DEEP LEARNING-BASED SEVERITY PREDICTION FOR DIABETIC RETINOPATHY

Project report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION

ENGINEERING

By

SANKET SAXENA(171008)

SHIVAM SINHA(171019)

UNDER THE GUIDANCE OF

DR.SHRUTI JAIN

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

MAY 2021

TABLE OF CONTENTS

CAPTION	PAGE NO.
DECLARATION	iv
ACKNOWLEDGEMENT	v
LIST OF ACRONYMS AND ABBREVIATIONS	vi
LIST OF FIGURES	vii
LIST OF TABLES	ix
ABSTRACT	x
CHAPTER-1: INTRODUCTION	11
1.1 Why ConvNets over Feed-Forward Neural Nets?	12
1.2 Convolution Layer — The Kernel	13
1.3 Pooling Layer	14
1.4 Fully Connected Layer	15
1.5 Motivation	16
1.6 Objective	16
1.7 Outline of the Report	17
CHAPTER-2: LITERATURE REVIEW	18
CHAPTER-3: DATA AUGMENTATION AND ADAM OPTIMIZATION	19
3.1 Data augmentation	20
3.2 Adam optimization	21
CHAPTER-4: Different CNN Architecture	23
4.1 Inception V3	24
4.2 ResNet	26
4.3 VGG-16	30
4.4 Cross Entropy loss	33
CHAPTER 5 Results and Discussion	36

5.1 COST CURVE 37
5.2 Learning Rate vs Iteration 38
5.3 Output of Neural Network 40
5.4 VGG-16 Epoch vs Accuracy AND Epoch vs Loss 41
5.5 Inception V-3 Epoch vs Accuracy AND Epoch vs Loss 43
5.6 Res-Net 101 Training Accuracy and Training Loss 45

CHAPTER 6 CONCLUSION 49
REFERENCES 50
LIST OF PUBLICATIONS 53
PLAGIARISM REPORT 54

DECLARATION

We hereby declare that the work reported in the B.Tech Project Report entitled **"DEEP LEARNING-BASED SEVERITY PREDICTION FOR DIABETIC RETINOPATHY"** submitted at Jaypee University of Information Technology, Waknaghat, India is an authentic record of our work carried out under the supervision of "Dr.Shruti Jain". We have not submitted this work elsewhere for any other degree or diploma.

Sanket Saxena

Shivam Sinha

SHIVAM SINHA SANKET SAXENA

171019 171008

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

(SHRUTI JAIN)

Dr. Shruti Jain

Associate Professor

Jaypee University of Information Technology

Waknaghat, Distt. Solan, Himachal Pradesh.

Date:

Head of the Department/Project Coordinator

ACKNOWLEDGEMENT

The achievement and ultimate result of this undertaking required a great deal of direction and help from numerous individuals and I am very favored to have this up and down the fruition of my task. All that I have done is simply because of such oversight and help and I would not neglect to say thanks to them.

We regard and say thanks to my mentor **Dr. Shruti Jain**, for giving me a chance to accomplish the project on the topic **"DEEP LEARNING-BASED SEVERITY PREDICTION FOR DIABETIC RETINOPATHY"** and giving us all help and direction which made me complete the undertaking appropriately. We are very appreciative to Dr Shruti jain for offering a particularly decent help and direction, in spite of the fact that she had occupied timetable dealing with the aministrative undertakings.

I'm grateful to and sufficiently lucky to get steady consolation, backing and direction from all Teaching staffs of Electronics Department which helped us in effectively finishing our task work. Additionally, I might want to stretch out our earnest regards to all staff members and especially **Mr Abhishek Ray** in lab for their convenient help.

SHIVAM SINHA (171019)

SANKET SAXENA(171008)

LIST OF ACRONYMS

1.	CNN	Convolutional Neural Network
2.	GPUs	Graphics processing units
3.	SVM	Supervised machine learning
4.	DNN	Deep Neural Network
5.	RAM	Random access memory
6.	RELU	Rectified Linear Unit
7.	DA	Data Augmentation
8.	NN	Neural Network
9.	DR	Diabetic Retinopathy

LIST OF FIGURES

1. Figure 1 A CNN sequence to classify handwritten digits.
2. Figure 2 4 x 4 x 3 RGB Image
3. Figure 3 Convoluting a 5x5x1 image with a 3x3x1 kernel
4. Figure 4 3 x 3 pooling over 5x5 convolved feature
5. Figure 5 Depicting the Fully Connected Layer
6. Figure 6 Methodology for detection of various stages of DR
7. Figure 7 Some of image transformations for data augmentation
8. Figure 8 Showing Adam Optimization
9. Figure 9 Computation of Adam optimization algorithm
10. Figure 10 Different CNN Architecture LeNet the basic architecture
11. Figure 11 INCEPTION-V3
12. Figure 12 Residual Block : a building block for ResNet
13. Figure 13 Another look at ResNet 34
14. Figure 14 Conv1-Convolution
15. Figure 15 Conv 1-Max Pooling
16. Figure 16 A visualization of the VGG architecture
17. Figure 17 Different VGG Configuration
18. Figure 18 Matrix of Image
19. Figure 19 Cost Curve
20. Figure 20 Learning Rate vs Iteration
21. Figure 21 Training Accuracy vs Epochs
22. Figure 22 Output of Neural Network

23. Figure 23 VGG-16 epoch vs accuracy

24. Figure 24 VGG-16 epoch vs loss

25. Figure 25 INCEPTION V3 epoch vs accuracy

26. Figure 26 INCEPTION V3 epoch vs loss

27. Figure 27 The training accuracy and loss and validation accuracy and loss

28. Figure 28 Training accuracy(RES-NET 101)

29. Figure 29 Training loss(RES-NET 101)

30. Figure 30 Recall vs Epoch

31. Figure 31 Precision of the model at every epoch

LIST OF TABLES

1. **Table 1: The various measures and parameter of various ResNet Architectures……………………………………………………………......27**

2. **Table 2: Table depicting the cost after every 100 iteration……………………………………………………………………….39**

3. **Table 3 : Comparison table depicting various features of both network and its outputs………………………………………………………………………….41**

4. **Table 4: Training loss and accuracy and Validation loss and accuracy for some intermediate epochs……………………………………………………………42**

5. **Table 5: It depicts the various network architecture and the loss and accuracy obtained after training the different images……………………………………48**

ABSTRACT

In the area of ophthalmology, we started exploring computer-aided diagnostic screening for a disease of the attention known as diabetic retinopathy. Diabetic retinopathy is the quickest developing motive of preventable blindness globally , with nearly 415 million diabetic patients at threat worldwide. The disease is generally recognized through a noticeably educated medical doctor through inspecting a retinal experiment of the attention.This disease may be averted if it is detected in its early stage, however if undetected, the disease progresses into irreversible blindness, and in plenty of the world, there really aren't enough docs to be had to guide the extent of screening required to protect the population.

We believe that Machine Learning can help doctors identify patients in need, particularly among underserved populations., we present different CNN architecture that can predict the diabetic retinopathy disease much accurately, potentially helping doctors screen more patients in settings with limited resources.The results show that our algorithm's performance are exciting , but there is still a lot of work to do.First, adding various technique to reduce bias in our algorithm can improve the performance of our algorithm. Also we will be using Convolutional Neural Network as they have the strong potential to improve our prediction.

As there have been lots of advancements in Machine Learning recently ,we hope we can use those different techniques and can come across a solution for our problem of medical imaging in healthcare broadly.

CHAPTER 1

INTRODUCTION

A CNN is progressively mind-boggling engineering construed more from the human visual perspective. A previous study done on DR suggests the use of CNN but with a different approach. Among other managed calculations involved, the proposed arrangement is to locate a superior and advanced way to classify the fundus picture with little pre-preparing techniques. Different fundus image databases available have been discussed.

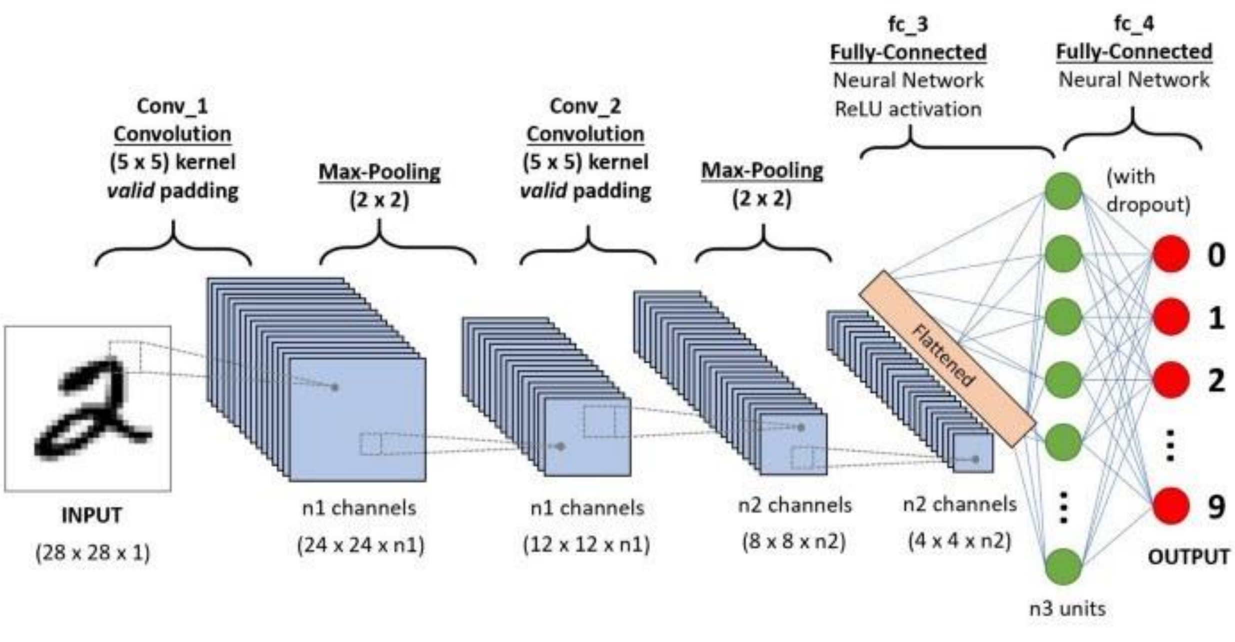


Fig 1: A CNN sequence to classify handwritten digits.

The main reason for choosing CNN was that they have taken inspiration from animal's visual cortex. The CNN is considered over NN because the fewer features are required in CNN which in turn helps in reducing overfitting problem in the proposed model.

1.1 Why ConvNets over Feed-Forward Neural Nets?

A ConvNet can viably get the Spatial and Temporal conditions in an image through the use of critical channels. The plan plays out a better fitting than the image dataset in view of the abatement in the amount of limits included and reusability of burdens. With everything taken into account, the association can be set up to appreciate the headway of the image better.

A Conv-Net designing is one of the least demanding case an overview of Layers that change the image volume into a yield volume (for instance holding the class scores).There are two or three specific kinds of Layers.

Every bit section recognizes a data 3D volume and changes it to a yield 3D volume through a differentiiable function. Each Layer could possibly have limits.

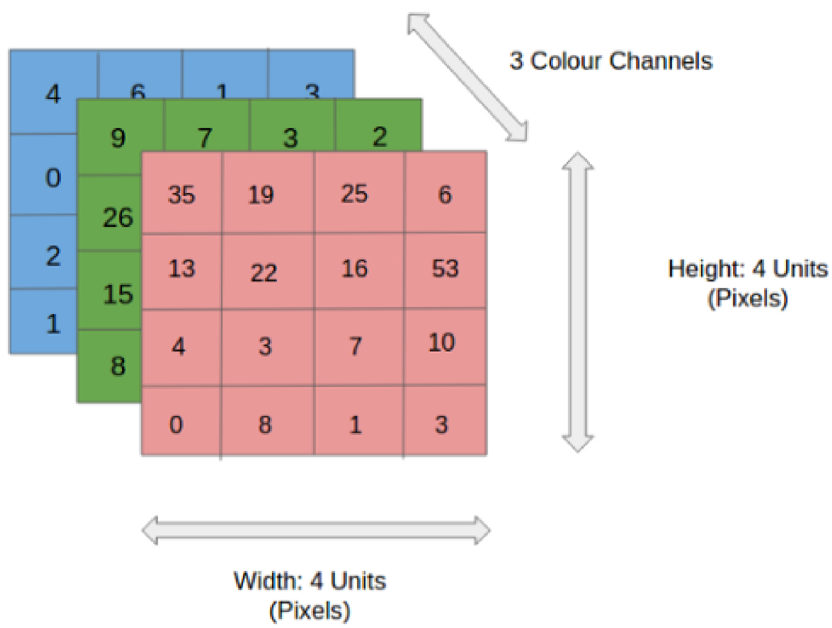


Fig 2: 4 x 4 x 3 RGB Image

The figure 2 is a RGB image which has been disconnected by the rest of its three concealing planes — Red, Green, and Blue. Various other concealing spaces in which pictures exist — Grayscale, RGB, HSV, CMYK, etc

We can imagine how computationally raised things would get once the photos show up at estimations, state 8K (7680×4320). The piece of the ConvNet is to reduce the photos into a structure which is less complex to gauge, without losing features which are essential for getting a respectable figure. This is huge when we are to design a plan which isn't only adequate at learning features yet moreover is versatile to gigantic datasets.[2]

The fundus images are attained by different cameras and by changing its field of views, angles, clarity, and ratio collected from different datasets. Data augmentation consists of different steps : flipping images, contrast adjustment, brightness adjustments are made.

1.2 Convolution Layer — The Kernel

The convolution is the main plane for extracting the highlights from an information image. Convolution stores the connection between pixels by learning the highlights of the image using small information squares. It is a numerical activity that requires two sources of information.

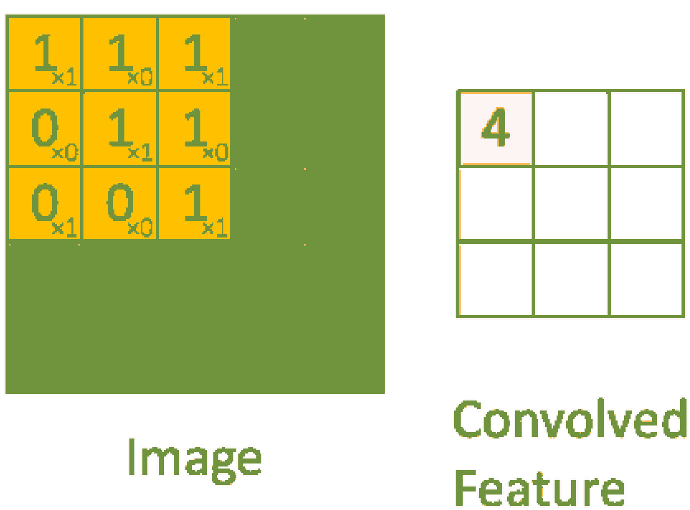


Fig 3: Convoluting a 5x5x1 image with a 3x3x1 kernel to get a 3x3x1 convolved feature

Image Description = 5 (Height) x 5 (Breadth) x 1 (Number of channels, eg. RGB)

In the above show, the green fragment takes after our 5x5x1 data picture. The part drew in with doing the convolution movement in the underlying section of a Convolutional Layer is known as the Kernel/Filter, K, addressed in the concealing yellow. We have picked K as a 3x3x1 lattice.[3]

Resizing is the primary step of the pre-processing. Before nourishing into the architecture for classification, the images are converted to grayscale and afterward to the L model. It is a monochrome image that is utilized to emphasize the MAs, and vessels in the fundus images and helps in flattening the images in a single dimension for further dealing

It provides us with two results to the movement — first in which it is merged or covolved incorporate is lessened in dimensionality when diverged from the data, and next one which dimensionality is extended or remains as in the past.

1.3 Pooling Layer

Like the Convolution layer, the grouping layer is forced to reduce the spatial size of the folded feature. To reduce the expected performance of managing information through dimension reduction. Additionally, it is useful for isolating dominant perspectives that are rotation and position invariants while maintaining the pattern of effective model preparation.

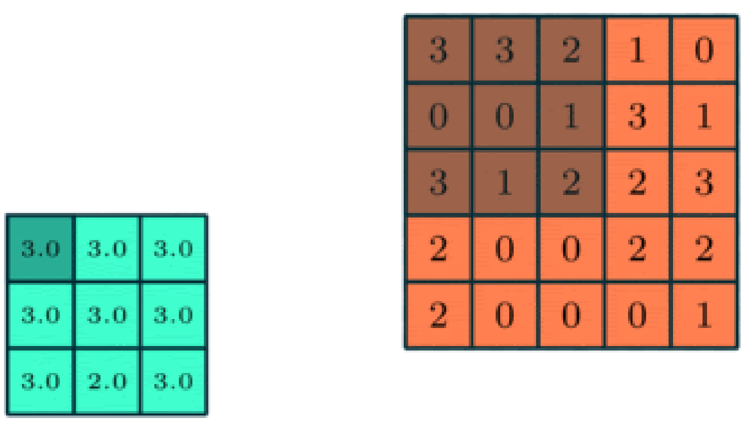


Fig 4:3x3 pooling over 5x5 convolved feature

There are generally two types of grouping: maximum grouping and average grouping. The maximum grouping restores the best motivator from the fragment of the image covered by the core. On the other hand, the average grouping restores the typical of the great general properties of each bit. of the image covered by the kernel. [4]

1.4 Fully Connected Layer (FC Layer)

Fully Connected Layer is used after the normal/ max-pooling layer. All neurons in the past layer from the max-pooling layer are taken by a completely associated layer and associated with each neuron [20]. After the stacked or profound different layers, the last layer which stacked toward the end for ordering the fundus picture is a softmax layer (Classification Layer).

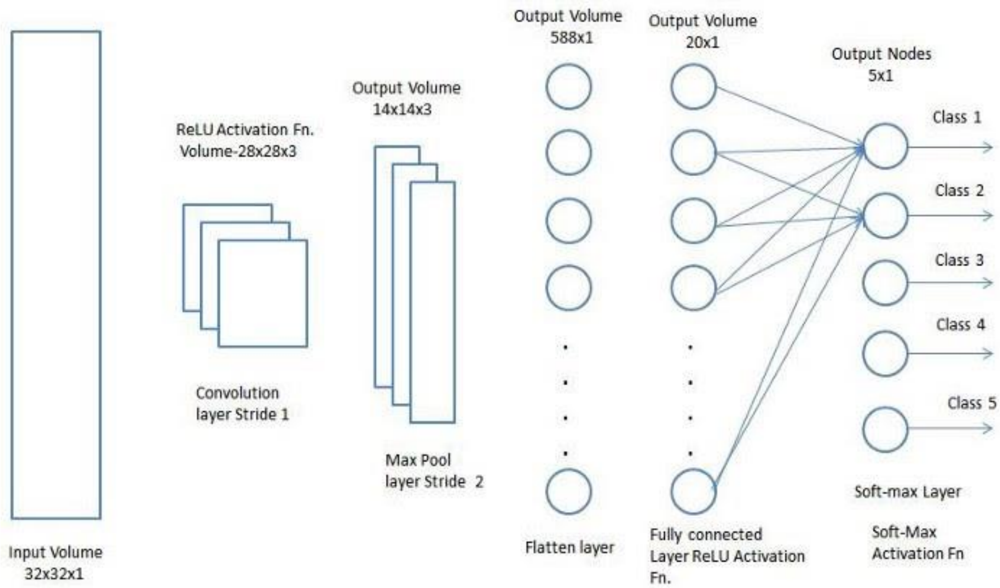


Fig 5: Depicting the Fully Connected Layer

A CNN is a kind of feed-forward counterfeit neural system wherein the network design between its neurons is propelled by the association of a creature's visual cortex. In profound learning, the convolutional neural system utilizes unpredictable engineering made out of stacked layers in which is especially very much adjusted to characterize the pictures. The convolutional layer includes the fraction of channels. Each channel is convolved and focus by framing another layer or initiation map. Every enactment map contains some critical highlights or trademark of the information image [15, 16] In the convolutional layer, $m \times m$ channel is tangled with the $N \times N$ input neuron layer that results in the size of $(N - m + 1) \times (N - m + 1)$. The pooling Layer is one of the most noteworthy layers that assist the system from avoiding overfitting by lessening the boundaries and calculation in the system. It's only a scale back to the pixels with highlights. For $N \times N$ input layer, will yield a layer of $N/K \times N/K$. ReLU Layer is an actuation work communicated by Eq 1.

$$F(x) = \max(0, x) \qquad (1)$$

1.5 MOTIVATION

As, we know that there are almost 80 million people in India who are suffering from sight lost or any other eyes diseases. So one of the major challenge is the prediction of the eye diseases. So by seeing the facts and figures and the before prediction problems we decide to build a program by which we can help the people and the doctors all over the world to successfully and accurately predict the retinal diseases.

1.6 Objective

i. To develop and implement a novel, reliable approach for medical retinal image using the deep learning and convolutional neural network approach in order to obtain better values.

ii. To compare various Neural Architecture on different images of the eye using the convolutional neural network.

iii. To detect and classify Diabetic retionpathy diseases by employing the convolutional neural network algorithms.

1.7 Outline of the Report

The study encompasses deep learning as well as convolutional neural network method in order to detect and classify the severity of the diabetic retinopathy of various medical images of the retina. The proposed framework aims at providing the various results of the trained images on different neural architectures and give the accuracy and losses obtained from it.

The research is organized as follows:

Chapter 2 includes the analyses of research work by different scholars, providing a better understanding of techniques based on convolutional neural network and deep learning.

Chapter 3 include the the deep learning techniques which are Adam optimization and Data Augmentation which helps in the prediction of the severity of the diabetic retinopathy disease.

Chapter 4 include the Convolutional Neural Network architecture. These architecture help in prediction of the accuracy and losses obtained after the training of the images.

Chapter 5 illustrate the graphs and the table obtained as a result after the images are trained. We also get an idea about the severity, losses and the accuracy of our model. Further the tables compares the different architecture and gives us an idea of the same as brief.

Chapter 6 provides the conclusion on the findings of this dissertation and proposed improvements, as well as potential study directions for further research.

CHAPTER 2

LITERATURE REVIEW

In the following period we have read the listed research papers regarding our project which are as following.

(Liao W.*et al*.,2019):The research paper propose a very novel accountable model i.e Convet which which not only used for the accurate diagnosis but rather also used for the various more transparent by also underlining the various areas further recognised by the network.The given model is made accountable by using a distinct EAMNet model not only for the correct glaucoma diagnosis but rather for various transparent interpretation in some regions.[11]

(Pratt H.*et al*.,2016):.The author propose a CNN way to deal with diagnosing DR from advanced fundus pictures and precisely arranging its severity.In this paper we built up an organization with CNN design and information expansion which can recognize the complicated highlights associated with the classification errand, for example, miniature aneurysms, exudate and hemorrhages on the retina and thusly give a conclusion naturally and without client input.[12]

(Shaban M.*et al.*,2020): All through the research paper, a profound Convolutional Neural Network (CNN) with 18 convolutional layers and 3 completely associated layers is proposed to break down fundus pictures and naturally recognize controls (for example no DR), moderate DR (for example a blend of mellow and moderate Non Proliferative DR (NPDR)) and serious DR (for example a gathering of extreme NPDR, and Proliferative DR (PDR)) with an approval precision of 88%-89%, an affectability of 87%-89%, a particularity of 94%-95%, and a Quadratic Weighted Kappa Score of 0.91–0.92 when both 5-overlay, and 10-overlap cross approval techniques were utilized individually.[13]

(Poplin R.*et al*.,2018): In this paper, profound learning engineering is proposed which can separate new information from retinal fundus pictures and can do Prediction of cardiovascular danger factors from retinal fundus.This paper shows that how profound taking in models prepared on information from 284,335 patients and approved on two free datasets of 12,026 and 999 patients, we anticipated cardiovascular danger factors not recently thought to be available in retinal pictures, for example, age (mean outright blunder inside 3.26 years), sexual orientation (territory under the collector working trademark bend (AUC) = 0.97), smoking status (AUC = 0.71), systolic pulse (mean supreme mistake inside 11.23 mmHg) and major unfavorable heart occasions (AUC =0.70).[14]

CHAPTER 3

DATA AUGMENTATION AND ADAM OPTIMIZATION

Adam is an ad libbed calculation which can be used then again the customary unexpected point fall framework to elate network loads reiteration arranged in creation information.
Information enlargement in information examination are strategies used to build the measure of information by adding somewhat changed duplicates of previously existing information or recently made manufactured information from existing information. It goes about as a regularizer and lessens overfitting when preparing an AI model.[5] It is firmly identified with oversampling in information investigation.

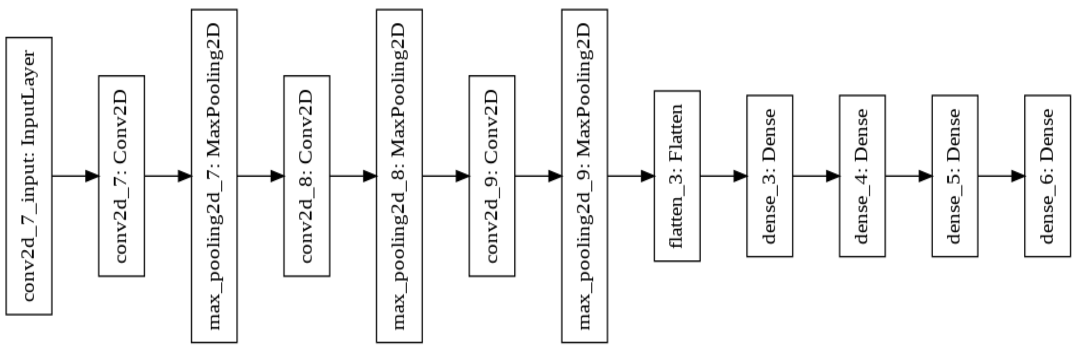


Fig 6: Methodology for detection of various stages of DR

3.1 DATA AUGMENTATION

The forecast exactness of the Supervised Deep Learning models is to a great extent dependent on the sum and the variety of information accessible during preparing. The connection between profound learning models and measure of preparing information required is comparable to that of the connection between rocket motors (profound learning models) and the immense measure of fuel (tremendous measures of information) needed for the rocket to finish its central goal (achievement of the profound learning model).[6]

DL models prepared to accomplish elite on complex assignments for the most part have an enormous number of concealed neurons. As the quantity of shrouded neurons expands, the quantity of teachable boundaries likewise increases.In basic terms, the measure of information required is corresponding to the quantity of learnable boundaries in the model. The quantity of boundaries is relative to the intricacy of the assignment.

Information increase can be utilized to address both the necessities, the variety of the preparation information, and the measure of information. Other than these two, expanded information can likewise be utilized to address the class lopsidedness issue in grouping undertakings. [7]

The enlargement strategies utilized in profound learning applications relies upon the kind of the information. To expand plain mathematical information, strategies, for example, SMOTE or

SMOTE NC are famous. These procedures are commonly used to address the class unevenness issue in order errands.

Figure 7: Some of common image transformations applied for data augmentation

3.2 ADAM OPTIMIZATION

Adam is an improvised computation which can be utilized alternately the traditional contingent angle plummet system to exhilarate network loads repetition situated in making data. It first came into picture or notice by Diederik Kingma from Open AI and Jimmy Ba from the University of Toronto by their 2015 ICLR paper (banner) named "Adam: A Method for Stochastic Optimization".

It’s a versatile learning rate advancement calculation that has been planned explicitly for preparing profound neural networks . The calculation uses the intensity of versatile learning rates strategies to discover singular learning rates for every boundary[8]. It additionally has focal points of Adagrad, which functions admirably in settings with scanty inclinations, however battles in the non-curved enhancement of neural organizations, and RMSprop, which handles to determine a portion of the issues of Adagrad and functions admirably in single settings.

Algorithm 1 : Adam Optimization

Require: α: Stepsize
Require: β1, β2 ∈ [0, 1): Exponential rot rates for the second gauges
Require: f(θ): Stochastic target work with boundaries θ
Require: θ0: Initial boundary vector
m0 ← 0 (Initialize first second vector)
v0 ← 0 (Initialize second vector)
t ← 0 (Initialize timestep)
while θt not united do
t ← t + 1
gt ← ∇ θft(θt−1) (Get inclinations w.r.t. stochastic target at timestep t)
mt ← β1 · mt−1 + (1 − β1) · gt (Update one-sided first second gauge)
vt ← β2 · vt−1 + (1 − β2) · g 2 t (Update one-sided second crude second gauge)
mb t ← mt/(1 − β t 1) (Compute inclination revised first second gauge)
vbt ← vt/(1 − β t 2) (Compute inclination adjusted second crude second gauge)
θt ← θt−1 − α · mb t/(√ vbt +) (Update boundaries)[7]
end while
return θt (Resulting boundaries)

The optimizer can be taken a gander at a mix of RMSprop and Stochastic Gradient Descent with energy. Further the optimizer utilizes the squared inclinations to calibrate the learning rate like RMSprop, also it exploits energy by utilizing moving normal of the angle rather than slope itself like SGD with momentum. It is an adaptable learning rate blueprint, which inferred, it records singular learning rates for various boundaries. The name came into known from versatile second assessment, and the explanation it's called is because it utilizes assessments of first and second snapshots of angle to maintain the learning rate for every weight of neural organization. Adam is not the same as an outmoded style of imaginary angle descent. Stochastic slope drop makes an outlying learning rate which is termed for all weight refreshes and the learning rate doesn't change during training [9]. A learning rate is saved up for every organization's weight (boundary) and independently adjusted as getting to know unfolds. The approach registers a person's flexible taking in quotes for numerous limitations from exams of first and 2d snapshots of the inclinations.

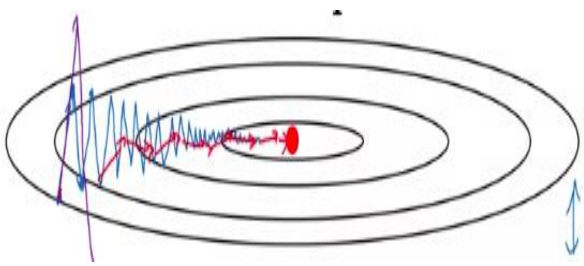


Fig 9: Computation of Adam optimization algorithm

Figure 9 shows the steps that are used for Adam algorithm in comparison to normal gradient descent. The blue line denotes the gradient descent steps while the red line denotes the steps taken by Adam Optimization [10]. It can be easily understood that Adam optimization is taking larger steps in the horizontal direction and taking very smaller steps in the vertical direction due to this it is much faster than gradient descent.

CHAPTER 4

Convolutional Neural Netwok Architecture

CNN structure is propelled with the aid of using the association and usability of the visible cortex and meant to emulate the community example of neurons within the human mind. The neurons inner a CNN are component into a third-dimensional design, with every set of neurons inspecting a touch district or spotlight of the picture.

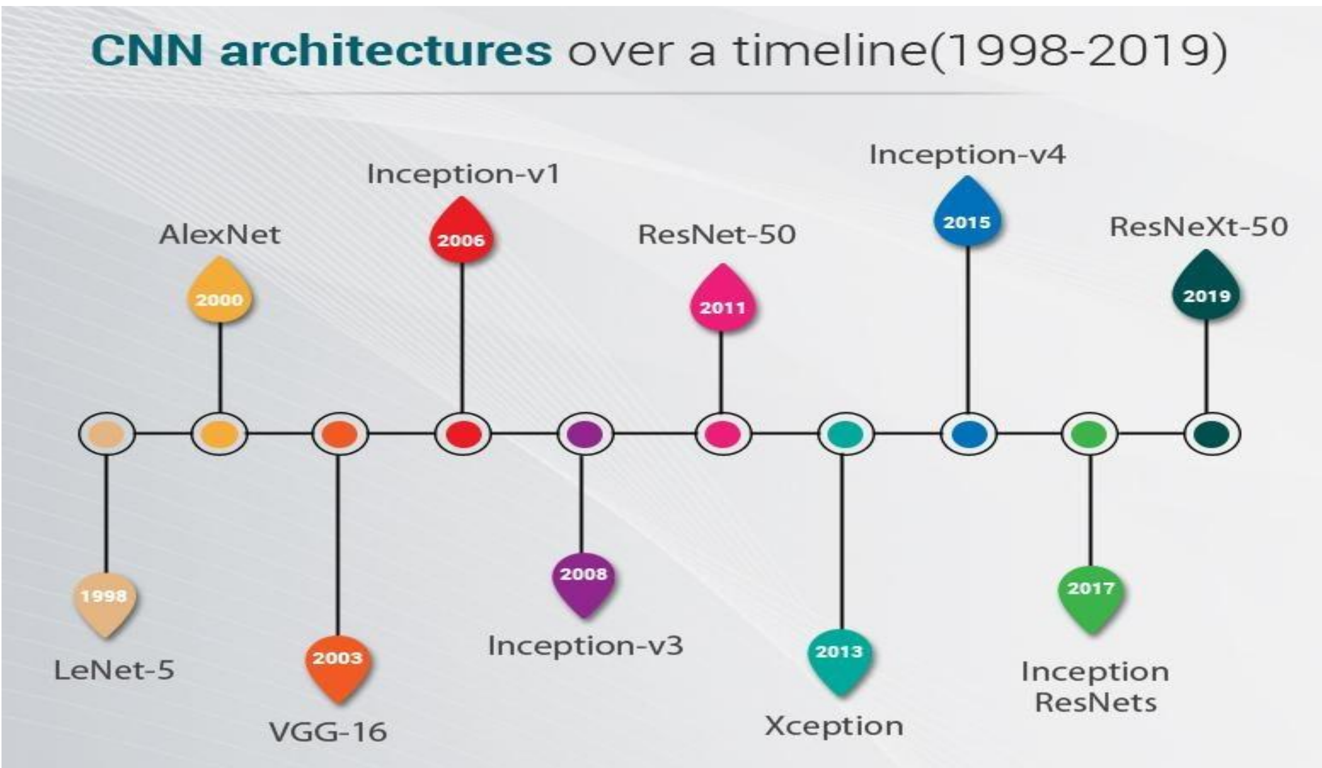


FIG 10: Different CNN Architecture LeNet being the basic architecture.

4.1 INCEPTION V3

Inception v3 is a convolutional neural agency engineering from the Inception own circle of relatives that makes a few upgrades consisting of using Label Smoothing, Factorized 7 x 7 convolutions, and the usage of an assistant classifer to unfold call data decrease down the agency (along the usage of clump standardization for layers withinside the sidehead).[17]

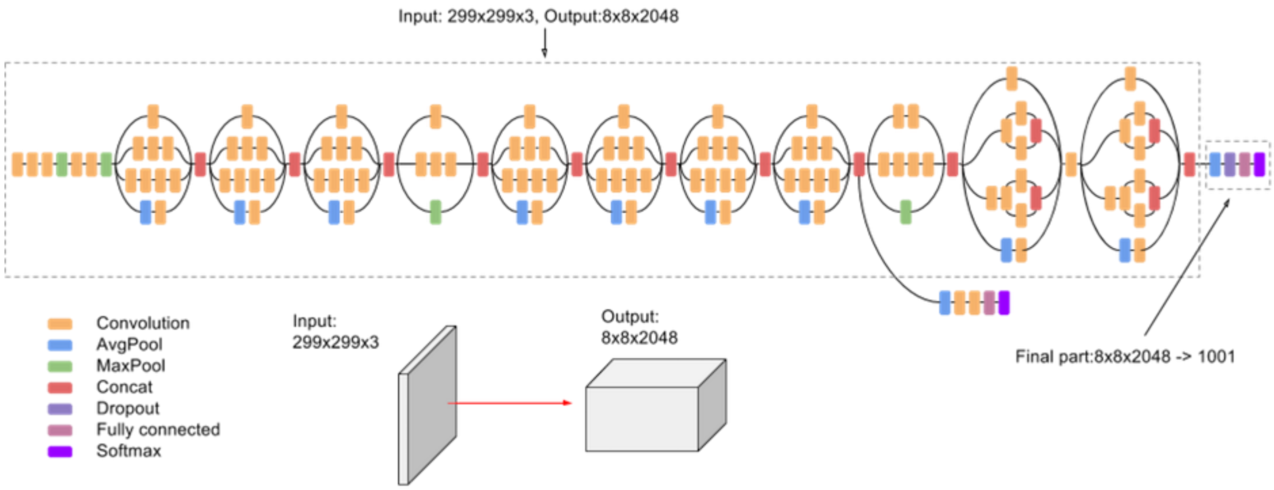


FIG 11: INCEPTION-V3

Conversely with VGGNet, Inception Networks (GoogLeNet/Inception v1) have wind up being even more computationally capable, both in regards to the amount of limits delivered by the association and the proficient cost achieved (memory and various resources). In case any movements are to be made to an Inception Network, care ought to be taken to guarantee that the computational advantages aren't lost. Thusly, the change of an Inception network for different use cases winds up being an issue due to the weakness of the new association's viability. In an Inception v3 model, a couple of methodologies for updating the association have been put proposed to remove the limits for less complex model variety. The techniques consolidate factorized convolutions, regularization, estimation decline, and parallelized computations.

4.1.2 Inception v3 Architecture

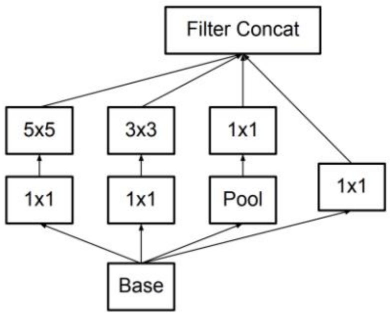
The model of an Inception v3 network is built, step-by-step, as explained below:

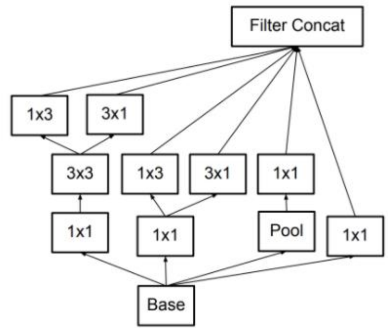
1. **Factorized Convolutions:** this assists with decreasing the computational effectiveness as it lessens the quantity of boundaries associated with an organization. It likewise keeps a mind the organization proficiency.

2. **Smaller convolutions:** supplanting extra convolutions with more modest convolutions easily activates faster preparing.

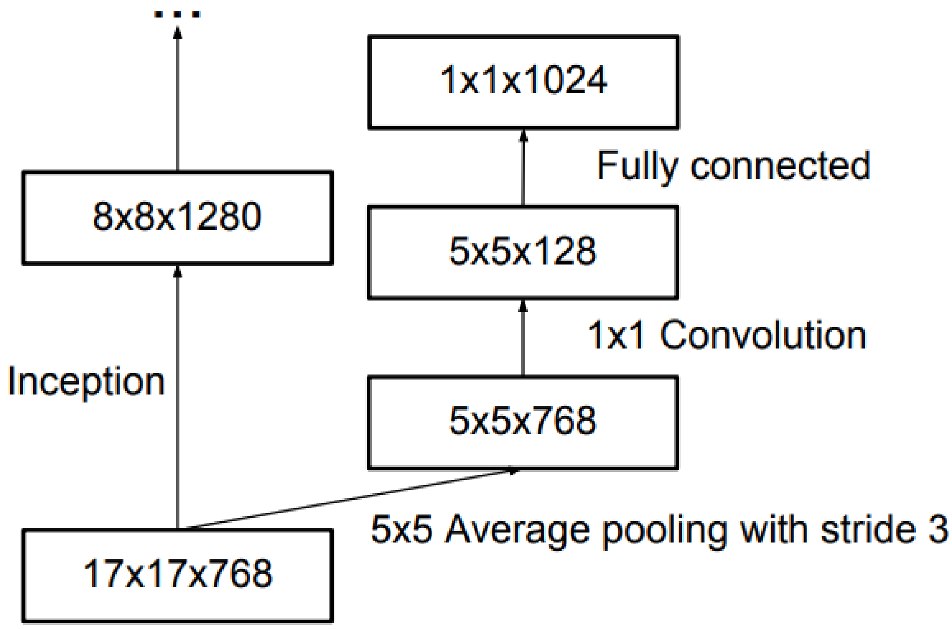
3. **Asymmetric convolutions:** A three × three convolution will be supplanted through a 1 × three convolution accompanied through a three × 1 convolution. In the occasion that a

three × three convolution is supplanted through a 2 × 2 convolution, the amount of boundaries might be particularly better than the deviated convolution proposed.

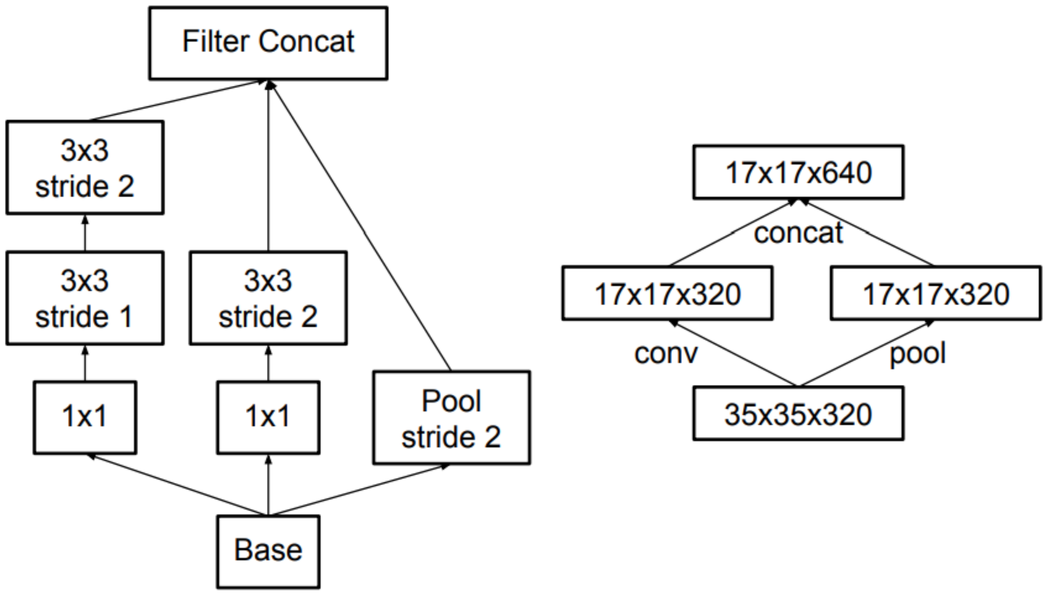




4. **Auxiliary classifier:** A Auxilary classifier is a touch CNN embedded among layers at some stage in preparing, and the misfortune added approximately is introduced to the essential employer misfortune. In GoogLeNet assistant classifiers had been applied for a greater profound employer, even as in Inception v3 a helper classifier is going approximately as a regularizer. [18]



5. **Grid size reduction:** Grid size decrease is generally done by pooling activities. Be that as it may, to battle the bottlenecks of computational expense, a more effective procedure is proposed:



4.2 ResNet

A resudal neural organization (ResNet) is a fake neural organization (ANN) of a sort that expands on develops known from pyramidal cells in the cerebral cortex. Lingering neural organizations do this by using skip associations, or alternate ways to bounce over certain layers. Commonplace ResNet models are carried out with twofold or triple-layer skirts that contain nonlinearities (ReLU) and cluster standardization in the middle. With regards to lingering neural organizations, a non-remaining organization might be portrayed as a plain organization.

Remaining Networks, or ResNets, learn lingering capacities regarding the layer contributions, rather than learning unreferenced capacities. Rather than trusting every couple of stacked layers straightforwardly fit an ideal fundamental planning, remaining nets let these layers fit a lingering planning. They stack remaining squares ontop of one another to frame organization: for example a ResNet-50 has fifty layers utilizing these squares.

ResNet utilizes a strategy called "lingering planning" to battle this issue. Rather than trusting that each couple of stacked layers straightforwardly fit an ideal hidden planning, the Residual Network expressly allows these layers to fit a lingering planning. The following is the structure square of a Residual organization.

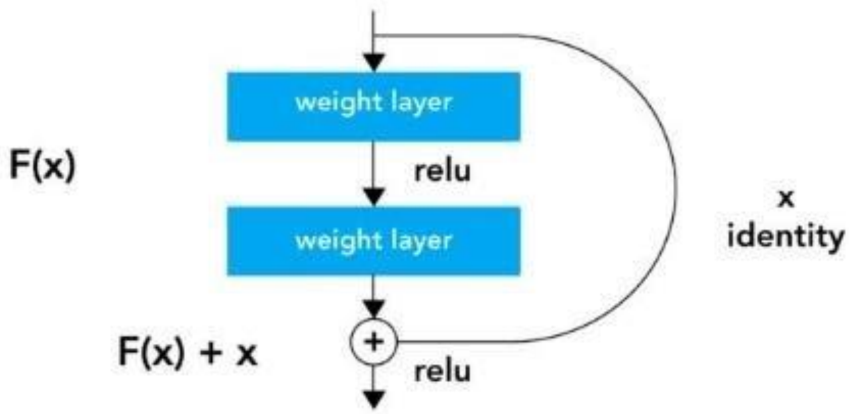


FIG 12. Residual Block : a building block for ResNet

1. ResNet Architecture

Contrasted with the regular neural organization designs, ResNets are generally straightforward. The following is image of a VGG organization, a simple 34-layer neural organization, also 34-layer lingering neural organization. In the plain organization, for the same that include map, the layers having a similar number of channels. On the off chance that the size of yield highlights is split the number of channels is crossed or multiplied, making the preparation interaction more perplexing.[19]

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 1: The various measures and parameter of various ResNet Architectures.

2. Different layers of ResNet

ResNet on the paper is for the most part clarified for ImageNet dataset. I like to see how really the volumes that are going through the model are changing their sizes. This way is more clear the system of a specific model, to have the option to change it to our specific necessities — we will perceive how changing the dataset powers to change the design of the whole model.
The Fig 13 represent another look at Conv.

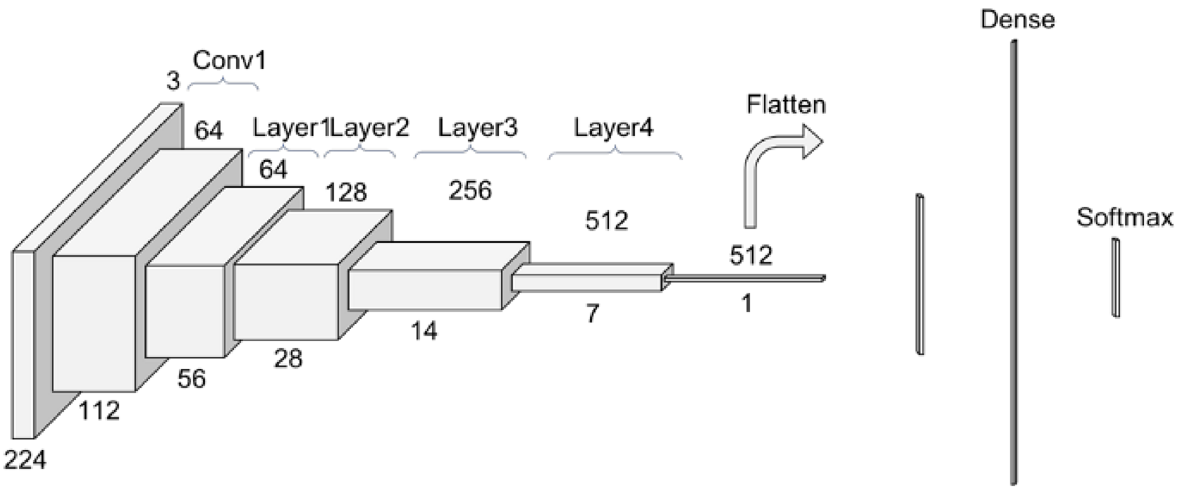


FIG 13: Another look at ResNet 34.

a. **Convolution 1** The initial step on the ResNet prior to entering the basic layer conduct is a square — called here Conv1 — comprising on a convolution + clump standardization + max pooling activity.

$$O = \frac{I + 2P - K}{S} + 1 = \frac{224 + 6 - 7}{2} + 1 = 112$$

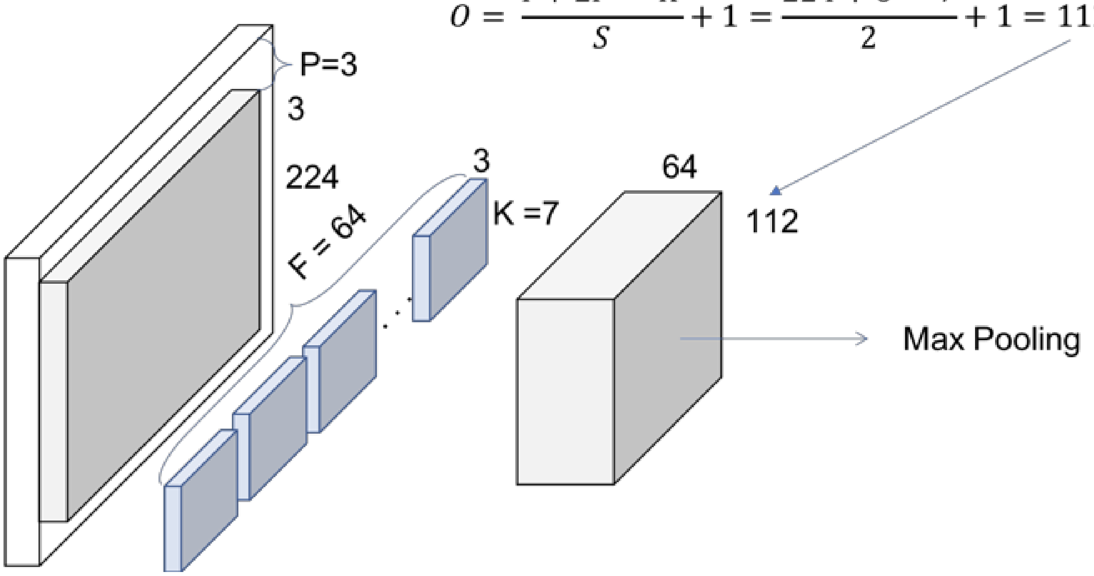


FIG 14: Conv1-Convolution

The subsequent stage is the cluster standardization, which is a component shrewd activity and hence, it doesn't change the size of our volume. At last, we have the (3x3) Max Pooling activity with a step of 2. We can likewise surmise that they first cushion the information volume, so the last volume has the ideal measurements.[20]

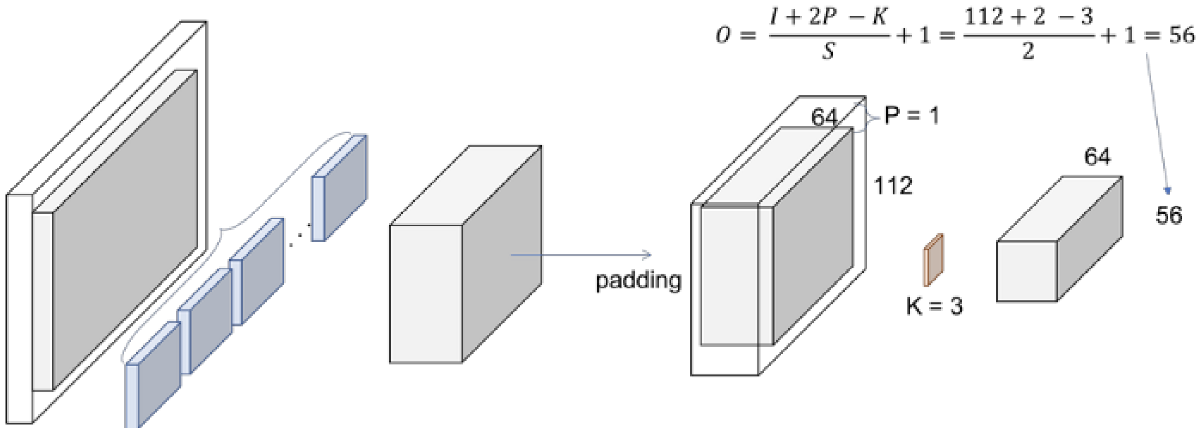


FIG 15: Conv 1-Max Pooling

b. ResNet Layers

Each layer of a ResNet is made out of a few squares. This is on the grounds that when ResNets go further, they ordinarily do it by expanding the quantity of tasks inside a square, yet the quantity of all out layers stays as before — 4. An activity here alludes to a convolution a clump standardization and a ReLU actuation to a contribution, aside from the last activity of a square, that doesn't have the ReLU.

In this manner, in the PyTorch execution they recognize the squares that incorporates 2 tasks — Basic Block — and the squares that incorporate 3 activities — Bottleneck Block. Note that typically every one of these tasks is called layer, however we are utilizing layer as of now for a gathering of squares.

5.2.3 How ResNet helps

The skip associations in ResNet tackle the issue of evaporating angle in profound neural organizations by permitting this other alternate route way for the slope to move through. The alternate way that these associations help is by permitting the model to get familiar with the character capacities which guarantees that the higher layer will perform in any event as great as the lower layer, and not more regrettable. It has been seen that lingering blocks make it astoundingly simple for layers to learn character.

4.3 VGG-16

VGGNet is a Convolutional Neural Network engineering. The complete name of VGG is the Visual Geometry Group.The unique reason for VGG's exploration on the profundity of convolutional networks is to see what the profundity of convolutional networks means for the exactness and precision of enormous scope picture arrangement and acknowledgment. - Deep-16 CNN), to develop the quantity of organization layers and to stay away from such a large number of boundaries, a little 3x3 convolution portion is utilized in all layers.[21]

4.3.1 **Architecture**

So contribution to VGG based convNet is a 224*224 RGB picture. Preprocessing layer takes the RGB picture with pixel esteems in the scope of 0–255 and deducts the mean picture esteems which is determined absurd ImageNet preparing set.

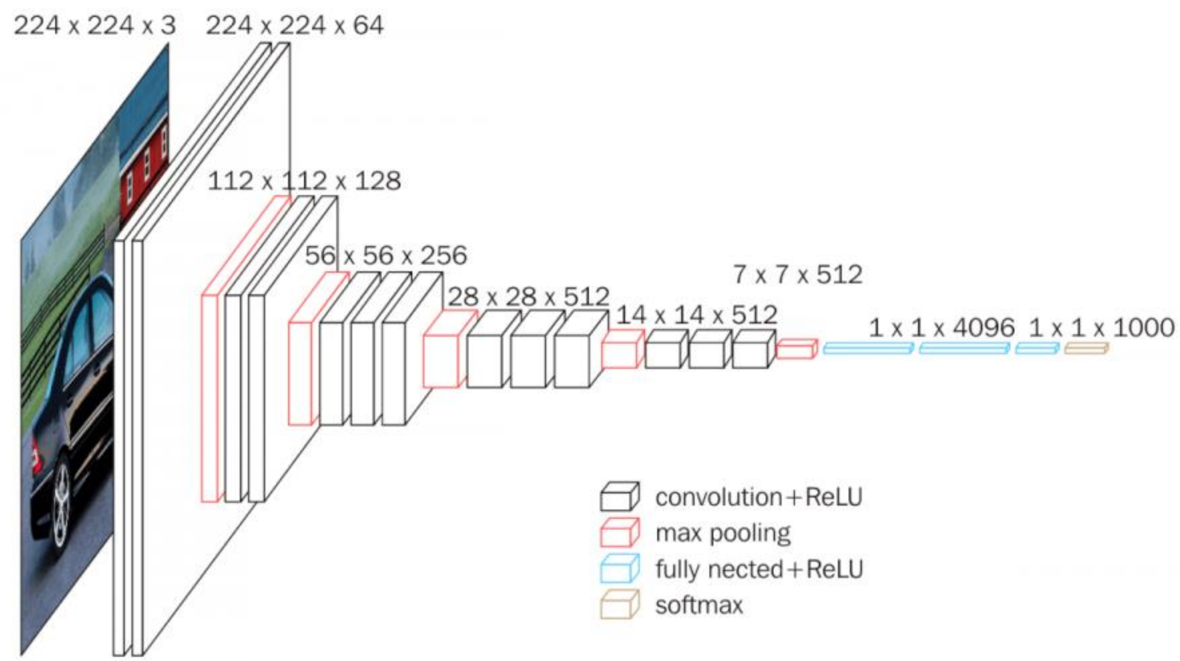


FIG 16: A visualization of the VGG architecture

The information pictures subsequent to preprocessing are gone through these weight layers. The preparation pictures are gone through a pile of convolution layers. There are absolute of 13 convolutional layers and 3 completely associated layers in VGG16 design. VGG has more modest channels (3*3) with more profundity as opposed to having huge channels. It has wound up having a similar compelling open field as though you just have one 7 x 7 convolutional layers.

Another variety of VGGNet has 19 weight layers comprising of 16 convolutional layers with 3 completely associated layers and same 5 pooling layers. In both variety of VGGNet there comprises of two Fully Connected layers with 4096 channels every which is trailed by another completely associated layer with 1000 channels to foresee 1000 names. Last completely associated layer utilizes softmax layer for characterization reason.

Architecture walkthrough:

- The initial two layers are convolutional layers with 3*3 channels, and initial two layers utilize 64 channels that outcomes in 224*224*64 volume as same convolutions are utilized. The channels are consistently 3*3 with step of 1
- After this, pooling layer was utilized with max-pool of 2*2 size and step 2 which decreases stature and width of a volume from 224*224*64 to 112*112*64.
- This is trailed by 2 more convolution layers with 128 channels. This outcomes in the new component of 112*112*128.
- After pooling layer is utilized, volume is diminished to 56*56*128.
- Two more convolution layers are added with 256 channels each followed by down inspecting layer that diminishes the size to 28*28*256.
- Two more stack each with 3 convolution layer is isolated by a maximum pool layer.
- After the last pooling layer, 7*7*512 volume is smoothed into Fully Connected (FC) layer with 4096 channels and softmax yield of 1000 classes.

4.3.2 **Configuration:**

The table underneath recorded diverse VGG design. Here ew can observe the 2 forms of VGG-16 (C and D). There isn't a lot of distinction between them with the exception of one that aside from some convolution layer there is (3, 3) channel size convolution is utilized rather than (1, 1). These two contains 134 million and 138 million boundaries individually.[22]

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

FIG 17: Different VGG Configuration

This picture is certainly utilized while presenting VGG16. This image contains a great deal of data. My understanding here might be restricted. On the off chance that you have any enhancements, kindly leave a message.

•Number 1 : This is an examination graph of 6 organizations. From A to E, the organization is getting further. A few layers have been added to check the impact.

•Number 2 : Each section clarifies the construction of each organization in detail.

•Number 3: This is a right method to do tests, that is, utilize the least difficult technique to take care of the issue , and afterward continuously upgrade for the issues that happen.

Network A: First notice a shallow organization, this organization can undoubtedly meet on ImageNet. And afterward?

Network A-LRN: Add something that another person (AlexNet) has tested to say is compelling (LRN), however it appears to be pointless. And afterward?

Network B: Then take a stab at adding 2 layers? Is by all accounts compelling. And afterward?

Network C: Add two additional layers of 1 convolution, and it will meet. The impact is by all accounts better. Somewhat energized. And afterward?

Network D: Change the 1 convolution part to 3 * 3. Attempt it. The impact has improved once more. Is by all accounts the best (2014).

4.3.3 Challenges Of VGG 16:

- It is very slow to train (the original VGG model was trained on Nvidia Titan GPU for 2-3 weeks).
- The size of VGG-16 trained imageNet weights is *528* MB. So, it takes quite a lot of disk space and bandwidth that makes it inefficient.

4.4 CROSS ENTROPY LOSS

When chipping away at a Machine Learning or a Deep Learning Problem, misfortune/cost capacities are utilized to enhance the model during preparing. The goal is quite often to limit the misfortune work. The lower the misfortune the better the model. Cross-Entropy misfortune is a most significant expense work. It is utilized to improve arrangement models.[23]

Cross-entropy is a proportion of the contrast between two likelihood dispersions for a given irregular variable or set of events.The cross entropy equation takes in two appropriations, p(x), the genuine conveyance, and q(x), the assessed dissemination, characterized over the discrete variable xx and is given by.

$$\mathbf{H(p,q)=-\sum_{\forall x} p(x)\log(q(x))}$$

Entropy is the quantity of pieces needed to communicate a haphazardly chose occasion from a likelihood appropriation. A slanted conveyance has a low entropy, though an appropriation where occasions have equivalent likelihood has a bigger entropy.

Entropy H(x) can be determined for an arbitrary variable with a gaggle of x in X discrete states discrete states and their probability P(x) as follows:

- **H(X) = – general x in X P(x) * log(P(x))**

Cross-entropy expands upon the possibility of entropy from data hypothesis and computes the quantity of pieces needed to address or send a normal occasion starting with one appropriation thought about then onto the next conveyance. [24]

The Cross-Entropy Loss is really the lone misfortune we are examining here. Different misfortunes names written in the title are different names or varieties of it. The CE Loss is characterized as:

$$CE = -\sum_{i}^{C} t_i log(s_i)$$

Where ti and si are the ground-truth and the CNN rating for every class ii in CC. As typically an initiation work (Sigmoid/Softmax) is implemented to the rankings earlier than the CE Loss calculation, we compose f(si)f(si) to allude to the actuations. In a double characterization issue, where C′=2C′=2, the Cross Entropy Loss can be characterized additionally as [discussion]:

$$CE = -\sum_{i=1}^{C'=2} t_i log(s_i) = -t_1 log(s_1) - (1 - t_1) log(1 - s_1)$$

- Calculated Loss and Multinomial Logistic Loss are different names for Cross-Entropy misfortune.

- The layers of Caffe, Pytorch and Tensorflow than utilize a Cross-Entropy misfortune without an inserted actuation work are:

- Caffe: Multinomial Logistic Loss Layer. Is restricted to multi-class grouping (doesn't uphold various marks).

- Pytorch: BCE Loss. Is restricted to double grouping (between two classes).

CHAPTER 5

Experimental Results and Discussions

Different fundus image databases are publically available for study purposes. Researchers have made some databases available through a hospital or Ophthalmologist. DIARETDB0 is a Standard Diabetic Retinopathy database. It has 130 images out of which 110 have signs of DR and 20 are normal. The resolution of images is 1500 × 1152 and Field of View (FOV) is 50 °. DIARETDB1 consists of 89 images with 5 normal images and 81 with signs of DR. It is obtained with ground truth collected from experts following an evaluation protocol. The DRIVE database has 40 images.

For the experimentation, data is collected from Kaggle software or manually written digit acknowledgment, for example, the MNIST dataset. CNN multi-layer profound engineering is actualized utilizing Theano and Lasagne libraries. Straightforward datasets are dealt with the equipment Intel i5 @3.20GHz, 8GB RAM Ubuntu 14.04. For dealing with an enormous Kaggle dataset, a Graphics Processing Unit is required. Amazon EC2 web administration occurrence is utilized. In this paper, the deep neural network technique is designed to classify the DR disease in different grades (0, 1, 2,3 4).

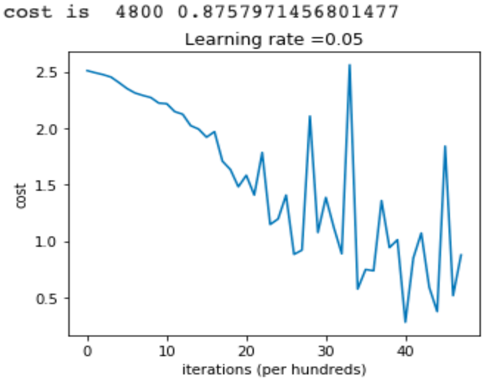
Initially, data is augmented using the *Image editor tool* which also helps in color balance adjustment, rotation, color adjustment, etc.. The Data Frame File is created which includes all the information of images like patient number level of DR and images names. Images were of dimensions 224 × 224 × 3, which were resized to 50 × 50 × 3 dimensions using MATLAB 2018b software. The images were converted into the matrix which is of unit 8 data type, authors have converted into the double type or vector of dimension 7500 as shown in Fig 10.

```
>> img
img =

ans(:,:,1) =

 Columns 1 through 16:

     0     0     0     0     0     0     0     3     1     0     0     3     2   140   164   177
     0     0     0     0     0     0     0     1     1     0     0     6   100   153   174   182
     0     0     0     0     0     0     0     1     0     0     4    38   162   177   184   188
     0     0     0     0     0     0     0     0     0     3     0   152   174   181   181   186
     0     0     0     0     0     0     1     0     3     1   151   161   175   172   186   190
     0     0     0     0     0     0     1     2     1     0   155   167   181   186   183   187
     0     0     0     0     0     0     1     8    10   154   169   177   184   184   188   191
     0     0     0     0     1     0     0     0    15   151   169   179   183   189   191   195
     0     0     0     0     1     0     0     4   166   158   172   181   186   190   193   195
     0     0     0     0     0     0     4     0   156   169   177   180   188   193   196   197
     0     0     0     0     0     0     6   112   158   177   183   183   190   195   201   203
     0     0     0     0     0     2     8   150   165   181   185   183   191   194   199   201
     0     0     0     0     0     0    12   158   169   185   186   187   192   196   202   203
     0     0     0     0     1     0   101   166   179   186   188   188   198   199   202   207
     0     0     0     0     2     2   158   174   179   184   188   189   197   199   202   207
     0     0     0     0     2     8   149   177   180   185   190   193   198   201   203   209
     0     0     0     0     9     0   155   180   185   190   194   199   201   206   205   211
     0     0     0     0     2    39   165   182   186   190   194   200   203   205   209   211
     0     0     0     0     5    87   171   182   187   194   197   200   204   209   209   213
     0     0     0     0     2   127   176   182   187   190   194   197   203   209   211   214
     0     0     0     0     5   135   176   182   189   194   197   201   204   209   209   212
     0     0     0     1     9   145   178   183   188   194   197   201   202   210   211   213
     0     0     0     2    11   147   181   184   187   193   196   201   205   210   212   213
     0     0     0     2     8   152   184   186   187   193   196   201   203   210   213   212
     0     0     0     2    10   154   188   187   187   193   196   201   204   210   212   215
     0     0     0     2     6   152   185   187   189   194   198   201   205   209   214   217
     0     0     0     2     4   154   185   187   189   192   200   202   207   211   216   218
     0     0     0     1    10   157   185   187   189   193   200   203   208   215   217   218
     0     0     0     1    13   157   184   186   187   194   199   207   211   218   218   221
     0     0     0     1    10   157   182   181   188   196   200   208   215   219   220   223
     0     0     0     0     3   148   182   186   187   190   200   207   216   221   224   226
     0     0     0     0     1   120   180   186   188   197   193   208   214   220   225   228
     0     0     0     0     3    63   168   182   188   196   203   201   216   224   225   230
     0     0     0     0     0     7   160   184   188   194   203   207   210   222   225   229
     0     0     0     0     3    12   160   180   187   192   199   205   212   216   215   230
     0     0     0     0     0     5   164   181   181   190   197   204   208   214   219   229
     0     0     0     0     0     1   120   175   180   186   195   204   210   217   219   222
     0     0     0     0     0     0    19   170   176   183   191   204   207   218   222   217
     0     0     0     0     0     3    13   161   160   180   192   201   208   216   218   218
     0     0     0     0     0     0     8   121   168   177   187   196   203   212   214   221
     0     0     0     0     0     1     2     5   160   165   181   190   202   209   214   218
     0     0     0     0     0     5     1     4   171   159   168   175   191   205   197   200
     0     0     0     0     0     4     3     0    38   158   178   187   194   200   210   214
     0     0     0     0     0     0     0     1    14   154   165   183   190   195   206   212
     0     0     0     0     0     0     0     5     1    23   166   176   186   196   201   201
     0     0     0     0     0     0     0     3     0     5   164   170   177   195   202   207
     0     0     0     0     0     0     0     2     0     6     6   152   179   186   195   204
     0     0     0     0     0     0     0     1     0     0     8    61   168   182   189   195
     0     0     0     0     0     0     0     0     0     0     4    16   128   168   177   185
     0     0     0     0     0     0     0     0     0     2     2     3    12   156   172   177
```

Figure 18: Matrix of Image

The DNN is implemented and the authors observed a cost curve as shown in Fig 19.

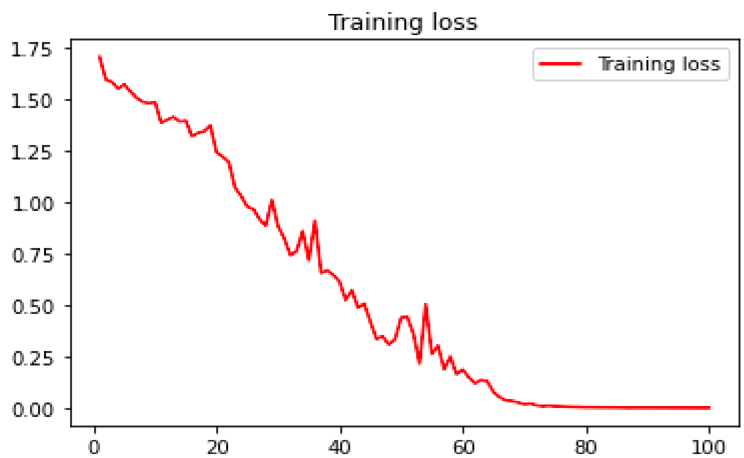
```
cost is  3700 0.7354018376268607
cost is  3800 1.357356296584426
cost is  3900 0.9407020984038749
cost is  4000 1.0103230426291951
cost is  4100 0.27954176629909105
cost is  4200 0.8485955818335696
cost is  4300 1.0695257574790231
cost is  4400 0.5889804562770888
cost is  4500 0.373873748224945
cost is  4600 1.839918235668986
cost is  4700 0.5151079420398439
cost is  4800 0.8757971456801477
```

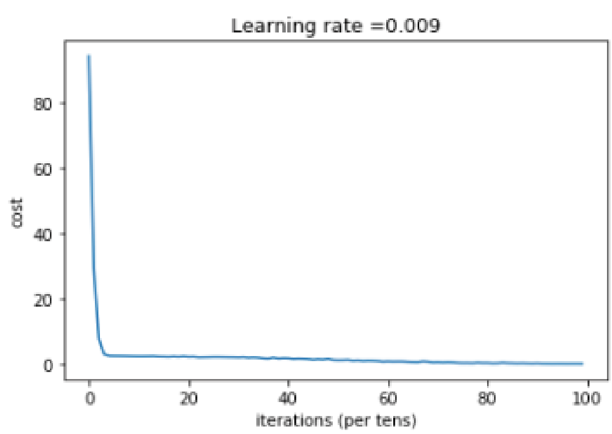



```
time: 1min 11s
```

Figure 19: Cost Curve

The cost curve tells about the performance of our algorithm for a particular iteration. The X-axis denotes the number of iterations our model had been trained on. Y-axis represents the probabilistic cost that our model has on a certain dataset. Lower cost corresponds to better performance and vice versa. We have observed after training deep neural networks indicates cost has gone down which means that our network is fitting data more accurately after whole training. Peaks are seen between the cost curve indicates the nonconvex nature of optimization, i.e. the nonconvex nature of the cost function. Moreover, the peaks may be present due to the learning rate that was chosen, if the learning rate is high, it's gonna maximize the function instead of minimizing it.






```
Tensor("Mean_1:0", shape=(), dtype=float32)
Train Accuracy: 0.992593
```

Fig 20: Learning Rate vs Iteration

From the curve shown in fig 20, it is seen that after each epoch the cost is decreasing. To overcome this problem, a mini-batch gradient descent approach is used in comparison to stochastic gradient descent. The advantage of mini-batch gradient descent is that it provides an advantage in the speed of learning. And it can be seen from the curve that cost is decreasing as the number of epochs is increasing. Table 2 shows the cost values for every 100 iterations from 4300 to 4800. As the cost first decreases slowly in the beginning, we decided to mention cost from 4300 iterations so that the decrease in cost can be easily viewable to the readers.

Table 2: Table depicting the cost after every 100 iteration

Iterations (in 100's)	**Cost**
4300	1.06
4400	0.58
00	0.37
4600	1.83
4700	0.51
4800	0.87

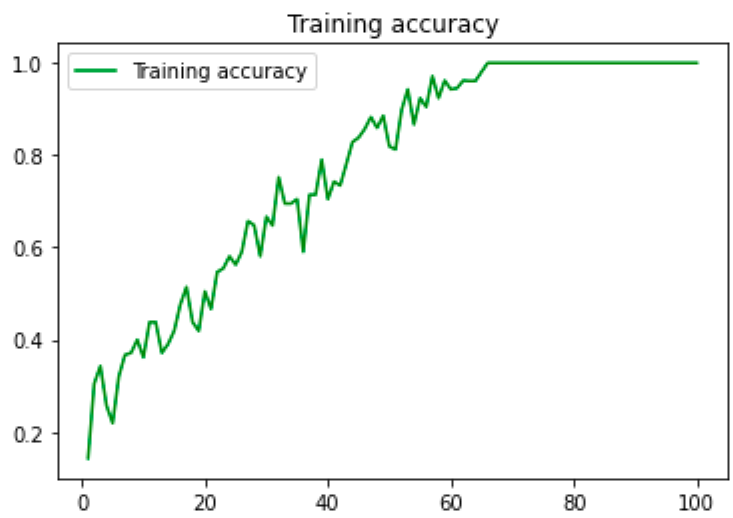
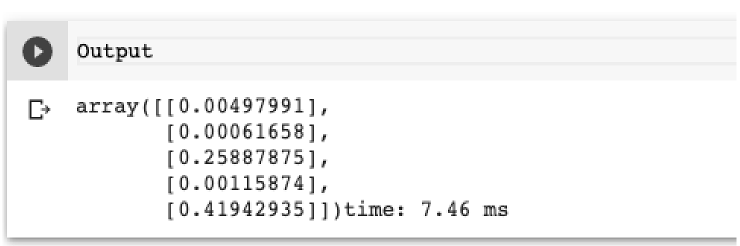


Fig 21:Training Accuracy vs Epochs

From Fig 21 it can be seen that training accuracy of our model has reached very closer to 100% on the training data.The model had been trained on 100 epochs our model converges to global optima of the cost function.

This output shows the probability of an image of being of a certain severity. The maximum probability is 0.4194 which is present at index 4, which means that the image which was fed to the deep neural network is of the severity of level 4 as shown in Fig.22.



```
Output

array([[0.00497991],
       [0.00061658],
       [0.25887875],
       [0.00115874],
       [0.41942935]])time: 7.46 ms
```

Fig 22: Output of Neural Network

The same simulations were carried out using conventional NN. This output shows the probability of an image of being of a certain severity. Here we see that the maximum probability is 0.101 Table 2 shows the comparison table between Conventional NN and CNN.

Table 3 : Comparison table depicting various features of both network and its outputs

	Neural Network	**Convolutional Neural Network**
Learning Rate	0.05	0.009
Probability	0.419	0.101
Time(ms)	7.46	5.62

From Table 3 and outputs it has been observed that with each iteration our cost is going down which means that our network is fitting data more accurately after every iteration in the neural network. While on CNN it has been observed from the curve that cost is decreasing as the number of epochs is increasing. Also, the time on CNN is less as that in the Neural network.

VGG-16

The table shows the accuracy and loss obtained at different epoch.

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	2.1792	68.81	0.9041	72.03
2	0.9205	73.67	0.8952	72.02
3	0.8863	74.19	0.9066	72.02

4	0.8814	73.82	0.8999	72.05
: :				
28	0.8488	74.25	0.9011	72.04
29	0.8643	73.52	0.9008	72.05

Table 4: Training loss and accuracy and Validation loss and accuracy for some intermediate epochs.

From Table 4, it has been observed that from the first epoch validation loss was almost the same 0.9041 which means this model achieved the Bayes error that was 0.8999. As it can be seen even for the training network for 29 epoch validation did not go down which means the model achieved the Bayes error and loss can't be further reduced, no matter for how many epochs you train the model. Further seeing these results, the training of the model is stopped.

Figure show that X-axis represents the epochs and the Y-axis represents the accuracy.It can be observed from the figure that at epoch 0 ,the accuracy was 68.81 % and at an intermediate epoch it reaches 73.85% and then reaches its maximum and remains constant.

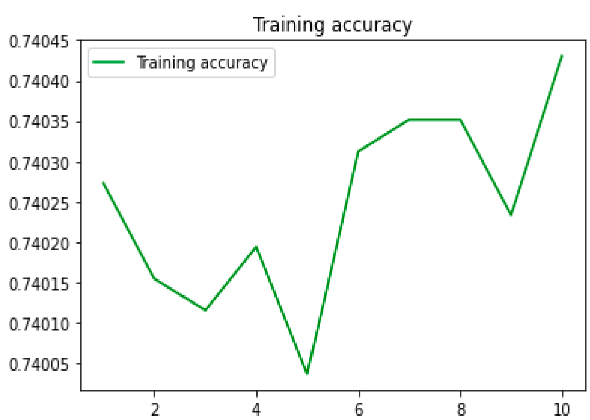


FIG 23: VGG-16 EPOCH VS ACCURACY

Figure show that X-axis represents the epochs and the Y-axis represents the loss.It can be observed from the figure that at epoch 0 ,the accuracy was 2.17 and at an intermediate epoch it reaches 0.8535 and then reaches its minimum of 0.8350 at the last epoch.

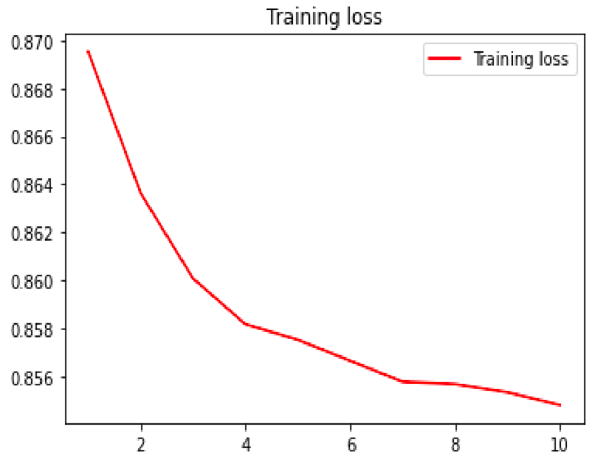


FIG 24: VGG-16 EPOCH VS LOSS

Inception V-3

From Fig ,it can be seen that training accuracy of our model has reached very closer to 74% on the training data.And a accuracy of 72% on the validation data.The model had been trained on 10 epochs our model converges to global optima of the cost function.X-axis of the Curve represents the epoch and Y-axis represents the accuracy.

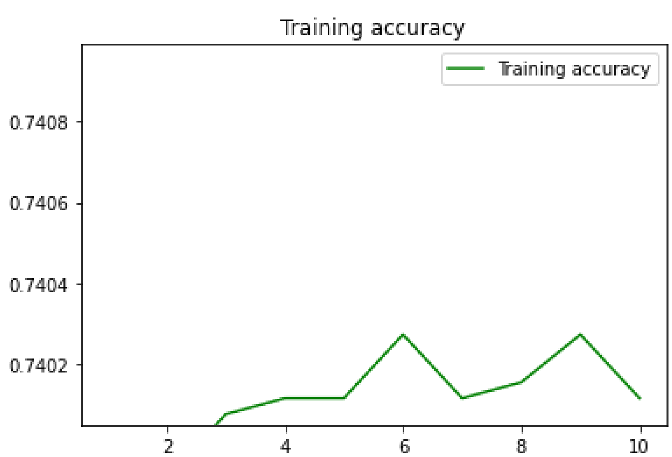


FIG 25: INCEPTION V3 EPOCH VS ACCURACY

From the curve, it is seen that after each epoch the cost is decreasing.A mini-batch gradient descent approach is used in comparison to stochastic gradient descent. X-axis of the Curve represents the epoch and axis represents the loss.

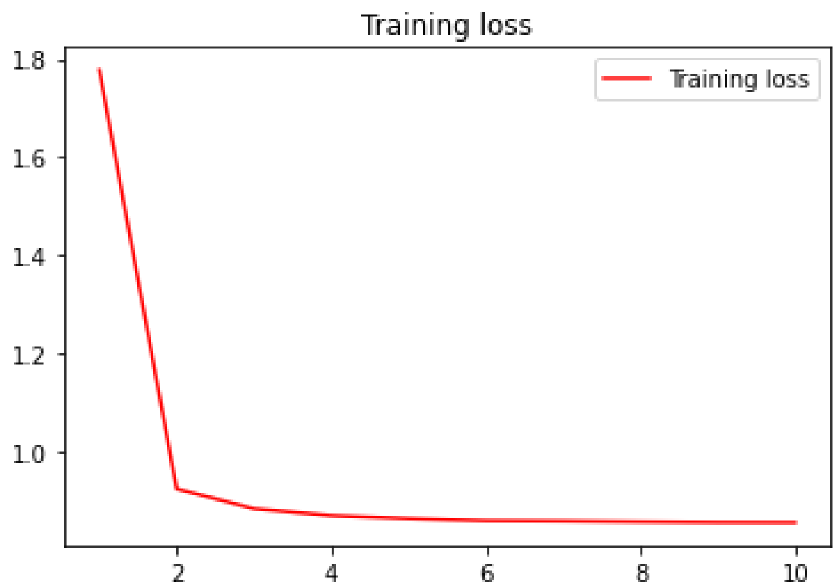


FIG 26: INCEPTION V3 EPOCH VS LOSS

Res-Net 101

```
Epoch 1/20
92/92 [==============================] - 49s 472ms/step - loss: 6.4536 - acc: 0.5045 - val_loss: 0.9843 - val_acc: 0.7131
Epoch 2/20
92/92 [==============================] - 40s 433ms/step - loss: 0.9692 - acc: 0.6752 - val_loss: 1.0179 - val_acc: 0.6940
Epoch 3/20
92/92 [==============================] - 41s 436ms/step - loss: 0.8110 - acc: 0.7110 - val_loss: 0.8065 - val_acc: 0.7404
Epoch 4/20
92/92 [==============================] - 41s 440ms/step - loss: 0.7531 - acc: 0.7355 - val_loss: 0.6995 - val_acc: 0.7637
Epoch 5/20
92/92 [==============================] - 41s 444ms/step - loss: 0.6949 - acc: 0.7455 - val_loss: 0.6419 - val_acc: 0.7514
Epoch 6/20
92/92 [==============================] - 42s 446ms/step - loss: 0.6761 - acc: 0.7516 - val_loss: 0.7141 - val_acc: 0.7322
```

|

\|/

```
Epoch 17/20
92/92 [==============================] - 42s 454ms/step - loss: 0.5717 - acc: 0.7776 - val_loss: 0.5888 - val_acc: 0.7678
Epoch 18/20
92/92 [==============================] - 42s 455ms/step - loss: 0.5951 - acc: 0.7794 - val_loss: 0.5621 - val_acc: 0.7828
Epoch 19/20
92/92 [==============================] - 42s 454ms/step - loss: 0.5730 - acc: 0.7858 - val_loss: 0.5328 - val_acc: 0.8128
Epoch 20/20
92/92 [==============================] - 42s 454ms/step - loss: 0.6024 - acc: 0.7712 - val_loss: 0.6076 - val_acc: 0.7664
```

FIG 27: The training accuracy and loss and validation accuracy and loss

The image above shows the training accuracy and loss and validation accuracy and loss that we got when we trained ResNet 101 on our gaussian filtered data. Gaussian filter helped us to reduce sample data noise.

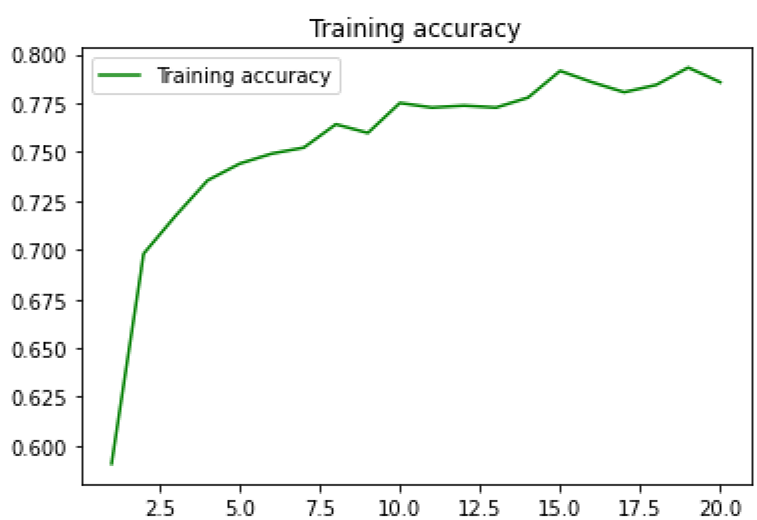


FIG 28:Training Accuracy

In Fig. above , X-axis represents the epochs and the Y-axis represents the accuracy , it can be observed from the image that at 1st epoch accuracy was 50.45% and at the 20th epoch the accuracy was 77.12% on training data and highest accuracy that we obtained while training our Network on validation data was 81.28%.

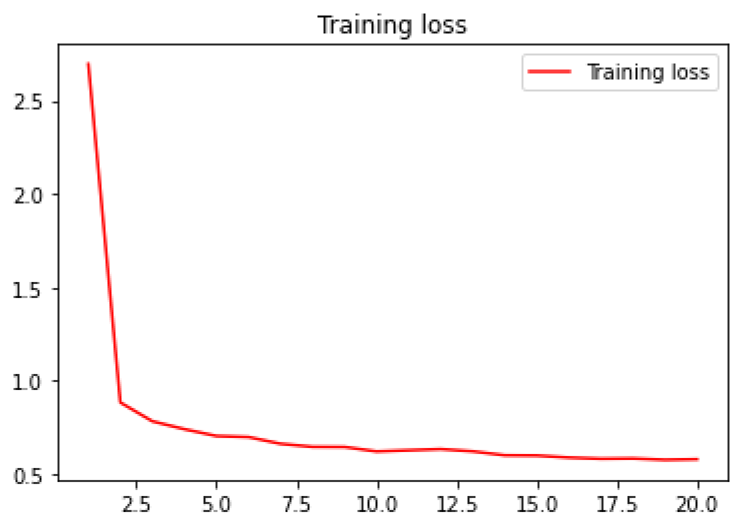


FIG 29: Training Loss

In Fig. above , X-axis represents the epochs and the Y-axis represents the loss , it can be observed from the image that at 1st epoch loss was 6.4536 and at the 20th epoch the loss was 0.6024 on training data and least loss that we obtained while training our Network on validation data was 0.5328.

These results that we obtained by training ResNet-101 are well fitted for this architecture as at 20th epoch training accuracy was 77.12% and validation accuracy was 76.64 which show that trained Network neither has bias nor variance.

```
Epoch 1/20
92/92 [==============================] - 100s 491ms/step - loss: 6.9577 - recall: 0.4642 - val_loss: 1.1529 - val_recall: 0.7104
Epoch 2/20
92/92 [==============================] - 43s 465ms/step - loss: 0.9518 - recall: 0.5820 - val_loss: 0.7049 - val_recall: 0.5096
Epoch 3/20
92/92 [==============================] - 42s 457ms/step - loss: 0.8190 - recall: 0.6155 - val_loss: 0.7017 - val_recall: 0.5328
Epoch 4/20
92/92 [==============================] - 43s 461ms/step - loss: 0.7344 - recall: 0.6489 - val_loss: 0.6668 - val_recall: 0.6626
Epoch 5/20
92/92 [==============================] - 42s 456ms/step - loss: 0.7174 - recall: 0.6600 - val_loss: 0.6462 - val_recall: 0.7391
```

|

\|/

```
Epoch 16/20
92/92 [==============================] - 43s 458ms/step - loss: 0.6019 - recall: 0.7194 - val_loss: 0.6037 - val_recall: 0.6653
Epoch 17/20
92/92 [==============================] - 43s 458ms/step - loss: 0.5881 - recall: 0.7199 - val_loss: 0.5723 - val_recall: 0.7527
Epoch 18/20
92/92 [==============================] - 42s 457ms/step - loss: 0.5878 - recall: 0.7171 - val_loss: 0.5412 - val_recall: 0.7350
Epoch 19/20
92/92 [==============================] - 43s 457ms/step - loss: 0.5635 - recall: 0.7369 - val_loss: 0.5575 - val_recall: 0.6516
Epoch 20/20
92/92 [==============================] - 42s 457ms/step - loss: 0.5914 - recall: 0.7271 - val_loss: 0.5178 - val_recall: 0.7049
```

FIG 30: RECALL VS EPOCH

The above figure shows the recall of the model at every epoch .At epoch 20th we were able to get 72.71 % of recall on training data while the max recall that we got on validation data was 76.37% .This means of all the positive samples our model was able to predict 76.37% correctly.

```
Epoch 1/20
92/92 [==============================] - 50s 490ms/step - loss: 0.6838 - precision: 0.8253 - val_loss: 0.6355 - val_precision: 0.8031
Epoch 2/20
92/92 [==============================] - 42s 454ms/step - loss: 0.5801 - precision: 0.8444 - val_loss: 0.5241 - val_precision: 0.8674
Epoch 3/20
92/92 [==============================] - 43s 461ms/step - loss: 0.5618 - precision: 0.8542 - val_loss: 0.5637 - val_precision: 0.8613
Epoch 4/20
92/92 [==============================] - 43s 457ms/step - loss: 0.5697 - precision: 0.8534 - val_loss: 0.5153 - val_precision: 0.8842
Epoch 5/20
92/92 [==============================] - 43s 458ms/step - loss: 0.5420 - precision: 0.8443 - val_loss: 0.4851 - val_precision: 0.8779
Epoch 6/20
92/92 [==============================] - 43s 460ms/step - loss: 0.5376 - precision: 0.8412 - val_loss: 0.4677 - val_precision: 0.9002
```

\|/

```
Epoch 16/20
92/92 [==============================] - 43s 460ms/step - loss: 0.5160 - precision: 0.8527 - val_loss: 0.4828 - val_precision: 0.9011
Epoch 17/20
92/92 [==============================] - 43s 460ms/step - loss: 0.5372 - precision: 0.8545 - val_loss: 0.4648 - val_precision: 0.8933
Epoch 18/20
92/92 [==============================] - 43s 459ms/step - loss: 0.5111 - precision: 0.8552 - val_loss: 0.4409 - val_precision: 0.8750
Epoch 19/20
92/92 [==============================] - 43s 457ms/step - loss: 0.4925 - precision: 0.8615 - val_loss: 0.5508 - val_precision: 0.9041
Epoch 20/20
92/92 [==============================] - 43s 459ms/step - loss: 0.4786 - precision: 0.8640 - val_loss: 0.6577 - val_precision: 0.8729
```

FIG 31: Precision of the model at every epoch

The above figure shows the precision of the model at every epoch .At epoch 20th we were able to get 86.40% of recall on training data while the max precision that we got on validation data was 90.41% .This means that every image that we evaluate on our model will have 90.41% chances of getting classified correctly.

This means that every image that we evaluate on our model will have 90.41% chances of getting classified correctly.

The **table 5** below shows us the various network architecture used in our project. Also the two other columns depicts the Loss and the Accuracy we had achieved by training the images on the following networks. We depicted that ResNet-101 is the best suitable one as it had the least loss and the best Accuracy.

S.No	Network Arch.	Loss	Accuracy (%)
1	VGG-16	0.8643	73.52
2	Inception-V3	0.8514	74
3	ResNet-101	0.5328	81.28

Table 5: It depicts the various network architecture and the loss and accuracy obtained after training the different images.

CHAPTER 6

CONCLUSION

In our proposed solution, CNN is a solid method to manage all levels of diabetic retinopathy stages. Our framework plan with dropout methods yielded enormous portrayal exactness. The

clinical measurements are an unmistakable sign that the proportion of patients to ophthalmologists is over a lakh and henceforth a productive robotization framework is profoundly attractive for mass screening of DR, especially in country regions where the circumstance is all the more disturbing.From the results that the author has gotten it can be further concluded that for any constant size Network architecture, the loss can be further reduced after a certain number i.e. Bayes error. Further, it can be concluded that increasing the number of epochs will not always lead to a lower loss or higher accuracy, it's only possible only until loss reaches Bayes error. Also after using different CNN architecture we observed that VGG-16 and Inception-V3 performed almost the same for our problem , but ResNet-101 gave us results that were better than VGG-16 and Inception-V3 network Architecture.

.

REFERENCES

1. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
2. https://cs231n.github.io/convolutional-networks/
3. https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/
4. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
5. www.towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c
6. https://en.wikipedia.org/wiki/Data_augmentation
7. https://www.mygreatlearning.com/blog/understanding-data-augmentation/
8. www.towardsdatascience.com/adam-optimization-algorithm-1cdc9b12724a
9. Diederik P. Kingma, Jimmy Lei Ba∗ University of Toronto, ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION, a conference paper at ICLR 2015, pp 2-5
10. www.coursera.org/learn/deep-neural-network/lecture/y0m1f/gradient-descent-with-momentum
11. WangMin Liao, BeiJi Zou, RongChang Zhao*, YuanQiong Chen, ZhiYou He, and MengJie Zhou, Clinical Interpretable Deep Learning Model for Glaucoma Diagnosis, IEEE Journal of Biomedical and Health Informatics, IEEE 2019, 23 October 2019, VOL.24 pp 4-10.

12. H. Pratt, F. Coenen, D. M. Broadbent, S. P. Harding, Y. Zheng,Convolutional Neural Networks for Diabetic Retinopathy, International Conference On Medical Imaging Understanding and Analysis 2016, MIUA 2016, 6-8 July 2016, Loughborough, UK VOL.6 pp 15-19

13. Mohamed Shaban, Zeliha Ogur ,Ali Mahmoud ,Andrew Switala, Ahmed Shalaby,Hadil Abu Khalifeh, Mohammed Ghazal, Luay FraiwanID A convolutional neural network for the screening and staging of diabetic retinopathy 2020,UCF 2020,Dec 2 2019-June 22 2020,Florida Vol 6 pp 4-12

14. R. Poplin, A.V. Varadarajan, K. Blumer, Y. Liu, Michael V. McConnell, G. S. Corrado, L. Peng, Dale R. Webster, "Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning", Nature Biomedical Engineering, volume 2, pp 158–164 (2018)

15. A Rodtook,Sirikan Chucherd,::Automated Optic Disc Localization Algorithm by Combining A Blob of Corner Patterns, Brightness and Circular Structures Models 2019 Association for Computing Machinery ITCC 2019, August 16–18, 2019, Singapore VOL.7 pp 45-52

16. A Rodtook,Sirikan Chucherd,::Automated Optic Disc Localization Algorithm by Combining A Blob of Corner Patterns, Brightness and Circular Structures Models 2019 Association for Computing Machinery ITCC 2019, August 16–18, 2019, Singapore VOL.7 pp 45-52

17. https://www.analyticsvidhya.com/blog/2020/10/what-is-the-convolutional-neural-network-architecture/

18. https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/

19. https://paperswithcode.com/method/resnet

20. https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8

21. https://www.geeksforgeeks.org/vgg-16-cnn-model/

22. https://www.kaggle.com/blurredmachine/vggnet-16-architecture-a-complete-guide

23. https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e

24. https://machinelearningmastery.com/cross-entropy-for-machine-learning/

LIST OF PUBLICATIONS

1. S.Sinha , S.Saxena and S.Jain **“COMPUTER-AIDED DIAGNOSTIC SYSTEM FOR DIABETIC RETINOPATHY USING CONVOLUTIONAL NEURAL NETWORK”** 7-8 November INNOVATIONS IN INFORMATION AND COMMUNICATION TECHNOLOGIES2020 IICT **[Published]**

2. S.Sinha , S.Saxena and S.Jain **“TRENDS OF ACCURACY ON DIABETIC RETINOPATHY CONSIDERING EPOCH EFFECT AND BAYES ERROR”** **April** 2021 IEEE IAS GUCON **[Communicated]**

PLAGIARISM

Plag report (Shivam)

ORIGINALITY REPORT

18%	15%	6%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Jaypee University of Information Technology Student Paper	2%
2	Submitted to University of Essex Student Paper	2%
3	towardsdatascience.com Internet Source	1%
4	Submitted to Texas A&M University - Commerce Student Paper	1%