin level, Skin thickness as '0', which is practically not possible and can be considered as missing information or noise in the dataset as for normal human skin fold thickness cannot be less than 10 mm. Hence for cleaning the data we have removed the entries where the Blood Pressure, BMI, and Glucose levels are 0. and then operated on the dataset.

THE WORKFLOW: The relationship between these numerical data and image-based data is shown by the Venn diagram in Figure 2 depicts that early screening of Diabetes before Screening of DR is required i.e. Numerical data type based screening for Diabetes, after this image data based DR screening and severity grading. Hence, we came up with two approaches as shown by the flowchart in Figure 3.

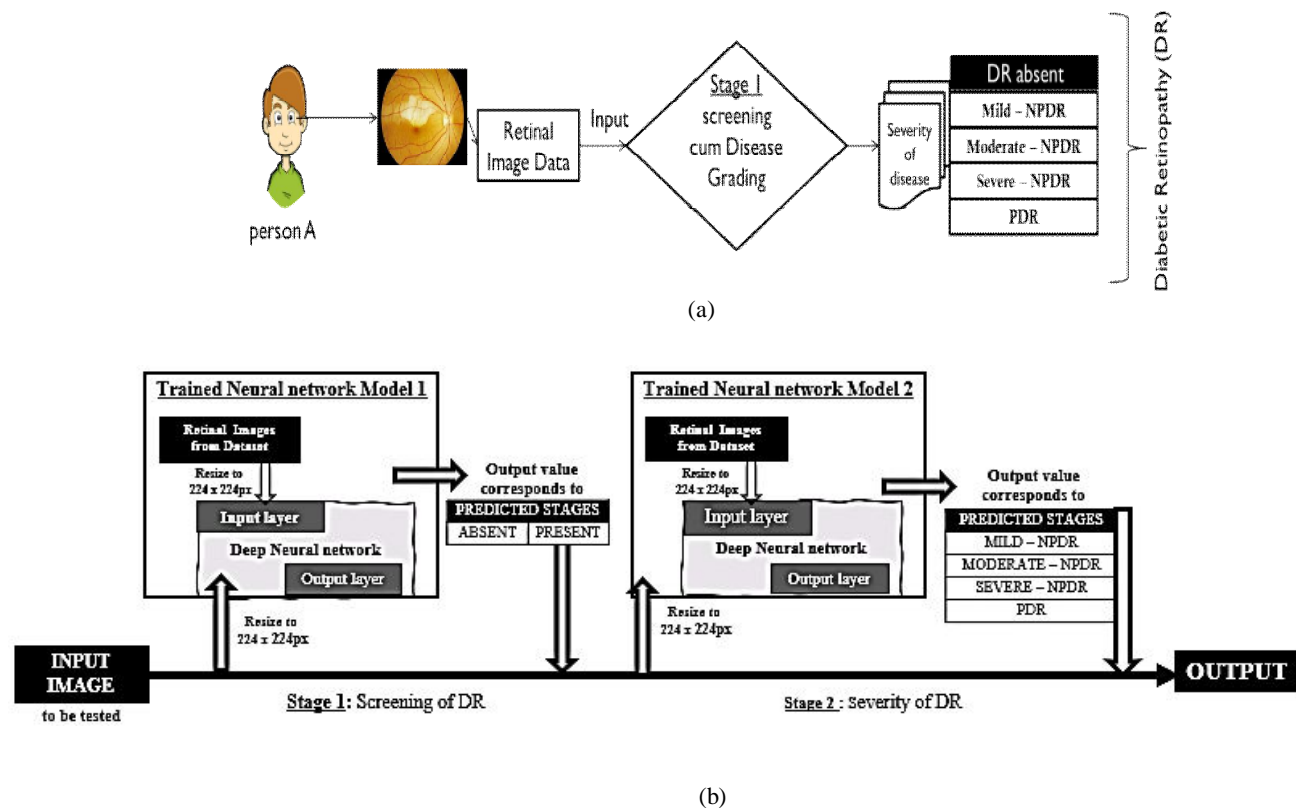


Figure 3: Flowchart showing two approaches of solving the problem: (a) single-stage Broad 5 categories classification i.e. detecting the non-DR patients and DR severity altogether; (b) two-stage model i.e. screening of DR patients and grading severity at later stage elaborated dataflow.

DEEP LEARNING: The use of Neural Networks (NN) have seen an increase over time due to their computational efficiency and accuracy. We have also used Convolutional Neural Networks (CNN) and also custom architectures with a large number of layers between input and output layers in a NN responsible for making it Deep Neural Network (DNN). We have made variations in Hyper-parameters. These parameters are responsible for determining the structure of a NN and are changed to control the behaviour of NN before training. The Hyper-Parameters are: **1.** The number of epochs is the number of times of availability of the whole training data to NN while it is training. One must increase it until a proper fit is achieved i.e. while training accuracy is increasing, the validation accuracy decreases. **2.** The batch size determines how much data should be given at a time to network. And in one epoch all batches are provided hence, whole training Dataset is shown for one epoch. **3.** The number of layers determines the structure and depth of NN. increasing their units combined with regularization techniques often increase accuracy. **4.** Regularization technique is responsible for avoiding overfitting and provide generalization. We have used Dropout Regularization which cancels out random neurons. **5.** Kernel size is the dimension of space to be operated on per operation. **6.** Activation functions (basic mathematical functions used to introduce nonlinearity to models) are some examples.

We have used 5 activation functions in our research and compared the efficiencies. the activation functions are :

A. *SIGMOID* function : a logistic function narrows the input to a range (0,1). The formula is given by

$$sigmoid(x) = \sigma = \frac{1}{1+e^{-x}} \tag{1}$$

B. *TANH* function: non-linear function, input is scaled down in range (-1,1), and given by the formula

$$tan\Box(x) = \frac{2}{1+e^{-2x}} - 1 \tag{2}$$

C. *ReLU* function: stands for the rectified linear unit. It is given by the formula

$$ReLU(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \tag{3}$$

D. *ELU* function: stands for an exponential linear unit. It is given by the formula

$$ELU(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x-1) & \text{if } x \leq 0 \end{cases} \tag{4}$$

E. *SELU* function: stands for a scaled exponential linear unit, one of the newer version of ELU. It is given by the formula

$$SELU(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x-1) & \text{if } x \leq 0 \end{cases} \tag{5}$$

Where,

$$\alpha \approx 1.673263242354377284817042991671$$

$$\lambda \approx 1.0507009873554804934193349852946$$

Diabetes Classification: Using Numerical Dataset, prediction of the Presence of Diabetes in a patient is performed. The dataset had a total of 7 features as described in the previous section based on which the images are classified for the presence and absence of Diabetes. We tested Logistic Regressing (LR) and a custom DNN. The architecture of DNN using 4 Dense layers is shown in Figure 4. The results of both these methods is shown in Table 4 in the results section.

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 12)	108
dense_2 (Dense)	(None, 8)	104
dense_3 (Dense)	(None, 8)	72
dense_4 (Dense)	(None, 1)	9
Total params: 293		
Trainable params: 293		
Non-trainable params: 0		

Figure 4: Architecture of 4 Layer DNN used with Numerical Data

DR classification: After studying the previous literature of the domain of DR applications and related image classification tasks, we have used the three types CNN i.e. AlexNet [12], GoogLeNet [13], ResNet [14] with variations in hyper-parameters and three CNN of custom architectures as stated in following sub-sections. We decided to come up with a general screening technique as shown in Figure 3(a). The CNN was trained on 3662 images after pre-processing them. The experiment is mentioned in the following sub-sections:

A. Using AlexNet [12]: AlexNet architecture has 5 Convolutional (CONV) Layers and 3 Fully connected (FC) Layers, ReLU as the activation function, Multiple CONV Kernels, filters extract features in an image. The first CONV Layer of AlexNet has kernel size 11x11x3 and 96 filters. An overlapping, Max Pooling layer is followed after the first two CONV layers. Then directly connected third, fourth and fifth CONV layers. Again, a Maxpooling Layer after the fifth CONV layer, the output of which goes into a series of two FC layers. The SoftMax classifier with 1000 class labels is fed by the second FC layer for output. The architecture of AlexNet is shown in Appendix A, Figure A.1 (a).

Hyperparameter variations in AlexNet: The variations in activation function hyper-parameter are done in stock

AlexNet architecture. The variant architecture is shown in Appendix A, Figure A.1 (b). mentioning 'Activation functions', these are changed different activation functions and assigned version name as new variant architecture shown in Table 3. The NN model is trained for all variants and the result of the accuracies is included in Table 8.

Table 3: Architecture mapping of activation functions and version names

Activation function	network version name
ReLU	Version 1
SELU	Version 2
ELU	Version 3
TANH	Version 4
SIGMOID	Version 5

B. Using GoogLeNet [13]: GoogLeNet has 22 layers, and 4 million parameters almost 12x fewer parameters than AlexNet (hence, faster than AlexNet and much more accurate). It makes use of 'inception' module(s) which is a combination of multiple CONV layers (i.e., 5×5 CONV layer, 1×1 CONV layer, 3×3 CONV layer) with a single output vector, of concatenated output filter banks of the inception module, forming the input for next layer. Hence, Time spent on choosing 2-3 versions and operating with them is reduced by combining those versions and training. The result will automatically be reflecting the best one of those variations. The architecture of GoogLeNet is shown in Appendix A, Figure A.1 (c) and uses ReLU as the activation function. **Hyperparameter variations in GoogLeNet:** The variations in the activation function hyperparameter is similarly done in stock GoogLeNet architecture to AlexNet to maintain uniformity in the experiment. The NN model is trained for all variants and the result of the accuracies is included in Table 9.

C. Using ResNet [14]: ResNet has 50 layers. It makes use of skip/shortcut connections i.e. Output of some of the previous layers (feature weights) are re-added/combined to feature weights of next hopping layers, hence creating a 'residual' block. Use of 'residual' modules ensures higher layer will be at least accurate as of the lower preceding layers and not the worse. ReLU is used as the activation function in stock network architecture. The first CONV Layer of ResNet contains 64 filters of kernel size 7x7x3. An overlapping, Max Pooling layer is followed after the first two CONV layers. Then the architecture is divided into four stages. The architecture of ResNet is shown in Appendix A, Figure A.1 (d). We have used ResNet for DR severity grading in the second stage of our approach. After training the NN model, the result of the accuracies is included in Table 13.

D. using Custom CNN architectures for classification: We have built 3 neural networks of custom architectures with details as mentioned in the following subsections:

D.1. NN version 1: The architecture showed in Appendix A, Figure A.2 (a). This NN consists of 7 CONV Layers and 4 FC Layers. Kernel size in the first two layers is 7x7 with 96 and 256 filters, respectively. An overlapping, Max Pooling layer is followed after the first 5 CONV layers, 6th and 7th CONV layer. Then a series of 4 FC layers. The SoftMax classifier with 2 class labels is fed by the last FC layer for output. ReLU is used as an activation function all over the network. This NN was trained on the same dataset and the accuracy is included in Table 10.

D.2. NN version 2: The architecture showed in Appendix A, Figure A.2 (b). This NN consists of 23 CONV Layers and the first CONV layer consists of 3x3 kernel size and 196 filters. It includes the Inception modules similar as GoogLeNet. The last fully-connected layer feeds into a SoftMax classifier with 2 class labels. ReLU is used as an activation function after the CONV layers. This NN was trained on the same dataset and the accuracy is included in Table 10.

D.2. NN version 3: The architecture is showed in Appendix A, Figure A.2 (b). This NN consists of 23 CONV Layers and the first CONV layer consist of 2x2 kernel size and 128 filters. It includes the Inception modules similar to GoogLeNet. SoftMax classifier with 2 class labels is fed by the last FC layer for output. ReLU is used as an activation function after every CONV layer. This NN was trained on the same dataset and the accuracy is included in Table 10.

5. RESULTS

Diabetes Classification: Using Numerical Dataset, for predicting the Presence of Diabetes in a patient we tested, Logistic Regressing (LR) and a custom DNN using 4 dense layers. The confusion matrices of both these methods are shown in Table 4(a, b) and it can be observed that the custom DNN achieved the accuracy of 80.01% and proved to be more accurate than LR which achieved an accuracy of 77.01%.

Table 4: Confusion matrices of models tested on Numerical Data: (a) DNN with 4 Dense Layers; (b) Logistic Regression

(a)			(b)		
Actual Class\ Predicted Class	Class-0	Class-1	Actual Class\ Predicted Class	Class-0	Class-1
Class-0	475	25	Class-0	448	52
Class-1	154	114	Class-1	121	147

DR classification: Using Pictorial Dataset, for approach 1, the images were split in the Test, Train and Validation sets in ratio 10:70:20 respectively with a number of images as shown in Table 5.

Table 5: Distribution of Images per class for approach 1

Class label	Number of images in		
	Test set	Training set	Validation set
0	180	1264	361
1	37	259	74
2	99	700	200
3	20	135	38
4	29	207	59

7.1-Single-stage Broad 5 categories classification using GoogLeNet: Using GoogLeNet architecture for classification of images directly into 5 target classes, i.e. no DR and grading DR in its stages, the accuracies, and losses by the trained NN are shown in Table 6. Variations in activation functions were first made using the same NN architecture. These networks were trained for 40 epochs.

Table 6: Broad classification (all 5 categories together: screening cum severity grading) using GoogLeNet architecture and its variations

5 categories classification		Activation function	Number of epochs	Max accuracy achieved (%)	Loss achieved
Network architecture used	Changed network version name				
GoogLeNet	Version 1	ReLU	40	85.84	0.38
	Version 2	SELU	40	73.4	0.81
	Version 3	ELU	40	76.29	0.67
	Version 4	TANH	40	69.75	0.87
	Version 5	SIGMOID	40	69.22	0.85

7.2-Two-stage model i.e. DR patients screening and grading severity at a later stage: After the results we observed that the accuracy is not optimum due to the no of unequal images per class. So, we changed our approach from as stated in Figure 3(a) to Figure 3(b) so that we will have an approximately equal number of images per class, (accuracy of NN increases if Classes are balanced in general). The total number of images used for training was 3662 and a split of 70:30 was done for training and validation set.

7.2.1 Stage 1: Binary classification: Using approach 2, the images were split in the Test, Train and Validation sets in 10:70:20 ratios respectively with number of images as shown in Table 7.

Table 7: Distribution of Images per class for approach 2 stage 1

Class label	Number of images in		
	Test set	Training set	Validation set
0	180	1264	361
1	185	1300	372

A. Using AlexNet: We had trained the NN with 5 activation functions for 50 epochs each on the same dataset, results are shown by Table 8. The architecture with ELU as activation functions found out to be most accurate with an accuracy of 97.46 %, SELU as an activation function was also more accurate (95.9%) than the activation function ReLU used in Stock Network architecture with 86.18% accuracy. The least accurate model was trained when TANH is used as an activation function (accuracy of 64.06%). the training behaviour graph of version 3 having ELU activation function is shown in Figure 5.

Table 8: Binary classification, stage 1 screening of Figure 3(b) using AlexNet architecture and its variations

Stage 1 using AlexNet		Activation function	Number of epochs	Max Accuracy achieved (%)	Loss achieved
Network architecture re used	Changed network version name				
AlexNet	Version 1	ReLU	50	86.18	0.3623
	Version 2	SELU	50	95.90	0.1078
	Version 3	ELU	50	97.46	0.0785
	Version 4	TANH	50	64.06	0.8770
	Version 5	SIGMOID	50	91.86	0.2315

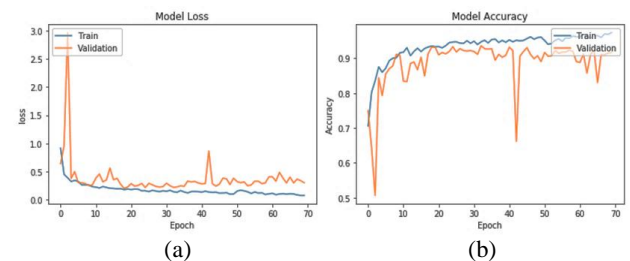


Figure 5: The training behavior graph of version 3 having ELU activation function showing (a) model loss and (b) model accuracy

B. Using GoogLeNet: Using GoogLeNet for the prediction on the same dataset with hyperparameter variation, the stock architecture with activation function ReLU trained out to be most accurate with an accuracy of 97.45 %. and the least accurate network with activation function as TANH with 87.19 % accuracy. All the NN were trained for 40 epochs. As shown by Table 9.

Table 9: Binary classification, stage 1 screening of Figure 3(b) using GoogLeNet architecture and its variations

Stage 1 using GoogLeNet		Activation function	Number of epochs	Max Accuracy achieved (%)	Loss achieved
Network architecture used	Changed network version name				
GoogLeNet	Version 1	ReLU	40	97.45	0.0835
	Version 2	SELU	40	88.99	0.2870
	Version 3	ELU	40	91.38	0.2382
	Version 4	TANH	40	87.19	0.3126
	Version 5	SIGMOID	40	91.03	0.2230

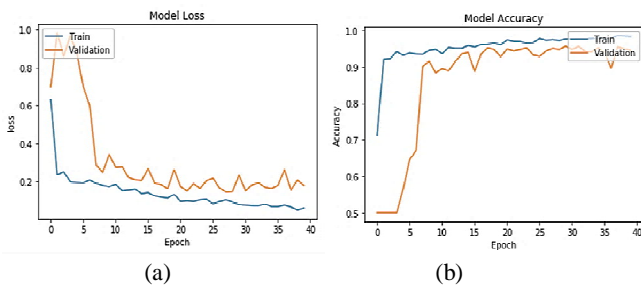


Figure 6: The training behavior graph of version 1 having ReLU activation function, (a) model loss and (b) model accuracy

C. Using Custom CNN architectures for classification

Using three custom NN architectures as mentioned in the methodology section for the prediction on the same dataset with hyperparameter variation, All the NN were trained for 40 epochs and results are shown in Table 10.

Table 10: Binary classification, NN of custom architectures

Binary stage 1 using NN of custom architectures		Kernel size	Activation function	Number of epochs	Max Accuracy achieved (%)	Loss achieved
Network architecture used	Changed network version name					
7 CONV layers, 4 dense layers	Version 1	7x7	ReLU	40	91.45	0.0835
23 CONV layers, 2 dense layers,	Version 2	3x3			80.38	0.524
	Version 3	2x2			80.18	0.529

7.2.2 Stage 2: Disease Severity Grading: Using approach 2, the images were split in the Test, Train and Validation sets in 20:50:30 ratio respectively with number of images as shown in Table 11.

Table 11: Distribution of Images per class for approach 2 stage 2

Class label	Number of images in		
	Test set	Training set	Validation set
0	74	185	111
1	200	499	300
2	38	98	57
3	59	147	89

D. Using ResNet for the prediction on the same dataset, the stock architecture with activation function ReLU trained NN for 40 epochs. The confusion matrix is shown in Table 12.

Table 12: Confusion matrix Prediction of ResNet architecture.

Number of images actual/predicted	Class 1 - Predicted	Class 2 - Predicted	Class 3 - Predicted	Class 4 - Predicted
Class 1 - Actual	52	9	14	2
Class 2 - Actual	23	143	10	25
Class 3 - Actual	3	1	28	2
Class 4 - Actual	7	4	7	41

Table 13: Class wise accuracy of the network ResNet

Network	Class	Accuracy (%)
ResNet	1	67.5
	2	71.1
	3	82.35
	4	69.5

6. CONCLUSION

After this experiment, we observe that for this given dataset the least suitable activation function is TANH and versions of ReLU (ReLU, SELU, ELU) differ in accuracies depending on architectures. SIGMOID activation function resulted in similar accuracy (~ 91%) in both architectures. From the experiments, we also conclude that Data augmentation using will increase the dataset but the distinct features per class will remain same (if unbalanced class), which results again in inaccurate classification and hence poor screening algorithm. Hence unaugmented dataset is used for classification. Based on the architecture of the neural network: GoogLeNet is performing with high accuracy than AlexNet on the unaugmented balanced dataset. We saw that ResNet is supposed to give more accuracy. But due to an insufficient number of dataset elements and imbalanced dataset, ResNet is not highly accurate as expected. We have successfully built 3 custom neural networks and tested their accuracies which were comparable

with 91% accuracy. The screening model with AlexNet version with changed hyper-parameter activation function as ELU performed 97.46% accuracy and GoogLeNet stock network performs with 97.45% accuracy.

The accuracy comparison will not be proper as we took the Image and Numerical Dataset from different sources. It will give proper Comparison if the Numerical Data and Pictorial Retinal Image Data is segregated and indexed. The collection of this kind of dataset will be extremely helpful for further research and will give an apt comparison of which one is better.

ACKNOWLEDGEMENT

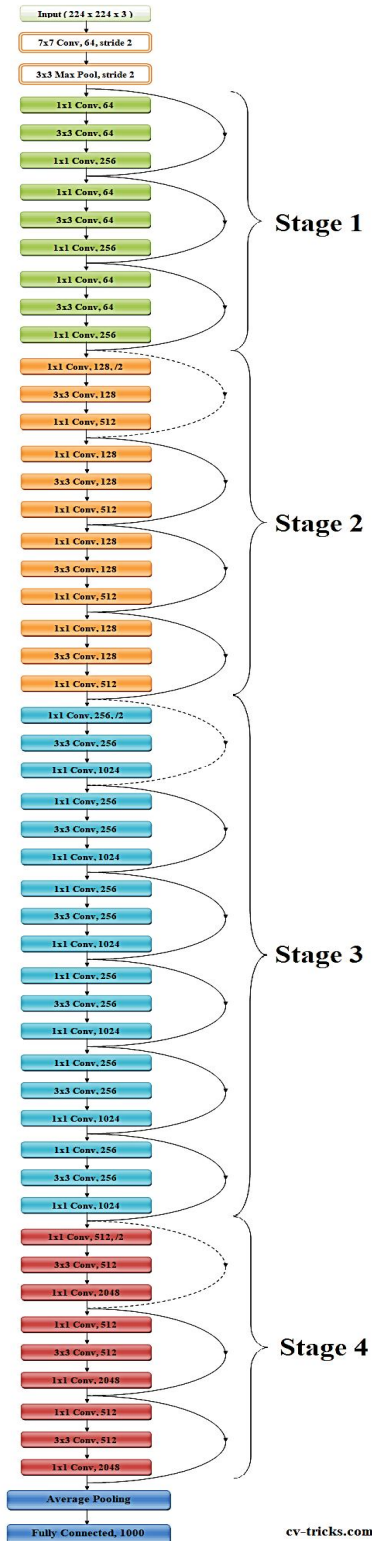
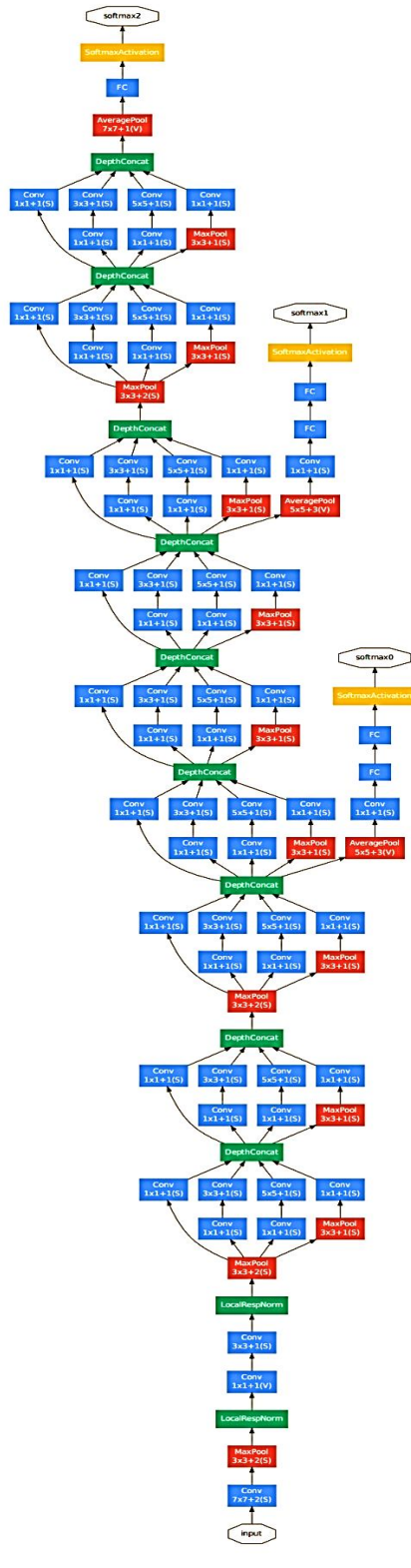
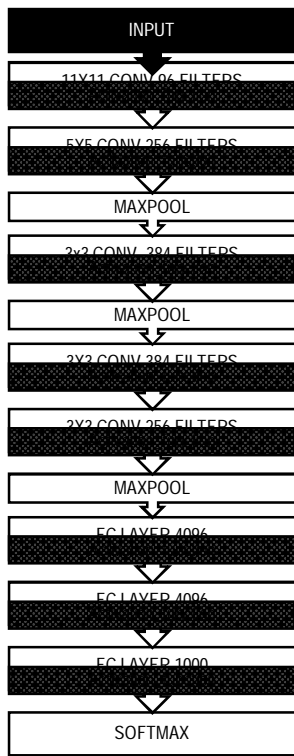
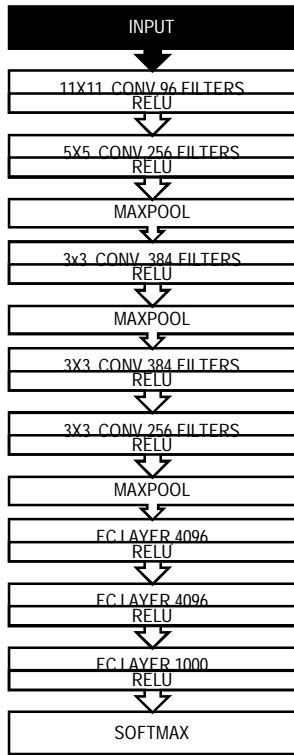
The authors express their sincere gratitude to Prof N.R Shetty, Advisor and Dr H.C Nagaraj, Principal, Nitte Meenakshi Institute of Technology for providing with constant encouragement and support to carry out research. The authors also extend their thanks and gratitude to the Vision Group on Science and Technology (VGST), Government of Karnataka, India to acknowledge their research and providing financial support to set up the infrastructure required to carry out the research.

REFERENCES

1. Gulshan, Varun, et al. "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs." *Jama* 316.22 (2016): 2402-2410.
2. Lam, Carson, et al. "Automated detection of diabetic retinopathy using deep learning." *AMIA summits on translational science proceedings* 2018 (2018): 147.
3. Porwal, Prasanna, et al. "Indian diabetic retinopathy image dataset (IDRiD): a database for diabetic retinopathy screening research." *Data* 3.3 (2018): 25.
4. Wilkinson, C. P. "Ferris F3, Klein RE, Lee PP, Agardh CD, Davis M, et al. Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales." *Ophthalmology* 110.9 (2003): 1677-82.

5. Shaohua Wan, **Deep convolutional neural networks for diabetic retinopathy detection by image classification**, www.elsevier.com/locate/compeleceng
6. Bernhard M. Ege, **Screening for diabetic retinopathy using computer-based image analysis and statistical classification**, *Computer Methods and Programs in Biomedicine* 62 (2000) 165–175.
7. Zhixi Li, **An Automated Grading System for Detection of Vision-Threatening Referable Diabetic Retinopathy based on Color Fundus Photographs**, <https://doi.org/10.2337/dc18-0147>
8. Suvajit Dutta, **Classification of Diabetic Retinopathy Images by Using Deep Learning Models**, *International Journal of Grid and Distributed Computing*, Vol. 11, No. 1 (2018), pp.89-106
<http://dx.doi.org/10.14257/ijgdc.2018.11.1.09>
9. Frank D. Verbraak, **Diagnostic Accuracy of a Device for the Automated Detection of Diabetic Retinopathy in a Primary Care Setting**, <https://doi.org/10.2337/dc18-0148>
10. PIMA Indians dataset, **PIMA Indians Diabetes Database**: <https://www.kaggle.com/uciml/pima-indians-diabetes-database>, *Kaggle*, [Dataset]
11. Asia Pacific Tele-Ophthalmology Society, "APTOS 2019 blindness detection," *Kaggle* <https://www.kaggle.com/c/aptos2019-blindness-detection/data>, 2019, [Dataset].
12. Krizhevsky, A., Sutskever, I., and Hinton, G. E. **ImageNet classification with deep convolutional neural networks**. In *NIPS*, pp. 1106–1114, 2012.
13. Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
14. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 770-778
DOI: 10.1109/CVPR.2016.90.

Appendix A



Stage 1

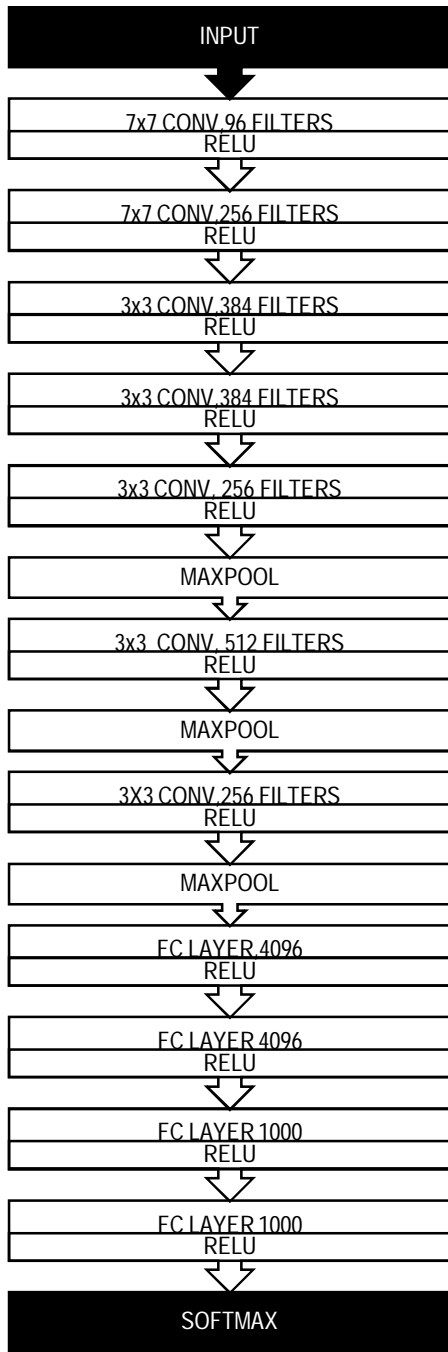
Stage 2

Stage 3

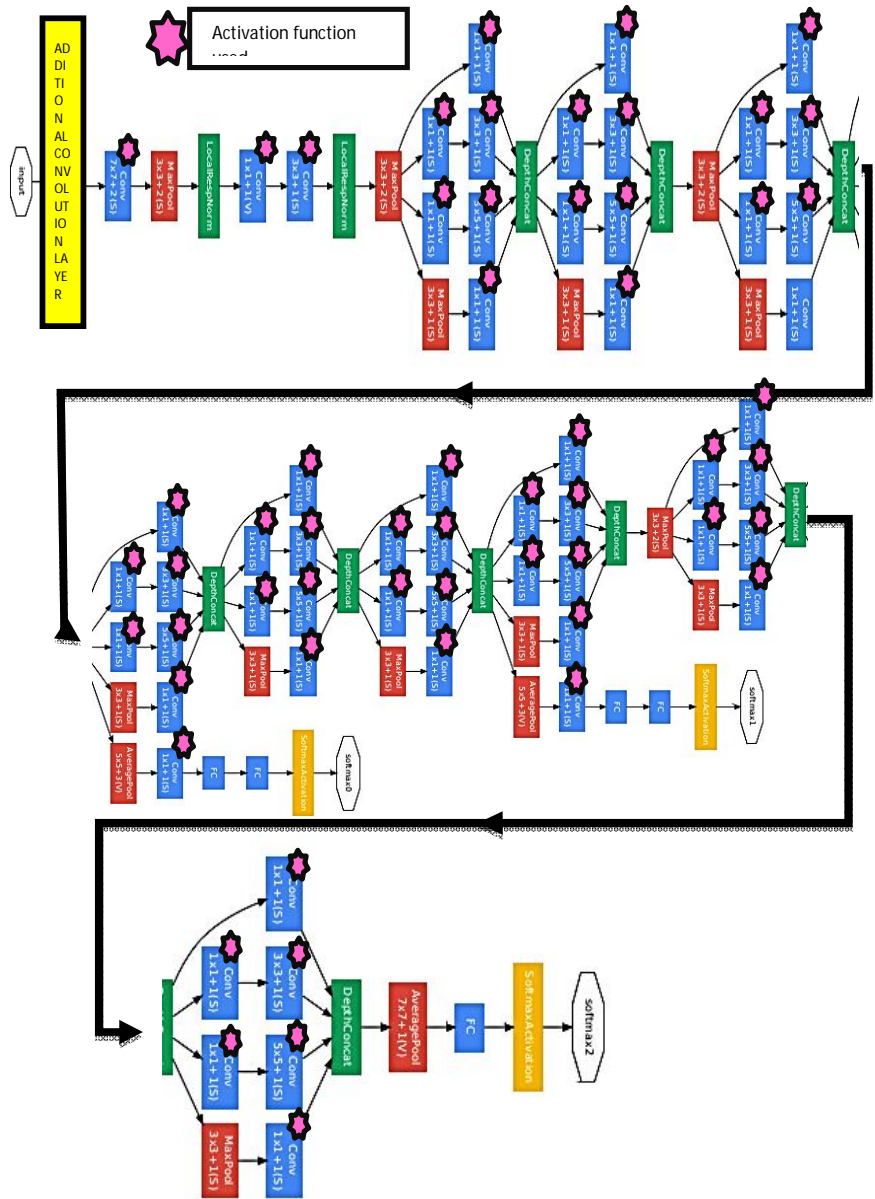
Stage 4

cv-tricks.com

Figure A.1: (a) stock architecture of AlexNet; (b) general architecture of version obtained when ‘activation functions’ are changed. ;(c) GoogLeNet Architecture; (d) ResNet50 Architecture



(a)



(b)

Figure A.2: custom architectures used (a) NN version 1; (b) NN version 2,3 (changes are made in ‘Additional convolutional’ layer’s kernel sizes , activation functions)