

**IMPACT OF INHERENT PARAMETERS ON THE PERFORMANCE OF
CONVOLUTIONAL NEURAL NETWORKS FOR THE CLASSIFICATION OF
HERBAL IMAGES**

Submitted in partial fulfilment of requirements for the award of
Bachelor of Engineering in
Electronics and Communication Engineering

by

KARLAPATI SAI TEJA (37130187)

HARSHAVARDHAN T (37130159)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

APRIL – 2021



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KARLAPATI SAI TEJA (Reg.No.37130187)** and **HARSHAVARDHAN T (Reg.No.37130159)** who carried out the project entitled "**IMPACT OF INHERENT PARAMETERS ON THE PERFORMANCE OF CONVOLUTIONAL NEURAL NETWORKS FOR THE CLASSIFICATION OF HERBAL IMAGES**" under our supervision from November 2020 to April 2021.

Internal Guide

Dr. E. Logashanmugam M.E., Ph.D.

Pro Vice Chancellor

Head of the Department

Dr.T.RAVI M.E., Ph.D.

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

We, **KARLAPATI SAI TEJA (Reg.No. 37130187)** and **HARSHAVARDHAN T (Reg.No.37130159)** hereby declare that the Project Report entitled **“IMPACT OF INHERENT PARAMETERS ON THE PERFORMANCE OF CONVOLUTIONAL NEURAL NETWORKS FOR THE CLASSIFICATION OF HERBAL IMAGES”** done by us under the guidance of **Dr. E. Logashanmugam M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE:

PLACE: Chennai

1. KARLAPATI SAI TEJA

2. HARSHAVARDHAN T

Signature of the students

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA M.E., Ph.D.** Dean, School of Electronics and Communication Engineering and **Dr. T. RAVI M.E., Ph.D.** Head of the Department, Department of Electronics and Communication Engineering for providing me necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. E. Logashanmugham M.E., Ph.D.** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

We would like to thank our beloved PARENTS for their constant love and support, without whom this wouldn't be possible and finally we would like to thank the ALMIGHTY for his wonderful blessing.

ABSTRACT

In recent years, machine learning has shifted more towards deep learning neural networks. Deep learning neural networks are broadly classified into Recurrent Neural Networks, Long Short Term Memory Recurrent Networks and Convolutional Neural Networks. While the former two are limited to one Dimensional applications, the latter is widely used for image analysis namely prediction and classification.

A Convolutional Neural Network (CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a CNN is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, CNN have the ability to learn these filters/characteristics.

However the performance of these networks is strongly dependent on the inherent parameters namely padding, filter size, number of connections, number of layers, strides and the nature of the pooling layers. Hence the proposed work aims at studying the impact of the above parameters on the performance of the networks and in optimising or modifying certain parameters for improving the performance of the network in the classification of herbal images.

List of Contents

S.No.	Title	Page No.
	Abstract	
1.	Introduction	1
	1.1 Problem Definition	2
	1.2 Objectives	2
	1.3 Organization of the report	3
	1.4 Literature Survey	3
2.	Research Database and Software Requirements	5
	2.1 MATLAB software details	9
	2.1.1 Applications of MATLAB	11
	2.1.2 Advantages of MATLAB	11
3.	Existing work for the classification of herbal images	13
	3.1 Convolution Neural Network – An Overview	13
	3.2 Classification of Herbal Images using GOOGLNET	16
	3.3 Classification of Herbal Images using SQUEEZENET	18
	3.4 Classification of Herbal Images using RESNET50	20

4.	Proposed Work	22
5.	Result	23
	5.1 Conclusion	30
	5.2 Future Work	31
	References	32

List of Tables

S.No.	Name	Page No.
1.	1.1: Literature Survey	3
2.	2.1: Database of Training Images	6
3.	2.2: Database of Testing Images	8
4	5.1: Confusion matrix of ALEXNET with learning rate of 10^{-4} .	24
5.	5.2: Confusion matrix of ALEXNET with learning rate of 10^{-7} .	25
6.	5.3: Confusion matrix of ALEXNET with learning rate of 10^{-5} .	26
7.	5.4: Confusion matrix of ALEXNET with Bias value 5.	27

8.	5.5: Confusion matrix of ALEXNET with learning rate of 10^{-4} and number of epochs is 5.	28
9.	5.6: Confusion matrix of ALEXNET with learning rate of 10^{-5} and number of epochs is 5.	29
10.	5.7: Efficiency of the neural networks with respect to the changed parameters.	30

List of figures

S.No.	Name	Page No.
1.	3.1: Block diagram of CNN	13
2.	3.2: Gradient of of a ReLU activation function	16
3.	3.3: Performance of GoogleNet in terms of sensitivity	18
4.	3.4: Architecture of GoogleNet	18
5.	3.5: Performance of SqueezeNet in terms of sensitivity	19

6.	3.6: Architecture of SqueezeNet	19
7.	3.7: Performance of Resnet50 in terms of sensitivity	20
8.	3.8: Architecture of RESNET50	21

CHAPTER 1

INTRODUCTION

India is a country noted for its rich cultural heritage and medicinal values. Right from ancient days, medicinal herbs have become an integral part in every household. Even today diseases like common cold, asthma and body pain are cured with herbs and herbal medicines. A wide variety of herbs with medicinal values are still prevalent in various parts of India especially in hills and mountains. It has attracted a large number of researchers for pursuing their research work. Also, when these varieties of herbs when categorized and digitized would be of great support to the field of medicine. In recent days, multiple researchers are working to bridge the gap between modern medicine and herbal medicine. Considerable research has been carried out in this area for automated characterization of anomaly/abnormality/flowers/herbs. Selvarasu et al used Euclidean based color image segmentation for the extraction of abnormal features. Conventionally features are extracted and these features are fed to the classifiers. Conventional classification techniques involve the following steps: Features are extracted from the input images, exemplars are created with the features as input parameters and the type of the herb as the output parameter, exemplars are divided into two sets one for training and the other for testing, neural network is trained and tested, trained neural network and its weights are stored. Having done this, any input image of the herb (to be classified) is considered. Features are extracted and these features are fed to the already trained neural network and the output is obtained. The output indicates the type of the herb. In the above case, performance of the classifiers is strongly dependent on the extracted features. Hence an efficient feature extraction technique has to be identified. Though various image segmentation techniques are available in literature these techniques do not work well for real time images. Computational complexity is more as the features are extracted with an image segmentation technique followed by feature representation. Hence it is necessary to develop a classifier that does not involve feature extraction but performs automated classification with higher efficiency.

1.1 Problem Definition

The proposed work aims at improving the accuracy, size and speed of the pretrained Deep Neural Networks in R2020b MATLAB Deep Learning Tool Box.

1.2 Objectives

Objectives of this project are listed below:

- To collect herbal images database.
- To analyze the impact of inherent parameters on CNN (with physical and mathematical approach)
- To optimize the number of layers, padding, strides, pooling layers, training methods, learning algorithms, initial weights for herbal classification.
- To train the network for classifying images.
- To evaluate the performance of novel and conventional CNN for the classification of herbal images.

1.3 Organization of the report

This is organized as follows:

- Chapter 2 is provided with research database and software requirements.
- Chapter 3 deals with existing CNN networks for classification of herbal plants.
- Chapter 4 deals with proposed work.
- Chapter 5 deals with Conclusion and future work.

1.4 LITERATURE SURVEY

TABLE 1.1: Literature survey

S. No.	Title	Authors	Content
1.	Evaluation of Convolutional neural network, Alex net and Google Net for fruit recognition	Nur Azida Muhammad, Amelina A Nasir, Zaidah Ibrahim, Nurbaity Sabri	The exhibition of essential Convolutional Neural Network (CNN), Alexnet and Googlenet in perceiving nine unique kinds of organic products from a freely accessible dataset.
2.	Weed Seedling Detection Using Mask R-CNN	Sanjay Patidar, Utkarsh Singh, Sumit Kumar Sharm, Himanshu	Mask R-CNN will assist the individual with isolating weed from essential yield so that harvest will get full sustenance and production gets expanded.
3.	Mutant classification using Convolutional neural network	Yeh Huann Goh Chee Ho Ng Yoon Ket Lee	The perceptible characteristics of freaks of a comparative genotype plant communicating with various biological conditions is basic to perceive how well the introduction of a particular property in different improvement environment. It Intends to assemble an ideal cnn for picture-based plant freak order task.
4.	Mobile leaf identification using Convolutional neural network	Kaori Sasaki Yuichi Sasaki, Yosuke Kobayashi, Tatsuhiro Akiyama	A Mobile application using convolutional neural networks (CNNs) that will help beginners identify plant species.
5.	Plant Leaf disease	Melike Sardogan, Adem Tuncer	Programmed techniques for arrangement of plant illnesses additionally help making a

	Detection and Classification based on CNN With LVQ algorithm	Yunus Ozen	move in the wake of recognizing the side effects of leaf diseases. A CNN and LVQ algorithm are used for the detection of infection on leaves and also classifies them.
6.	Deep Residual Learning for Image Recognition	Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun	Development of residual learning technique framework for efficient Image Recognition.
7.	Herbal Plant Recognition	Anh H. Vo, Hoa T. Dang, Bao T.	Development of Novel Neural networks for recognition of herbal plants for medicinal uses and business purposes.
8.	Performance Evaluation of Googlenet, SqueezeNet, and Resnet50 in the Classification of Herbal Images	M.A. Muthiah, E.Logashamugam, N.M. Nandhitha	Computer aided classification of medicinal herbs is of major concern in medicinal research. Real challenge lies in the complexity and variability of plants belonging to the same species. Convolutional Neural Networks are the recently accepted paradigm for classification with the help of pretrained neural networks. GoogleNet, SqueezeNet, Resnet 50 for the classification of medicinal herbs is studied.

CHAPTER 2

RESEARCH DATABASE AND SOFTWARE REQUIREMENTS

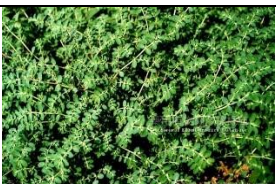
We have got the database images from various sources. Database consists of total 14 categories. Each category contains minimum of 8 and maximum of 30 images. Total database consists of 489 images. We have divided the total database into two halves. Half of the pictures are utilized for testing and half of the pictures are utilized for training. The below table contains names of 14 categories and one image from that category.

Table 2.1 Database of Training Images

S.No.	Image Name	Image
1.	Apiaceae	
2.	Asteraceae	
3.	Euphorbiaceae	
4.	Fabaceae	
5.	Lamiaceae	
6.	Liliaceae	

7.	Poaceae	
8.	Polygonaceae	
9.	Ranunculaceae	
10.	Rosaceae	
11.	Rubiaceae	
12.	Rutaceae	
13.	Solanaceae	
14.	Zingiberaceae	

Table 2.2 Database of Testing Images

S.No.	Image Name	Image
1.	Apiaceae	
2.	Asteraceae	
3.	Euphorbiaceae	
4.	Fabaceae	
5.	Lamiaceae	
6.	Liliaceae	
7.	Poaceae	
8.	Polygonaceae	

9.	Ranunculaceae	
10.	Rosaceae	
11.	Rubiaceae	
12.	Rutaceae	
13.	Solanaceae	
14.	Zingiberaceae	

We will be using MATLAB R2020B along with Deep Learning Tool Box.

2.1 MATLAB SOFTWARE DETAILS

MATLAB is a superior language for particular processing. Matlab facilitates computation, discernment, and coding in an ease to-use environment where issues and plans are conveyed in unmistakable documentation.

Typical utilizations include:

1. Estimation and math
2. Estimation headway
3. Illustrating, re-enactment, and prototyping
4. Data assessment, examination, and insight
5. Sensible and planning outlines
6. Application headway, including GUI building
7. An instinctive numerical figuring environment
8. System estimations

Designs

Programming (M-records)

Tool compartments (signal preparing, insights, advancement, emblematic math)

Matlab (MATRIX laboratory) is a mathematical figuring climate and fourth generation programming language.

Matlab with Simulink building a powerful model can really be enjoyable. It is additionally incredible enough to accomplish genuine work, when you gain proficiency with a portion of its high-level highlights (i.e., it's anything but an exercise in futility to learn). Truth be told, we will just utilize a portion of the somewhat fundamental highlights of Simulink for the models we fabricate, and you are free (and empowered) to investigate some different highlights that will assist you with issues you need to settle. Fundamentally you can think about a Simulink model as like depicting a wiring chart of your framework, where the factors stream on wires and are changed, or changed, by different gadgets. We should assemble a Simulink model for the direct spring-mass framework with gooey damping: First, you need to begin the tangle lab application, which you do in Unix by composing "Matlab" at the

shell brief, or in Windows by double tapping (quickly tapping the left mouse button twice) on the Matlab symbol, which you will discover covered some place with a lot of different symbols. This will get you to the ">>" Matlab brief. Type "Simulink" to begin Simulink, and raise the primary Simulink window (your windows may look fairly changed, contingent upon the machine and working framework): Let's go through the essential elements of this window first.

On top is a rundown of things that are related with pull-down menus. These will be copied on your model workspace (see underneath), so generally they are not very significant here. Except for the "document" menu, which you will use to make another model workspace or to raise a record program to get to a current (saved) model. Every one of the square symbols relate to a class of changes that will be utilized to depict the unique framework. You will for the most part utilize the changes depicted in "Sources," "Sinks," "Straight," "Nonlinear," and "Associations." The "Sources" symbol contains a set-up of sign generators, like irregular number generators and wave (sinusoidal, square, and so forth) generators, that can be utilized to give time-fluctuating sources of info that drive your dynamic framework (these are exogenous information sources, or outside inputs, implying that their elements are portrayed remotely to the model). The "Sinks" classification will for the most part be utilized to get to the "Degree" and XY charts to which you will associate the framework "wires" to get a "readout." The "Straight" class contains every one of the potential approaches to change a variable directly, like various sorts of gains. It additionally has the essential "integrator" block that we will utilize widely to demonstrate our first-request dynamic frameworks.

Correction and end debugging

While investigating, we can modify the estimation of a variable to check whether the new worth produces anticipated outcomes. While the code is stopped, dole out another worth to the alphabet in the command window, workspace program. At that point keep running and venturing through the program.

Ending debugging

In the wake of recognizing an issue, end the investigating meeting. It is ideal to stop investigate mode prior to altering a M-record. Else, we got startling outcomes when

we are running the record. To stop troubleshooting, debugging mode from debugging menu should be selected.

Merits of Matlab

End of mistakes right off the bat in the plan

Powerful control framework

Less emphasis in the improvement cycle, diminished advancement time and cost, and so forth

In addition, these instruments can be coordinated with constant equipment which means incorporating conventional disconnected re-enactment with true testing.

An apparently improved on method that absolves an originator from encoding his numerical model in the programming language embraced,

A chance of a various utilization of the essential numerical models, assembled at the libraries and joined on a case-by-case basis.

2.1.1 APPLICATIONS OF MATLAB:

Numerical Calculations

Data Analysis and Visualization

Software Development

Simulation

2.1.2 ADVANTAGES:

It's exceptionally simple for a beginner in PC programming.

It accompanies well-written manuals.

Huge client local area, sharing free codes. It allows you to work in Matrices without any problem. It allows you to do some convoluted computations rapidly, without composing an entire program or accounting page. Huge loads of additional items and workbenches accessible to do an assortment of errands.

Simple, intuitive environment

Quick mathematical calculations

Modest programming

Inexpensive software

Fun and simple climate for learning Linear Algebra!!

CHAPTER 3

EXISTING WORK FOR THE CLASSIFICATION OF HERBAL IMAGES

This chapter provides an overview of the existing GOOGLNET, SQUEEZENET AND RESNET50 architectures. Performance of these networks in classifying the potatoes from thermal images is also provided. Performance is measured in terms of sensitivity and accuracy.

3.1 CONVOLUTION NEURAL NETWORK – AN OVERVIEW

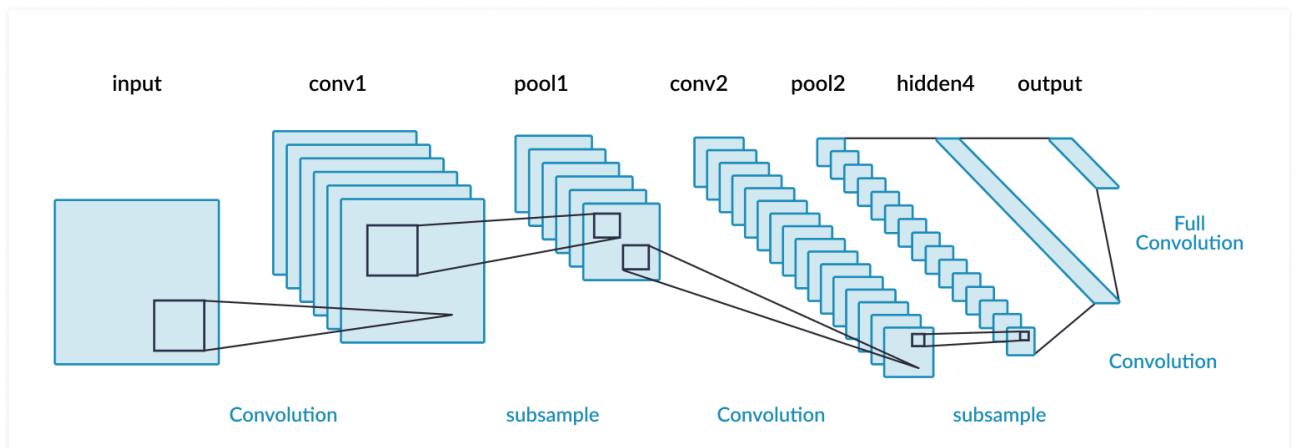


Fig 3.1 Block diagram of CNN

CNN is a type of Deep Neural Networks which can perceive, characterize specific highlights from pictures and are generally utilized for investigating visual pictures. Their applications range from picture and video acknowledgment, picture characterization, clinical picture investigation, PC vision and regular language preparing. The expression "Convolution" in CNN means the numerical capacity of convolution which is an exceptional sort of direct activity wherein two capacities are increased to create a third capacity which communicates how the state of one capacity is changed by the other. In straightforward terms, two pictures which can be addressed as grids are increased to give a yield that is utilized to extricate highlights from the picture.

There are 2 major parts CNN design:

A convolution tool that isolates and recognizes the different highlights of the picture for examination in an interaction called as Feature Extraction

A completely associated layer that uses the yield from the convolution interaction and predicts the class of the picture dependent on the highlights extricated in past stages.

Layers of Convolutional Neural network are:

1. Convolved Layer:

It is the important layer that is used for removing different highlights from the information pictures. In this layer, the numerical activity of convolution is performed between the info picture and a channel of a specific size $M \times M$. By sliding the channel over the info picture, the dab item is taken between the channel and the pieces of the information picture concerning the size of the channel ($M \times M$).

The yield is named as the Feature map which gives us data about the picture like the corners and edges. Afterward, this component map is taken care of to different layers to gain proficiency with a few different highlights of the info picture.

2. Strides

Stride helps in moving the channel grid over the pixels in the component network. On the off chance that the step size is 1, we can move the channel to 1 pixel at a time. On the off chance that the info grid measurement is $n \times n$, channel measurement is $f \times f$ and step size $= s$ then the yield size will be equivalent to $((n-f/s) + 1) \times ((n-f/s) + 1)$. In the event that we increment the step size to a higher worth, there is no yield so we can utilize cushioning for it.

3. Padding

Padding helps in making the output and input size equal.

Reasons to use padding:

To avoid shrinking of output image if we increase the number of convolution layers.
To make use of the outer edge along with the inner edge of the input image while doing convolution. If padding $p=1$ then it adds two rows and two columns for the

input matrix. Mostly we use zero padding. In zero padding we assign zeros to the newly formed rows and columns.

Types of Padding:

Valid padding: In valid padding there is no padding involved. So, the dimensions of input and output are different.

Same padding: Same padding helps in making the dimensions of convolution output dimension and input dimension equal.

4. Pooling Layer

- It is used to reduce the dimensions of the feature map.
- It is used to extract the dominant features.
- Strides and padding are also available for pooling.
- There are two types of pooling:
- Max pooling.
- Average pooling.
- Max pooling selects the maximum value in that respective 2D matrix.
- Average pooling takes the average of values in that respective 2D matrix.
- Max pooling is better than average pooling.

.

5.ACTIVATION FUNCTION

Activation functions are conditions that decide the output of a neural network. It helps in normalizing the output of every neuron to a reach between [1 and 0] (or) between [- 1 and 1]. ReLU is the most usually utilized enactment work in any neural organization. The essential condition of ReLU actuation work is $\max(y,0)$, where y is the particular tangled yield framework component.

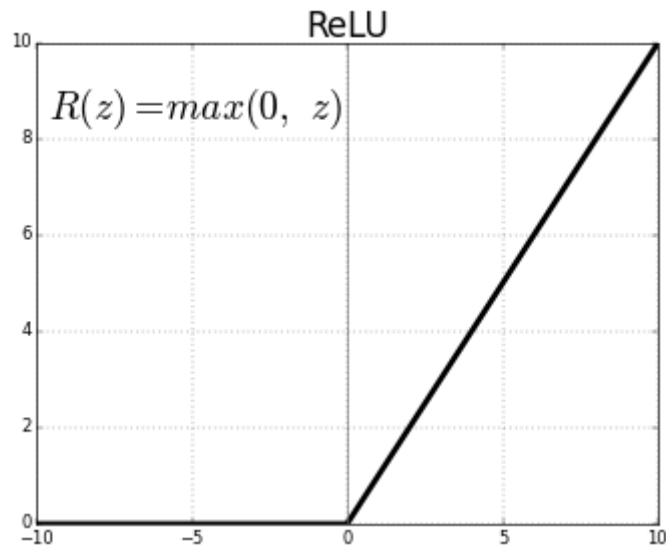


Fig 3.2 Gradient of a ReLU activation function

This graph implicates that the negative values in the feature matrix is made to 0. So, the values range from 0 to 1.

6. SOFTMAX

SoftMax gives us the last likelihood for each mark and allots decimal probabilities to each class in a multi-class issue. It is likewise a kind of Activation Function. Information esteems can be negative, zero, positive, or more noteworthy than one however the yield esteems will be somewhere in the range of 0 and 1, so they can be deciphered as probabilities.

7. FLATTENING

A Fully connected output layer takes the yield of the past layers 'flattens' them and transforms them into a single vector. Flattening is the primary benefit of CNN over Feed-Forward Neural Networks.

8. Fully Connected Layer (FC):

Full Connected (FC) layer comprises loads & inclinations alongside the neuron it utilized interface the neurons among 2 distinct layers. The layers are ordinarily situated before output layer & design several layers of a CNN Architecture.

3.2 CLASSIFICATION OF HERBAL IMAGES USING GOOGLNET

The GoogleNet architecture comprises of 22 layers (27 layers including pooling layers), and a piece of these layers are an aggregate of 9 inception modules.

The information layer of the Google Net engineering takes in a picture of the measurement 224 x 224.

Type: This alludes to the name of the current layer of the part inside the architecture

Patch Size: Refers to the size of the broad window used across convolution and pooling layers. Sweeping windows have equivalent height and width.

Stride: Defines the measure of move the channel/sliding window assumes control over the information picture.

Output Size: The subsequent yield measurements (height, width, number of feature maps) of the current design part after the information is gone through the layer.

Depth: Refer to the quantity of levels/layers inside an architecture part.

#1x1 #3x3 #5x5: Refers to the different convolution's channels utilized inside the inception module.

#3X3 reduce #5x5 reduce: Refers to the quantities of 1x1 channels utilized before the convolutions.

Pool Proj: This is the quantity of 1x1 channels utilized in the wake of pooling inside an inception module.

Params: Refers to the quantity of loads inside the current architecture segment.

Operations: Refers to the quantity of numerical activities did inside the segment.

In this progression, the pretrained GoogleNet architecture is stacked and is prepared and tried with two unique arrangements of pictures. At that point the architecture is finalized and the architecture is prepared for use for the given picture database.

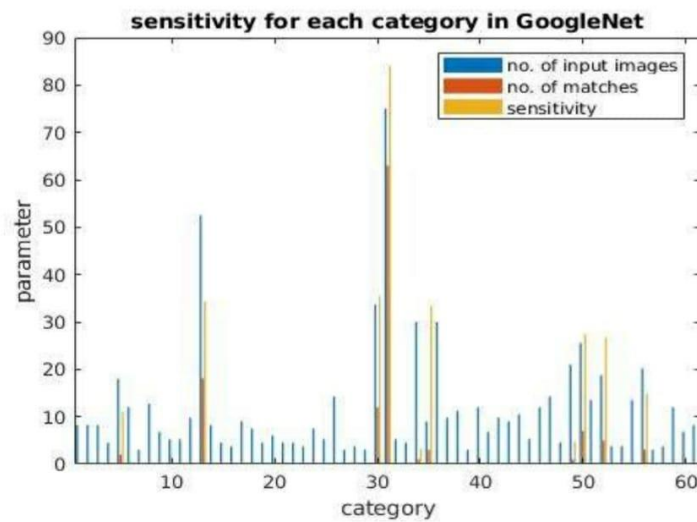


Fig 3.3 Performance of GoogleNet in terms of sensitivity

Half of pictures are utilized for preparing and half of pictures are utilized for testing. At that point the organization is utilized for deciding the class of the keyimage. In general exactness of GoogleNet is 16.5%

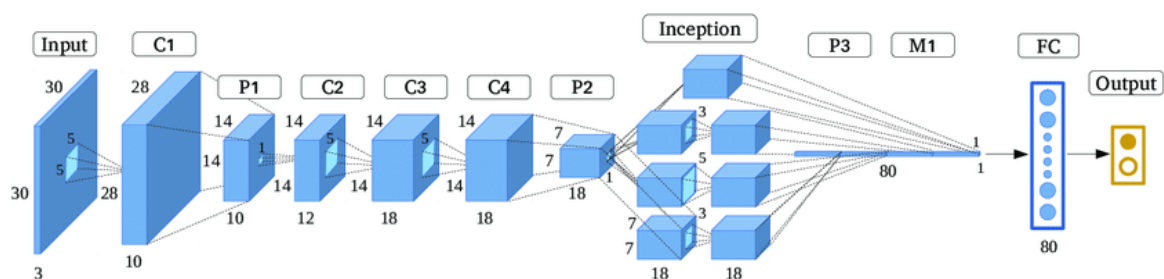


Fig 3.4 Architecture of GoogleNet

3.3 CLASSIFICATION OF HERBAL IMAGES USING SQUEEZENET

Having understood the feasibility of convolutional Neural Networks, the next step is to develop FPGA for the pretrained networks. In such case, smaller the size, easier is its implementation. Such a smaller CNN is SqueezeNet, where major modifications are performed. A major number of 3x3 filters in convolution layer are replaced with 1x1 filters without affecting the accuracy of operation. Number of

inputs to each filter is also reduced. A module called fire module is developed which has a convolution layer (1x1 filter) and an expand layer (that has both 1x1 and 3x3 filters). Also, SqueezeNet uses maxpooling technique.

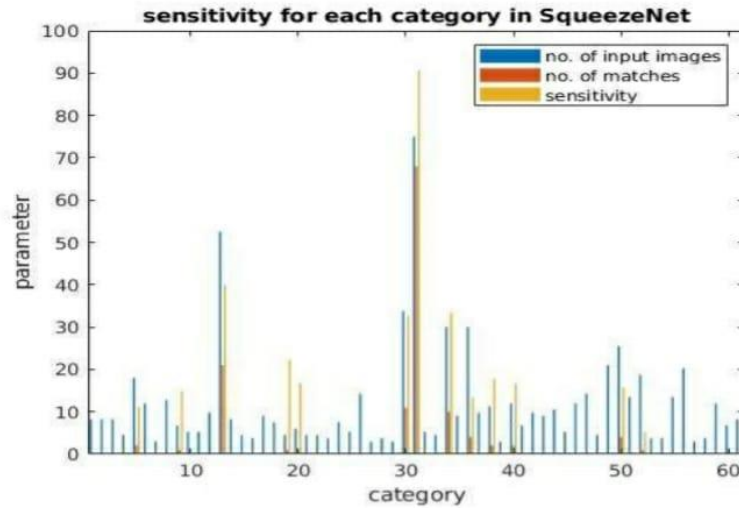


Fig 3.5 Performance of SqueezeNet in terms of sensitivity

In Squeezenet also, 50% of the images are used for training and 50% is used for testing. Then the network is used for determining the class of the key image. Overall accuracy of Squeezenet is 18.3%.

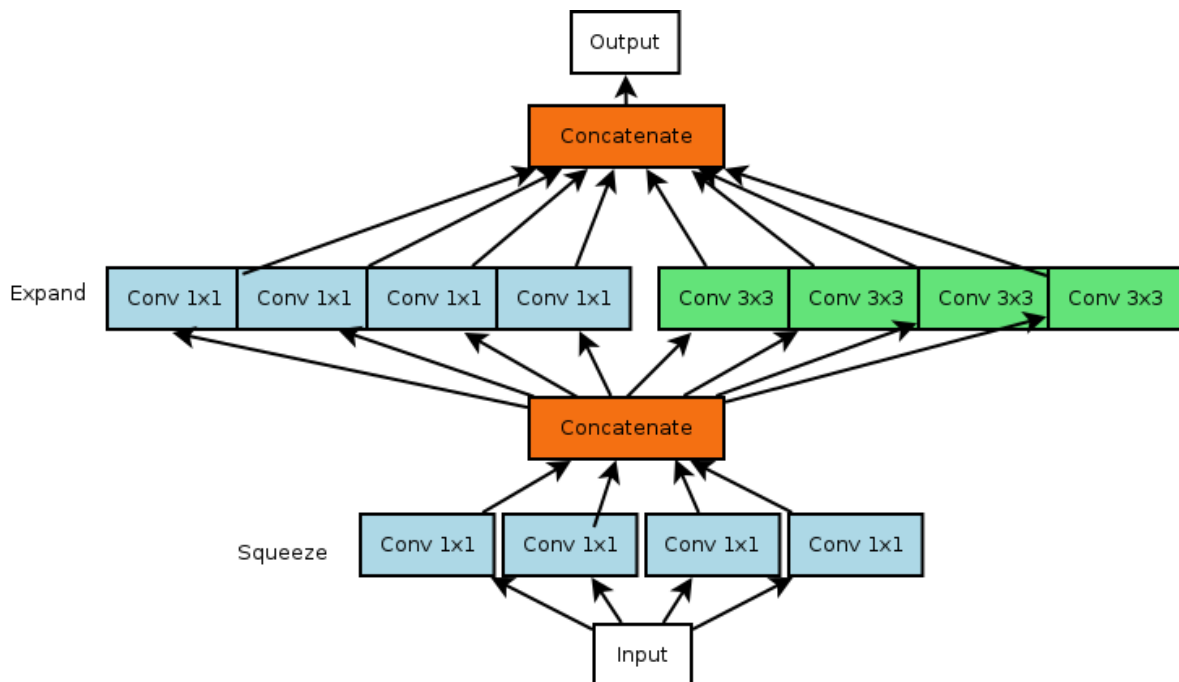


Fig 3.6 Architecture of SqueezeNet

3.4 CLASSIFICATION OF HERBAL IMAGES USING RESNET50

Any function can be represented with a single layer in a feedforward network. But it involves a large number of neurons in that layer and hence it may result in overfitting the data. Hence more and more layers are used in CNN. However, the weight gradient becomes smaller due to backpropagation of the error and hence repeated multiplication. It results in saturation of the results. Hence in Resnet in order to increase the performance one or more layers are skipped also residual mapping is fitted instead of fitting the desired mapping.

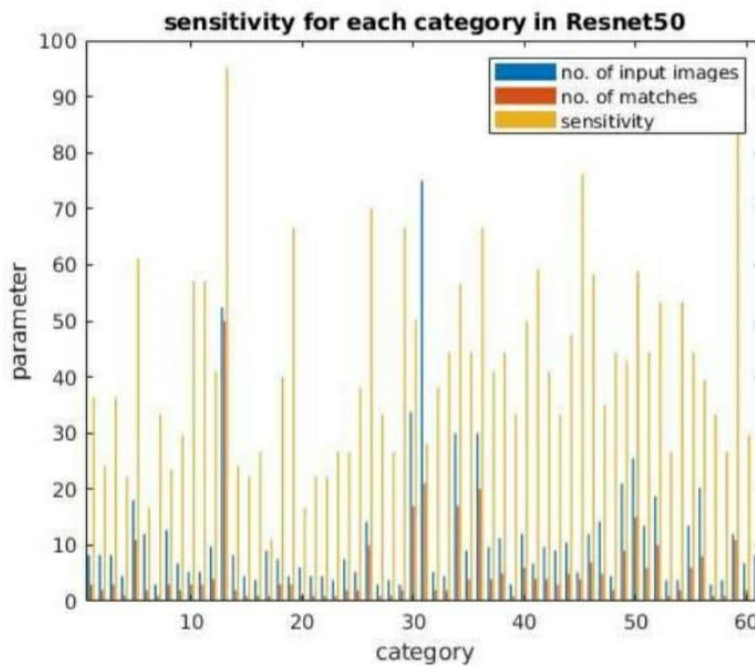


Fig 3.7 Performance of Resnet50 in terms of sensitivity

In resnet50 also, 50% of the images are used for training and 50% is used for testing. Then the network is used for determining the class of the key image. Overall accuracy of resnet50 is 47.1%.

It is found that GoogleNet works well only for herbal images that have high intraclass variance and less interclass variance. Squeezenet works well for large number of groups. On the other hand, Resnet50 works well for all groups. It is evident from the high sensitivity and accuracy of Resnet50 for all the groups of herbal images.

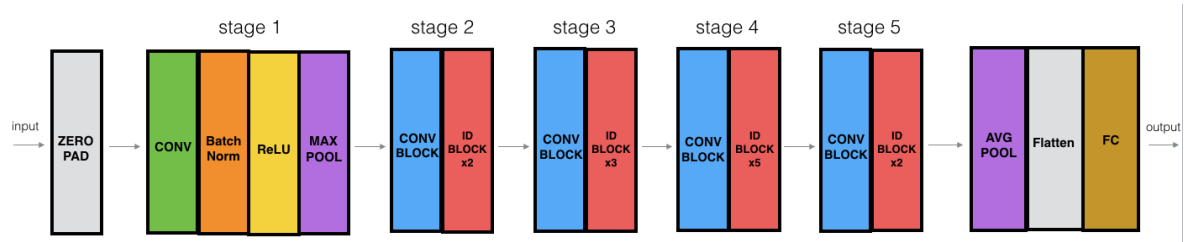


Fig 3.8 Architecture of RESNET50

CHAPTER 4

PROPOSED WORK

In Convolutional Modified ALEXNET Architecture, these are the modifications done; we have made the batch size as 10, maximum number of epochs to 5, initial learning rate as 10-5 and we will be changing it later on, we will be shuffling every epoch.

Learning rate:

The learning rate is a tuning parameter that determines the amount of change in the model. Selecting learning rate in correct manner is very important for the model. Optimization completely depends on learning rate. Learning rate should not be too low or too high, it must be moderate enough for the model.

Number of epochs

The number of epochs describes the number events that the learning estimation will work through the entire planning dataset. An epoch is contained at any rate one groups. The number of epochs is generally tremendous, as often as possible 100's or 1000's, allowing the learning estimation to run until the error from the network has been satisfactorily restricted. It is totally expected to make line plots that show epochs along the x-pivot as time and the error or capacity of the model on the y-centre point. These plots are sometimes called assumptions to learn and adjust. These plots can help with diagnosing whether the model has over learned, under insightful, or is suitably fit to the arrangement dataset.

Bias Vector:

A bias vector is an extra arrangement of loads in a neural organization that require no information, and this it relates to the yield of a fake neural organization when it has zero info. Predisposition addresses an additional neuron included with each pre-yield layer and stores the estimation of "1," for each activity. Predisposition units aren't attached to any past layer in the organization, so they don't address any type of movement, however are dealt with equivalent to some other weight. We have utilized the Adam Optimization Algorithm in this undertaking.

The Adam Optimization calculation is an expansion to probabilistic inclination plummet that has as of late seen more extensive reception for profound realizing

which can be utilized rather than the old-style stochastic angle plunge technique to refresh network loads iterative situated in preparing information.

Advantages of utilizing Adam are:

- I. Direct to carry out.
- II. Computationally effective.
- III. Little memory required.
- IV. Well suited for problem that are large in terms of data.

Working of Adam:

Adam is different to old style stochastic point dive. Stochastic incline drop keeps a singular learning rate for all weight revives and the learning rate doesn't change during planning. A learning rate is kept up for every association weight (limit) and freely changed as learning spreads out. Specifically, the computation calculates an amazing moving typical of the tendency and the squared incline, and the limits β_1 and β_2 control the decay speeds of these moving midpoints.

CHAPTER 5

RESULT

Table 5.1 Confusion matrix of ALEXNET with learning rate of 10^{-4} .

	class 1	class 2	class 3	class 4	class 5	class 6	class 7	class 8	class 9	class 10	class 11	class 12	class 13	class 14
class 1	2	8	0	0	1	1	0	0	0	0	0	0	0	0
class 2	4	17	1	2	4	3	0	0	4	0	0	0	1	1
class 3	2	3	4	2	0	3	0	0	1	0	1	1	4	0
class 4	0	9	4	8	4	1	0	0	3	1	5	1	3	0
class 5	0	4	1	5	4	0	0	0	1	1	1	0	2	0
class 6	0	6	2	1	0	7	1	0	0	0	0	0	1	1
class 7	0	2	0	0	2	3	0	1	0	0	0	0	0	0
class 8	2	5	0	1	1	1	0	0	0	0	1	0	0	0
class 9	2	6	0	1	0	1	0	0	0	0	2	1	0	0
class 10	1	7	0	0	2	1	0	2	2	0	1	0	1	0
class 11	2	2	0	0	0	1	0	0	0	1	3	0	0	0
class 12	0	1	0	2	0	0	0	0	0	0	2	2	3	3
class 13	0	3	1	2	0	0	0	0	0	0	4	0	3	0
class 14	0	0	0	0	0	2	0	0	0	0	1	0	0	8

The highlighted one indicates the correctly identified image of the respective class.

Table 5.2 Confusion matrix of ALEXNET with learning rate of 10^{-7} .

	class 1	class 2	class 3	class 4	class 5	class 6	class 7	class 8	class 9	class 10	class 11	class 12	class 13	class 14
class 1	0	1	0	3	1	1	2	0	0	0	0	0	4	0
class 2	3	7	1	10	0	3	0	0	0	2	0	3	6	2
class 3	0	4	0	6	0	1	1	0	0	2	0	3	2	2
class 4	0	8	1	12	3	2	0	0	0	4	0	4	2	3
class 5	0	9	0	2	0	1	1	1	0	1	0	3	1	0
class 6	0	3	1	4	1	1	0	0	0	1	0	1	3	4
class 7	0	1	0	0	1	1	0	0	1	0	1	0	2	1
class 8	0	1	0	4	0	0	0	0	0	0	0	1	5	0
class 9	0	4	0	4	0	0	0	0	0	0	0	2	3	0
class 10	0	3	2	5	2	0	0	0	0	1	0	2	1	1
class 11	0	4	1	2	0	0	0	0	0	0	0	1	0	1
class 12	0	3	0	5	0	0	0	0	0	0	0	5	0	0
class 13	0	4	0	3	1	0	0	0	1	1	0	2	1	0
class 14	1	0	0	2	2	1	1	0	0	2	0	0	1	1

The highlighted one indicates the correctly identified images of the respective class.

Table 5.3 Confusion matrix of ALEXNET with learning rate of 10^{-5} .

	class 1	class 2	class 3	class 4	class 5	class 6	class 7	class 8	class 9	class 10	class 11	class 12	class 13	class 14
class 1	1	6	0	3	0	1	0	0	0	1	0	0	0	0
class 2	2	13	3	7	3	1	0	0	2	0	0	0	6	0
class 3	0	1	3	5	5	4	0	0	1	2	0	0	0	0
class 4	0	8	1	15	6	3	0	1	1	3	0	1	0	0
class 5	0	5	2	4	5	1	0	0	0	1	0	0	1	0
class 6	0	5	1	4	2	4	0	0	1	0	1	1	0	0
class 7	1	4	0	1	0	1	0	0	0	0	0	0	1	0
class 8	0	1	1	2	1	1	2	1	1	0	0	0	1	0
class 9	0	3	1	4	2	0	0	0	1	1	0	1	0	0
class 10	1	8	2	1	0	2	0	0	1	0	0	0	2	0
class 11	0	3	1	3	2	0	0	0	0	0	0	0	0	0
class 12	0	2	2	6	0	0	0	1	0	1	0	0	0	1
class 13	0	2	3	3	0	0	0	0	0	2	0	1	2	0
class 14	0	2	2	1	0	3	1	0	0	0	0	0	0	2

The highlighted one indicates the correctly identified image of the respective class.

Table 5.4 Confusion matrix of ALEXNET with Bias value 5.

	class 1	class 2	class 3	class 4	class 5	class 6	class 7	class 8	class 9	class 10	class 11	class 12	class 13	class 14
class 1	0	4	0	4	3	0	0	0	1	0	0	0	0	0
class 2	0	6	0	16	3	0	0	3	3	3	0	0	3	0
class 3	0	4	0	8	3	0	0	2	0	0	2	1	0	1
class 4	1	7	0	13	3	0	0	2	4	5	3	0	1	0
class 5	0	2	0	12	2	0	0	1	0	1	0	0	1	0
class 6	1	0	0	5	4	0	0	1	1	1	1	4	1	0
class 7	0	0	1	0	3	0	0	0	2	0	1	0	1	0
class 8	1	2	0	2	1	1	0	1	2	0	1	0	0	0
class 9	0	1	0	6	0	0	0	0	1	1	0	1	3	0
class 10	0	2	0	6	2	0	0	1	1	3	1	1	0	0
class 11	0	1	0	4	2	0	0	1	1	0	0	0	0	0
class 12	0	0	0	3	2	0	0	0	1	2	1	4	0	0
class 13	0	1	0	2	0	0	0	0	0	6	3	1	0	0
class 14	0	0	0	2	2	0	0	0	1	1	1	3	0	1

The highlighted one indicates the correctly identified image of the respective class.

Table 5.5 Confusion matrix of ALEXNET with learning rate of 10^{-4} and number of epochs is 5.

	class 1	class 2	class 3	class 4	class 5	class 6	class 7	class 8	class 9	class 10	class 11	class 12	class 13	class 14
class 1	2	2	0	5	1	1	0	0	0	0	1	0	0	0
class 2	2	7	0	6	6	3	0	2	5	0	3	0	1	2
class 3	1	0	0	6	2	1	1	4	0	1	2	2	0	1
class 4	1	2	1	13	8	0	4	2	1	1	3	2	0	1
class 5	2	1	0	4	8	0	0	1	1	0	2	0	0	0
class 6	0	2	0	0	0	5	7	1	1	1	0	1	0	1
class 7	0	0	0	0	0	1	5	2	0	0	0	0	0	0
class 8	1	4	0	3	1	0	0	0	0	0	2	0	0	0
class 9	3	1	0	2	0	1	1	1	1	0	3	0	0	0
class 10	0	1	0	4	5	0	0	1	3	0	2	1	0	0
class 11	0	0	1	3	1	1	0	0	0	0	3	0	0	0
class 12	0	0	0	1	0	0	0	0	1	0	5	3	0	3
class 13	0	3	1	1	0	0	0	0	0	0	6	0	1	1
class 14	0	0	0	0	0	1	1	0	0	0	0	0	0	9

The highlighted one indicates the correctly identified image of the respective class.

Table 5.6 Confusion matrix of ALEXNET with learning rate of 10^{-5} and number of epochs is 5.

	clas s 1	clas s 2	clas s 3	clas s 4	clas s 5	clas s 6	clas s 7	clas s 8	clas s 9	clas s 10	clas s 11	clas s 12	clas s 13	clas s 14
Clas s 1	1	5	0	4	1	1	0	0	0	0	0	0	0	0
clas s 2	0	21	0	9	4	0	0	2	1	0	0	0	2	0
clas s 3	1	5	3	6	0	1	0	1	0	0	0	0	0	2
clas s 4	0	11	4	12	2	1	0	2	1	2	0	3	0	1
clas s 5	0	5	1	8	5	0	0	0	0	0	0	0	0	0
clas s 6	0	4	2	5	0	3	3	0	1	0	0	0	0	1
clas s 7	1	1	0	0	1	0	1	0	1	0	1	0	0	2
clas s 8	1	4	0	3	1	0	1	1	0	0	0	0	0	0
clas s 9	1	3	0	5	0	1	0	1	1	1	0	0	0	0
clas s 10	0	6	0	8	1	0	0	1	0	0	0	0	0	0
clas s 11	0	2	1	3	0	1	1	0	0	0	0	0	0	1
clas s 12	0	3	0	6	0	2	0	0	1	1	0	0	0	0
clas s 13	0	4	1	1	0	1	0	0	1	2	1	0	2	0
clas s 14	0	1	1	0	0	1	0	0	0	0	0	1	0	7

The highlighted one indicates the correctly identified image of the respective class.

Table 5.7 Efficiency of the neural networks with respect to the changed parameters.

	10 ⁻⁴	10 ⁻⁷	10 ⁻⁵	BIAS=5	EPOCHS=5, 10 ⁻⁴	EPOCHS=5, 10 ⁻⁵
class #1	16.67	0	8.333333	0	16.66667	8.333333
class #2	45.94595	18.91892	35.13514	16.21622	18.91892	56.75676
class #3	19.04762	0	14.28571	0	0	14.28571
class #4	20.51282	30.76923	38.46154	33.33333	33.33333	30.76923
class #5	21.05263	0	26.31579	10.52632	42.10526	26.31579
class #6	36.84211	5.263158	21.05263	0	26.31579	15.78947
class #7	0	0	0	0	62.5	12.5
class #8	0	0	9.090909	9.090909	0	9.090909
class #9	0	0	7.692308	7.692308	7.692308	7.692308
class #10	0	5.882353	0	17.64706	0	0
class #11	33.33333	0	0	0	33.33333	0
class #12	15.38462	38.46154	0	30.76923	23.07692	0
class #13	23.07692	7.692308	15.38462	0	7.692308	15.38462
class #14	72.72	9.090909	18.18182	9.090909	81.81818	63.63636

The highlighted indicates the highest efficiency of the respective class.

5.1 CONCLUSION

From the above table we can portray that for the arrangement of class #1 pictures, we need to change the learning rate of the neural organization to 10⁻⁴ or number of epochs to 5 with 10⁻⁴ learning rate. For the arrangement of class #2 pictures, we need to change the quantity of epochs to 5 and learning rate of the neural organization to 10⁻⁵. For the characterization of class #3 pictures, we need to change the learning rate of the neural organization to 10⁻⁴. For the order of class #4 pictures, we need to change the learning rate of the neural organization to 10⁻⁵. For the arrangement of class #5 pictures, we need to change the quantity of epochs to 5 and learning rate of the neural organization to 10⁻⁵. For the characterization of class #6 pictures, we need to change the learning rate of the neural organization to 10⁻⁴. For the order of class #7 pictures, we need to change the quantity of epochs to 5 and learning rate of the neural organization to 10⁻⁵. For the grouping of class #8 pictures learning rate can be 10⁻⁵ or bias to 5 or number of epochs equivalents to 5 with 10⁻⁵ learning rate. For the grouping of class #9 pictures learning rate can be 10⁻⁵ or bias to 5 or number of epochs equivalents to 5 with 10⁻⁴ learning rate or

number of epochs equivalents to 5 with 10^{-5} learning rate. For the order of class #10 pictures, we need to change the Bias of the neural organization to 5. From the above table we can portray that for the characterization of class #11 pictures, we need to change the learning rate of the neural organization to 10^{-4} or number of epochs to 5 and learning rate can be 10^{-4} . For the characterization of class #12 pictures, we need to change the learning rate of the neural organization to 10^{-7} . For the arrangement of class #13 pictures, we need to change the learning rate of the neural organization to 10^{-4} . For the arrangement of class #14 pictures, we need to change the quantity of epochs equivalents to 5 with 10^{-4} learning rate. By changing the previously mentioned boundaries regarding herbal grown plants, the efficiency for the arrangement of natural pictures can be expanded. This undertaking has halted at recreation level, equipment execution of this work should be possible in future too. Plausibility of other neural networks for the arrangement herbal grown pictures can be contemplated.

5.2 FUTURE WORK:

This project is stopped at simulation level. Hardware implementation of this work can be done in future as well. Feasibility of various neural networks for classification of plants can also be checked in future.

REFERENCES:

- [1] Evaluation of CNN, Alexnet and GoogleNet for Fruit Recognition Nur Azida Muhammad, Amelina Ab Nasir, Zaidah Ibrahim, Nurbaity Sabri [2018].
- [2] Weed Seedling Detection Using Mask Regional Convolutional Neural Network Sanjay Patidar, Utkarsh Singh, Sumit Kumar Sharma, Himanshu [2020].
- [3] Residual Component Estimating CNN for Image Super-Resolution Xian-Hua Han, YongQing Sun, Yen-Wei Chen [2019].
- [4] Optimum CNN-Based Plant Mutant Classification Yeh Huann Goh, Chee Ho Ng, Yoon Ket Lee, Choe Yung Teoh [2018].
- [5] Mobile Leaf Identification System using CNN applied to plants in Hokkaido Tatsuhiro Akiyama, Yosuke Kobayashi, Yuichi Sasaki, Kaori Sasaki [2019].
- [6] Implementation of Faster R-CNN in Paddy Plant Disease Recognition System Shadman Mahmood Khan Pathan Md. Firoj Ali [2019].
- [7] Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm Melike Sardogan, Adem Tuncer, Yunus Ozen [2018].
- [8] Vietnamese Herbal Plant Recognition Using Deep Convolutional Features Anh Vo, Bao Nguyen Thien [2019].

batch 66

by Plagiarism Checking

Submission date: 16-Apr-2021 12:29PM (UTC-0400)

Submission ID: 1561094022

File name: Report_batch_66_1_6_1.docx (1.71M)

Word count: 4414

Character count: 23441

CHAPTER 1

INTRODUCTION

India is a country noted for its rich cultural heritage and medicinal values. Right from ancient days, medicinal herbs have become an integral part in every household. Even today diseases like common cold, asthma and body pain are cured with herbs and herbal medicines. A wide variety of herbs with medicinal values are still prevalent in various parts of India especially in hills and mountains. It has attracted a large number of researchers for pursuing their research work. Also, when these varieties of herbs when categorized and digitized would be of great support to the field of medicine. In recent days, multiple researchers are working to bridge the gap between modern medicine and herbal medicine. Considerable research has been carried out in this area for automated characterization of anomaly/abnormality/flowers/herbs. Selvarasu et al used Euclidean based color image segmentation for the extraction of abnormal features. Conventionally features are extracted and these features are fed to the classifiers. Conventional classification techniques involve the following steps: Features are extracted from the input images, exemplars are created with the features as input parameters and the type of the herb as the output parameter, exemplars are divided into two sets one for training and the other for testing, neural network is trained and tested, trained neural network and its weights are stored. Having done this, any input image of the herb (to be classified) is considered. Features are extracted and these features are fed to the already trained neural network and the output is obtained. The output indicates the type of the herb. In the above case, performance of the classifiers is strongly dependent on the extracted features. Hence an efficient feature extraction technique has to be identified. Though various image segmentation techniques are available in literature these techniques do not work well for real time images. Computational complexity is more as the features are extracted with an image segmentation technique followed by feature representation. Hence it is necessary to develop a classifier that does not involve feature extraction but performs automated classification with higher efficiency.

1.1 LITERATURE SURVEY

S. No.	Title	Authors	Content
1.	¹² Evaluation of Convolutional neural network, Alex net and ⁸ Google Net for fruit recognition	Nur Azida Muhammad, Amelina A Nasir, Zaidah Ibrahim, Nurbaiti Sabri	The exhibition of essential Convolutional Neural Network (CNN), Alexnet and Googlenet in perceiving nine unique kinds of organic products from a freely accessible dataset.
2.	Weed Seedling Detection Using Mask R-CNN	Sanjay Patidar, Utkarsh Singh, Sumit Kumar Sharm, Himanshu ¹⁰	Mask R-CNN will assist the individual with isolating weed from essential yield so that harvest will get full sustenance and production gets expanded.
3.	Mutant classification using Convolutional neural network	Yeh Huann Goh Chee Ho Ng Yoon Ket Lee	The perceptible characteristics of freaks of a comparative genotype plant communicating with various biological conditions is basic to perceive how well the introduction of a particular property in different improvement environment. It Intends to assemble an ideal cnn for picture-based plant freak order task. ⁶
4.	Mobile leaf identification using Convolutional neural network	Kaori Sasaki Yuichi Sasaki, Yosuke Kobayashi, Tatsuhiro Akiyama	A Mobile application using convolutional neural networks (CNNs) that will help beginners identify plant species.
5.	Plant Leaf disease	Melike Sardogan, Adem Tuncer	Programmed techniques for arrangement of plant illnesses additionally help making a

	9 Detection and Classification based on CNN With LVQ algorithm 7	Yunus Ozen	move in the wake of recognizing the side effects of leaf diseases. A CNN and LVQ algorithm are used for the detection of infection on leaves and also classifies them.
6.	Deep Residual Learning for Image Recognition	Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun 11	Development of residual learning technique framework for efficient Image Recognition.
7.	Herbal Plant Recognition	Anh H. Vo, Hoa T. Dang, Bao T.	Development of Novel Neural networks for recognition of herbal plants for medicinal uses and business purposes.
8.	Performance Evaluation of Googlenet, SqueezeNet, and Resnet50 in the Classification of Herbal Images	M.A. Muthiah, E.Logashamugam, N.M. Nandhitha	Computer aided classification of medicinal herbs is of major concern in medicinal research. Real challenge lies in the complexity and variability of plants belonging to the same species. Convolutional Neural Networks are the recently accepted paradigm for classification with the help of pretrained neural networks. GoogleNet, SqueezeNet, Resnet 50 for the classification of medicinal herbs is studied.

CHAPTER 2

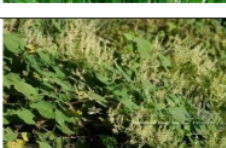
RESEARCH DATABASE AND SOFTWARE REQUIREMENTS

We have got the database images from various sources. Database consists of total 14 categories. Each category contains minimum of 8 and maximum of 30 images. Total database consists of 489 images. We have divided the total database into two halves. Half of the pictures are utilized for testing and half of the pictures are utilized for training. The below table contains names of 14 categories and one image from that category.

S.No.	Image Name	Image
1.	Apiaceae	
2.	Asteraceae	
3.	Euphorbiaceae	
4.	Fabaceae	
5.	Lamiaceae	
6.	Liliaceae	
7.	Poaceae	
8.	Polygonaceae	

9.	Ranunculaceae	
10.	Rosaceae	
11.	Rubiaceae	
12.	Rutaceae	
13.	Solanaceae	
14.	Zingiberaceae	

Table 2.1 Database Training Images

S.No.	Image Name	Image
1.	Apiaceae	
2.	Asteraceae	
3.	Euphorbiaceae	
4.	Fabaceae	
5.	Lamiaceae	
6.	Liliaceae	
7.	Poaceae	
8.	Polygonaceae	

9.	Ranunculaceae	
10.	Rosaceae	
11.	Rubiaceae	
12.	Rutaceae	
13.	Solanaceae	
14.	Zingiberaceae	

Table 2.2 Database Testing Images

We will be using MATLAB R2020B along with Deep Learning Tool Box.

2.1 MATLAB SOFTWARE DETAILS

MATLAB is a superior language for particular processing. Matlab facilitates computation, discernment, and coding in an ease to-use environment where issues and plans are conveyed in unmistakable documentation.

Typical utilizations include:

1. Estimation and math
2. Estimation headway
3. Illustrating, re-enactment, and prototyping
4. Data assessment, examination, and insight
5. Sensible and planning outlines
6. Application headway, including GUI building
7. An instinctive numerical figuring environment
8. System estimations

Designs

Programming (M-records)

Tool compartments (signal preparing, insights, advancement, emblematic math)

Matlab (MATRIX laboratory) is a mathematical figuring climate and fourth generation programming language.

Matlab with Simulink building a powerful model can really be enjoyable. It is additionally incredible enough to accomplish genuine work, when you gain proficiency with a portion of its high-level highlights (i.e., it's anything but an exercise in futility to learn). Truth be told, we will just utilize a portion of the somewhat fundamental highlights of Simulink for the models we fabricate, and you are free (and empowered) to investigate some different highlights that will assist you with issues you need to settle. Fundamentally you can think about a Simulink model as like depicting a wiring chart of your framework, where the factors stream on wires and are changed, or changed, by different gadgets. We should assemble a Simulink model for the direct spring-mass framework with gooey damping: First, you need to begin the tangle lab application, which you do in Unix by composing

"Matlab" at the shell brief, or in Windows by double tapping (quickly tapping the left mouse button twice) on the Matlab symbol, which you will discover covered some place with a lot of different symbols. This will get you to the ">>" Matlab brief. Type "Simulink" to begin Simulink, and raise the primary Simulink window (your windows may look fairly changed, contingent upon the machine and working framework): Let's go through the essential elements of this window first.

On top is a rundown of things that are related with pull-down menus. These will be copied on your model workspace (see underneath), so generally they are not very significant here. Except for the "document" menu, which you will use to make another model workspace or to raise a record program to get to a current (saved) model. Every one of the square symbols relate to a class of changes that will be utilized to depict the unique framework. You will for the most part utilize the changes depicted in "Sources," "Sinks," "Straight," "Nonlinear," and "Associations." The "Sources" symbol contains a set-up of sign generators, like irregular number generators and wave (sinusoidal, square, and so forth) generators, that can be utilized to give time-fluctuating sources of info that drive your dynamic framework (these are exogenous information sources, or outside inputs, implying that their elements are portrayed remotely to the model). The "Sinks" classification will for the most part be utilized to get to the "Degree" and XY charts to which you will associate the framework "wires" to get a "readout." The "Straight" class contains every one of the potential approaches to change a variable directly, like various sorts of gains. It additionally has the essential "integrator" block that we will utilize widely to demonstrate our first-request dynamic frameworks.

Correction and end debugging

While investigating, we can modify the estimation of a variable to check whether the new worth produces anticipated outcomes. While the code is stopped, dole out another worth to the alphabet in the command window, workspace program. At that point keep running and venturing through the program.

Ending debugging

In the wake of recognizing an issue, end the investigating meeting. It is ideal to stop investigate mode prior to altering a M-record. Else, we got startling outcomes

when we are running the record. To stop troubleshooting, debugging mode from debugging menu should be selected.

Merits of Matlab

End of mistakes right off the bat in the plan

Powerful control framework

Less emphasis in the improvement cycle, diminished advancement time and cost, and so forth

In addition, these instruments can be coordinated with constant equipment which means incorporating conventional disconnected re-enactment with true testing.

An apparently improved on method that absolves an originator from encoding his numerical model in the programming language embraced,

A chance of a various utilization of the essential numerical models, assembled at the libraries and joined on a case-by-case basis.

2.1.1 APPLICATIONS OF MATLAB:

Numerical Calculations

Data Analysis and Visualization

Software Development

Simulation

2.1.2 ADVANTAGES:

It's exceptionally simple for a beginner in PC programming.

It accompanies well-written manuals.

Huge client local area, sharing free codes. It allows you to work in Matrices without any problem. It allows you to do some convoluted computations rapidly, without composing an entire program or accounting page. Huge loads of additional items and workbenches accessible to do an assortment of errands.

Simple, intuitive environment

Quick mathematical calculations

Modest programming

Inexpensive software

Fun and simple climate for learning Linear Algebra!!

CHAPTER 3

EXISTING WORK FOR THE CLASSIFICATION OF HERBAL IMAGES

This chapter provides an overview of the existing GOOGLNET, SQUEEZENET AND RESNET50 architectures. Performance of these networks in classifying the potatoes from thermal images is also provided. Performance is measured in terms of sensitivity and accuracy.

3.1 CONVOLUTION NEURAL NETWORK – AN OVERVIEW

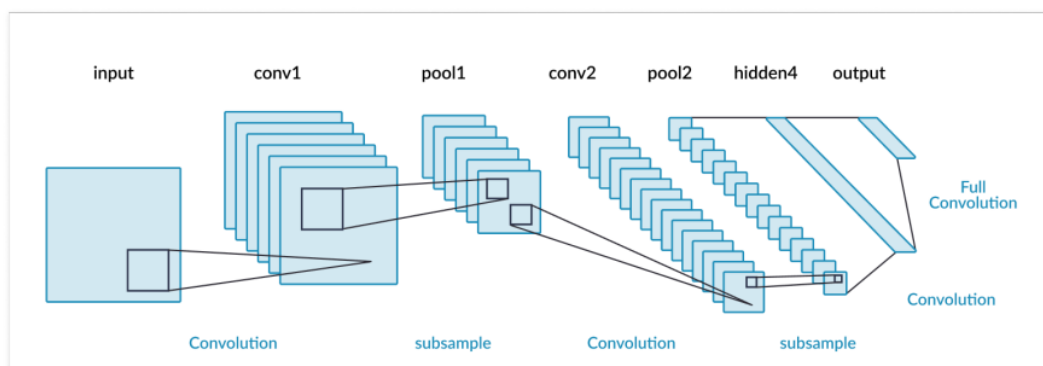


Fig.3.1 Block diagram of CNN

CNN is a type of Deep Neural Networks which can perceive, characterize specific highlights from pictures and are generally utilized for investigating visual pictures. Their applications range from picture and video acknowledgment, picture characterization, clinical picture investigation, PC vision and regular language preparing. The expression "Convolution" in CNN means the numerical capacity of convolution which is an exceptional sort of direct activity wherein two capacities are increased to create a third capacity which communicates how the state of one capacity is changed by the other. In straightforward terms, two pictures which can be addressed as grids are increased to give a yield that is utilized to extricate highlights from the picture.

There are 2 major parts CNN design:

A convolution tool that isolates and recognizes the different highlights of the picture for examination in an interaction called as Feature Extraction

A completely associated layer that uses the yield from the convolution interaction and predicts the class of the picture dependent on the highlights extricated in past stages.

Layers of Convolutional Neural network are:

1. Convolved Layer:

It is the important layer that is used for removing different highlights from the information pictures. In this layer, the numerical activity of convolution is performed between the info picture and a channel of a specific size $M \times M$. By sliding the channel over the info picture, the dab item is taken between the channel and the pieces of the information picture concerning the size of the channel ($M \times M$).

The yield is named as the Feature map which gives us data about the picture like the corners and edges. Afterward, this component map is taken care of to different layers to gain proficiency with a few different highlights of the info picture.

2. Strides

Stride helps in moving the channel grid over the pixels in the component network. On the off chance that the step size is 1, we can move the channel to 1 pixel at a time. On the off chance that the info grid measurement is $n \times n$, channel measurement is $f \times f$ and step size $= s$ then the yield size will be equivalent to $((n-f/s) + 1) \times ((n-f/s) + 1)$. In the event that we increment the step size to a higher worth, there is no yield so we can utilize cushioning for it.

3. Padding

Padding helps in making the output and input size equal.

Reasons to use padding:

To avoid shrinking of output image if we increase the number of convolution layers. To make use of the outer edge along with the inner edge of the input image while doing convolution. If padding $p=1$ then it adds two rows and two columns for

the input matrix. Mostly we use zero padding. In zero padding we assign zeros to the newly formed rows and columns.

Types of Padding:

Valid padding: In valid padding there is no padding involved. So, the dimensions of input and output are different.

Same padding: Same padding helps in making the dimensions of convolution output dimension and input dimension equal.

4. Pooling Layer

- It is used to reduce the dimensions of the feature map.
- It is used to extract the dominant features.
- Strides and padding are also available for pooling.
- There are two types of pooling:
 - Max pooling.
 - Average pooling.
 - Max pooling selects the maximum value in that respective 2D matrix.
 - Average pooling takes the average of values in that respective 2D matrix.
- Max pooling is better than average pooling.

5. ACTIVATION FUNCTION

Activation functions are conditions that decide the output of a neural network. It helps in normalizing the output of every neuron to a range between [1 and 0] (or) between [-1 and 1]. ReLU is the most usually utilized enactment work in any neural organization. The essential condition of ReLU actuation work is $\max(y, 0)$, where y is the particular tangled yield framework component.

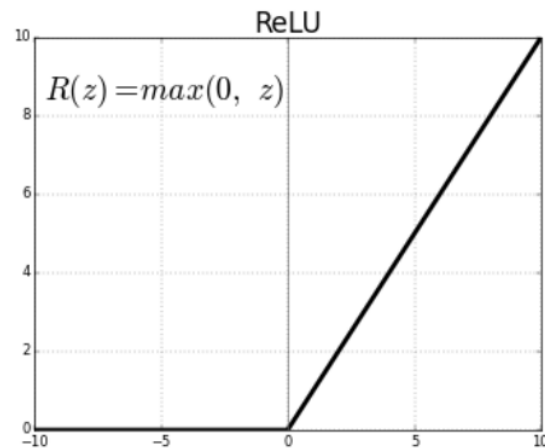


Fig.3.2 Gradient of a ReLU activation function

This graph implicates that the negative values in the feature matrix is made to 0. So, the values range from 0 to 1.

6. SOFTMAX

SoftMax gives us the last likelihood for each mark and allots decimal probabilities to each class in a multi-class issue. It is likewise a kind of Activation Function. Information esteems can be negative, zero, positive, or more noteworthy than one however the yield esteems will be somewhere in the range of 0 and 1, so they can be deciphered as probabilities.

7. FLATTENING

A Fully connected output layer takes the yield of the past layers 'flattens' them and transforms them into a single vector. Flattening is the primary benefit of CNN over Feed-Forward Neural Networks.

8. Fully Connected Layer (FC):

Full Connected (FC) layer comprises loads & inclinations alongside the neuron it utilized interface the neurons among 2 distinct layers. The layers are ordinarily situated before output layer & design several layers of a CNN Architecture.

3.2 CLASSIFICATION OF HERBAL IMAGES USING GOOGLNET

The GoogleNet architecture comprises of 22 layers (27 layers including pooling layers), and a piece of these layers are an aggregate of 9 inception modules.

The information layer of the Google Net engineering takes in a picture of the measurement 224 x 224.

Type: This alludes to the name of the current layer of the part inside the architecture

Patch Size: Refers to the size of the broad window used across convolution and pooling layers. Sweeping windows have equivalent height and width.

Stride: Defines the measure of move the channel/sliding window assumes control over the information picture.

Output Size: The subsequent yield measurements (height, width, number of feature maps) of the current design part after the information is gone through the layer.

Depth: Refer to the quantity of levels/layers inside an architecture part.

#1x1 #3x3 #5x5: Refers to the different convolution's channels utilized inside the inception module.

#3X3 reduce #5x5 reduce: Refers to the quantities of 1x1 channels utilized before the convolutions.

Pool Proj: This is the quantity of 1x1 channels utilized in the wake of pooling inside an inception module.

Params: Refers to the quantity of loads inside the current architecture segment.

Operations: Refers to the quantity of numerical activities did inside the segment.

In this progression, the pretrained GoogleNet architecture is stacked and is prepared and tried with two unique arrangements of pictures. At that point the architecture is finalized and the architecture is prepared for use for the given picture database.

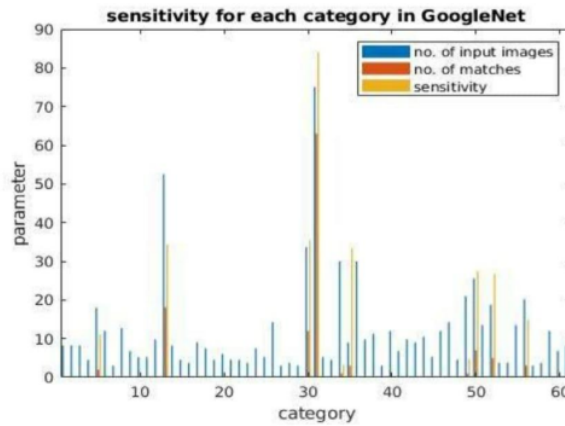


Figure 3.3: Performance of GoogleNet in terms of sensitivity

Half of pictures are utilized for preparing and half of pictures are utilized for testing. At that point the organization is utilized for deciding the class of the keyimage. In general exactness of GoogleNet is 16.5%

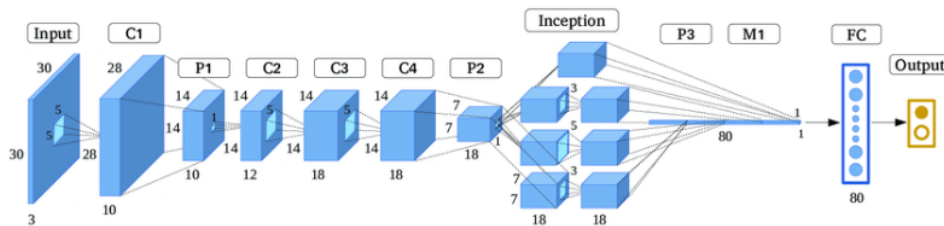


Figure 3.4: Architecture of GoogleNet

3.3 CLASSIFICATION OF HERBAL IMAGES USING SQUEEZENET

Having understood the feasibility of convolutional Neural Networks, the next step is to develop FPGA for the pretrained networks. In such case, smaller the size, easier is its implementation. Such a smaller CNN is SqueezeNet, where major modifications are performed. A major number of 3x3 filters in convolution layer are replaced with 1x1 filters without affecting the accuracy of operation. Number of

inputs to each filter is also reduced. A module called fire module is developed which has a convolution layer (1x1 filter) and an expand layer (that has both 1x1 and 3x3 filters). Also, SqueezeNet uses maxpooling technique.

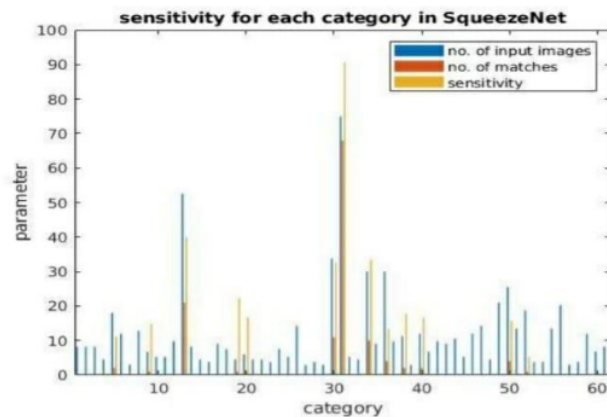


Figure 3.5: Performance of SqueezeNet in terms of sensitivity

In Squeezenet also, 50% of the images are used for training and 50% is used for testing. Then the network is used for determining the class of the key image. Overall accuracy of Squeezenet is 18.3%.

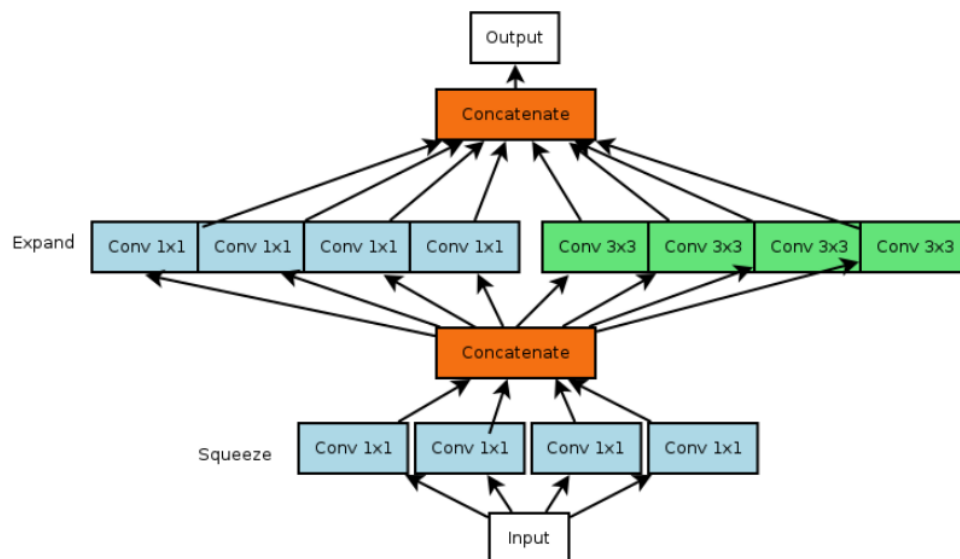


Figure 3.6: Architecture of SqueezeNet

3.4 CLASSIFICATION OF HERBAL IMAGES USING RESNET50

Any function can be represented with a single layer in a feedforward network. But it involves a large number of neurons in that layer and hence it may result in overfitting the data. Hence more and more layers are used in CNN. However, the weight gradient becomes smaller due to backpropagation of the error and hence repeated multiplication. It results in saturation of the results. Hence in Resnet in order to increase the performance one or more layers are skipped also residual mapping is fitted instead of fitting the desired mapping.

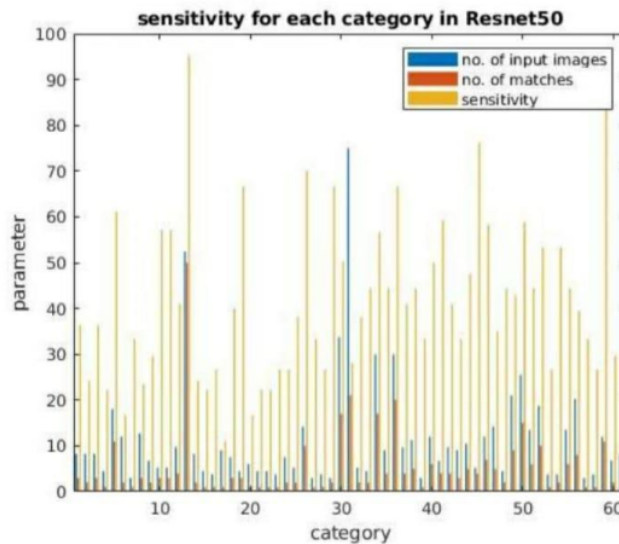


Figure 3.6: Performance of Resnet50 in terms of sensitivity

In resnet50 also, 50% of the images are used for training and 50% is used for testing. Then the network is used for determining the class of the key image. Overall accuracy of resnet50 is 47.1%.

It is found that GoogleNet works well only for herbal images that have high intraclass variance and less interclass variance. Squeezenet works well for large number of groups. On the other hand, Resnet50 works well for all groups. It is

evident from the high sensitivity and accuracy of Resnet50 for all the groups of herbal images.

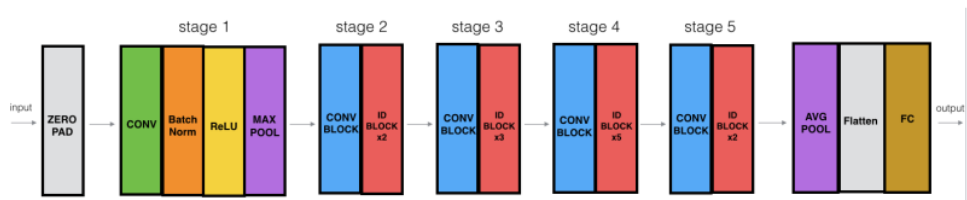


Figure 3.7: Architecture of RESNET50

CHAPTER 4

PROPOSED WORK

In Convolutional Modified ALEXNET Architecture, these are the modifications done; we have made the batch size as 10, maximum number of epochs to 5, initial learning rate as 10-5 and we will be changing it later on, we will be shuffling every epoch.

Learning rate:

The learning rate is a tuning parameter that determines the amount of change in the model. Selecting learning rate in correct manner is very important for the model. Optimization completely depends on learning rate. Learning rate should not be too low or too high, it must be moderate enough for the model.

Number of epochs

The number of epochs describes the number events that the learning estimation will work through the entire planning dataset. An epoch is contained at any rate one groups. The number of epochs is generally tremendous, as often as possible 100's or 1000's, allowing the learning estimation to run until the error from the network has been satisfactorily restricted. It is totally expected to make line plots that show epochs along the x-pivot as time and the error or capacity of the model on the y-centre point. These plots are sometimes called assumptions to learn and adjust. These plots can help with diagnosing whether the model has over learned, under insightful, or is suitably fit to the arrangement dataset.

Bias Vector:

A bias vector is an extra arrangement of loads in a neural organization that require no information, and this it relates to the yield of a fake neural organization when it has zero info. Predisposition addresses an additional neuron included with each pre-yield layer and stores the estimation of "1," for each activity. Predisposition units aren't attached to any past layer in the organization, so they don't address any type of movement, however are dealt with equivalent to some other weight. We have utilized the Adam Optimization Algorithm in this undertaking.

⁴ The Adam Optimization calculation is an expansion to probabilistic inclination plummet that has as of late seen more extensive reception for profound realizing

5
which can be utilized rather than the old-style stochastic angle plunge technique to refresh network loads iterative situated in preparing information.

Advantages of utilizing Adam are:

- I. Direct to carry out.
- II. Computationally effective.
- III. Little memory required.
- IV. Well suited for problem that are large in terms of data.

Working of Adam:

Adam is different to old style stochastic point dive. Stochastic incline drop keeps a singular learning rate for all weight revives and the learning rate doesn't change during planning. A learning rate is kept up for every association weight (limit) and freely changed as learning spreads out. Specifically, the computation calculates an amazing moving typical of the tendency and the squared incline, and the limits β_1 and β_2 control the decay speeds of these moving midpoints.

CHAPTER 5

RESULT

	class 1	class 2	class 3	class 4	class 5	class 6	class 7	class 8	class 9	class 10	class 11	class 12	class 13	class 14
class 1	2	8	0	0	1	1	0	0	0	0	0	0	0	0
class 2	4	17	1	2	4	3	0	0	4	0	0	0	1	1
class 3	2	3	4	2	0	3	0	0	1	0	1	1	4	0
class 4	0	9	4	8	4	1	0	0	3	1	5	1	3	0
class 5	0	4	1	5	4	0	0	0	1	1	1	0	2	0
class 6	0	6	2	1	0	7	1	0	0	0	0	0	1	1
class 7	0	2	0	0	2	3	0	1	0	0	0	0	0	0
class 8	2	5	0	1	1	1	0	0	0	0	1	0	0	0
class 9	2	6	0	1	0	1	0	0	0	0	2	1	0	0
class 10	1	7	0	0	2	1	0	2	2	0	1	0	1	0
class 11	2	2	0	0	0	1	0	0	0	1	3	0	0	0
class 12	0	1	0	2	0	0	0	0	0	0	2	2	3	3
class 13	0	3	1	2	0	0	0	0	0	0	4	0	3	0
class 14	0	0	0	0	0	2	0	0	0	0	1	0	0	8

TABLE 5.1: The learning rate for the neural network is 10^{-4} . The highlighted one indicate the correctly classified image.

	class 1	class 2	class 3	class 4	class 5	class 6	class 7	class 8	class 9	class 10	class 11	class 12	class 13	class 14
class 1	0	1	0	3	1	1	2	0	0	0	0	0	4	0
class 2	3	7	1	10	0	3	0	0	0	2	0	3	6	2
class 3	0	4	0	6	0	1	1	0	0	2	0	3	2	2
class 4	0	8	1	12	3	2	0	0	0	4	0	4	2	3
class 5	0	9	0	2	0	1	1	1	0	1	0	3	1	0
class 6	0	3	1	4	1	1	0	0	0	1	0	1	3	4
class 7	0	1	0	0	1	1	0	0	1	0	1	0	2	1
class 8	0	1	0	4	0	0	0	0	0	0	0	1	5	0
class 9	0	4	0	4	0	0	0	0	0	0	0	2	3	0
class 10	0	3	2	5	2	0	0	0	0	1	0	2	1	1
class 11	0	4	1	2	0	0	0	0	0	0	0	1	0	1
class 12	0	3	0	5	0	0	0	0	0	0	0	5	0	0
class 13	0	4	0	3	1	0	0	0	1	1	0	2	1	0
class 14	1	0	0	2	2	1	1	0	0	2	0	0	1	1

TABLE 5.2: The learning rate for the neural network is 10^{-7} . The highlighted one indicate the correctly classified image.

	class 1	class 2	class 3	class 4	class 5	class 6	class 7	class 8	class 9	class 10	class 11	class 12	class 13	class 14
class 1	1	6	0	3	0	1	0	0	0	1	0	0	0	0
class 2	2	13	3	7	3	1	0	0	2	0	0	0	6	0
class 3	0	1	3	5	5	4	0	0	1	2	0	0	0	0
class 4	0	8	1	15	6	3	0	1	1	3	0	1	0	0
class 5	0	5	2	4	5	1	0	0	0	1	0	0	1	0
class 6	0	5	1	4	2	4	0	0	1	0	1	1	0	0
class 7	1	4	0	1	0	1	0	0	0	0	0	0	1	0
class 8	0	1	1	2	1	1	2	1	1	0	0	0	1	0
class 9	0	3	1	4	2	0	0	0	1	1	0	1	0	0
class 10	1	8	2	1	0	2	0	0	1	0	0	0	2	0
class 11	0	3	1	3	2	0	0	0	0	0	0	0	0	0
class 12	0	2	2	6	0	0	0	1	0	1	0	0	0	1
class 13	0	2	3	3	0	0	0	0	0	2	0	1	2	0
class 14	0	2	2	1	0	3	1	0	0	0	0	0	0	2

TABLE 5.3: The learning rate for the neural network is 10^{-5} . The highlighted one indicate the correctly classified image.

	class 1	class 2	class 3	class 4	class 5	class 6	class 7	class 8	class 9	class 10	class 11	class 12	class 13	class 14
class 1	0	4	0	4	3	0	0	0	1	0	0	0	0	0
class 2	0	6	0	16	3	0	0	3	3	3	0	0	3	0
class 3	0	4	0	8	3	0	0	2	0	0	2	1	0	1
class 4	1	7	0	13	3	0	0	2	4	5	3	0	1	0
class 5	0	2	0	12	2	0	0	1	0	1	0	0	1	0
class 6	1	0	0	5	4	0	0	1	1	1	1	4	1	0
class 7	0	0	1	0	3	0	0	0	2	0	1	0	1	0
class 8	1	2	0	2	1	1	0	1	2	0	1	0	0	0
class 9	0	1	0	6	0	0	0	0	1	1	0	1	3	0
class 10	0	2	0	6	2	0	0	1	1	3	1	1	0	0
class 11	0	1	0	4	2	0	0	1	1	0	0	0	0	0
class 12	0	0	0	3	2	0	0	0	1	2	1	4	0	0
class 13	0	1	0	2	0	0	0	0	0	6	3	1	0	0
class 14	0	0	0	2	2	0	0	0	1	1	1	3	0	1

TABLE 5.4: The Bias value of the neural network is 5. The highlighted one indicate the correctly classified image.

	class 1	class 2	class 3	class 4	class 5	class 6	class 7	class 8	class 9	class 10	class 11	class 12	class 13	class 14
class 1	2	2	0	5	1	1	0	0	0	0	1	0	0	0
class 2	2	7	0	6	6	3	0	2	5	0	3	0	1	2
class 3	1	0	0	6	2	1	1	4	0	1	2	2	0	1
class 4	1	2	1	13	8	0	4	2	1	1	3	2	0	1
class 5	2	1	0	4	8	0	0	1	1	0	2	0	0	0
class 6	0	2	0	0	0	5	7	1	1	1	0	1	0	1
class 7	0	0	0	0	0	1	5	2	0	0	0	0	0	0
class 8	1	4	0	3	1	0	0	0	0	0	2	0	0	0
class 9	3	1	0	2	0	1	1	1	1	0	3	0	0	0
class 10	0	1	0	4	5	0	0	1	3	0	2	1	0	0
class 11	0	0	1	3	1	1	0	0	0	0	3	0	0	0
class 12	0	0	0	1	0	0	0	0	1	0	5	3	0	3
class 13	0	3	1	1	0	0	0	0	0	0	6	0	1	1
class 14	0	0	0	0	0	1	1	0	0	0	0	0	0	9

TABLE 5.5: The learning rate for the neural network is 10^{-4} and number of epochs is equal to 5. The highlighted one indicate the correctly classified image

	clas s 1	clas s 2	clas s 3	clas s 4	clas s 5	clas s 6	clas s 7	clas s 8	clas s 9	clas s 10	clas s 11	clas s 12	clas s 13	clas s 14
Clas s 1	1	5	0	4	1	1	0	0	0	0	0	0	0	0
clas s 2	0	21	0	9	4	0	0	2	1	0	0	0	2	0
clas s 3	1	5	3	6	0	1	0	1	0	0	0	0	0	2
clas s 4	0	11	4	12	2	1	0	2	1	2	0	3	0	1
clas s 5	0	5	1	8	5	0	0	0	0	0	0	0	0	0
clas s 6	0	4	2	5	0	3	3	0	1	0	0	0	0	1
clas s 7	1	1	0	0	1	0	1	0	1	0	1	0	0	2
clas s 8	1	4	0	3	1	0	1	1	0	0	0	0	0	0
clas s 9	1	3	0	5	0	1	0	1	1	1	0	0	0	0
clas s 10	0	6	0	8	1	0	0	1	0	0	0	0	0	0
clas s 11	0	2	1	3	0	1	1	0	0	0	0	0	0	1
clas s 12	0	3	0	6	0	2	0	0	1	1	0	0	0	0
clas s 13	0	4	1	1	0	1	0	0	1	2	1	0	2	0
clas s 14	0	1	1	0	0	1	0	0	0	0	0	1	0	7

TABLE 5.6: The learning rate for the neural network is 10^{-5} and number of epochs is equal to 5. The highlighted one indicate the correctly classified image.

	10 ⁻⁴	10 ⁻⁷	10 ⁻⁵	BIAS=5	EPOCHS=5, 10 ⁻⁴	EPOCHS=5, 10 ⁻⁵
class #1	16.67	0	8.333333	0	16.66667	8.333333
class #2	45.94595	18.91892	35.13514	16.21622	18.91892	56.75676
class #3	19.04762	0	14.28571	0	0	14.28571
class #4	20.51282	30.76923	38.46154	33.33333	33.33333	30.76923
class #5	21.05263	0	26.31579	10.52632	42.10526	26.31579
class #6	36.84211	5.263158	21.05263	0	26.31579	15.78947
class #7	0	0	0	0	62.5	12.5
class #8	0	0	9.090909	9.090909	0	9.090909
class #9	0	0	7.692308	7.692308	7.692308	7.692308
class #10	0	5.882353	0	17.64706	0	0
class #11	33.33333	0	0	0	33.33333	0
class #12	15.38462	38.46154	0	30.76923	23.07692	0
class #13	23.07692	7.692308	15.38462	0	7.692308	15.38462
class #14	72.72	9.090909	18.18182	9.090909	81.81818	63.63636

Table 5.7: This table indicates the efficiency of the neural networks with respect to the changed parameters.

5.1 CONCLUSION

From the above table we can portray that for the arrangement of class #1 pictures, we need to change the learning rate of the neural organization to 10⁻⁴ or number of epochs to 5 with 10⁻⁴ learning rate. For the arrangement of class #2 pictures, we need to change the quantity of epochs to 5 and learning rate of the neural organization to 10⁻⁵. For the characterization of class #3 pictures, we need to change the learning rate of the neural organization to 10⁻⁴. For the order of class #4 pictures, we need to change the learning rate of the neural organization to 10⁻⁵. For the arrangement of class #5 pictures, we need to change the quantity of epochs to 5 and learning rate of the neural organization to 10⁻⁵. For the characterization of class #6 pictures, we need to change the learning rate of the neural organization to 10⁻⁴. For the order of class #7 pictures, we need to change the quantity of epochs to 5 and learning rate of the neural organization to 10⁻⁵. For the grouping of class #8 pictures learning rate can be 10⁻⁵ or bias to 5 or number of epochs equivalents to 5 with 10⁻⁵ learning rate. For the grouping of class #9 pictures learning rate can be 10⁻⁵ or bias to 5 or number of epochs equivalents to 5 with 10⁻⁴ learning rate or

number of epochs equivalents to 5 with 10^{-5} learning rate. For the order of class #10 pictures, we need to change the Bias of the neural organization to 5. From the above table we can portray that for the characterization of class #11 pictures, we need to change the learning rate of the neural organization to 10^{-4} or number of epochs to 5 and learning rate can be 10^{-4} . For the characterization of class #12 pictures, we need to change the learning rate of the neural organization to 10^{-7} . For the arrangement of class #13 pictures, we need to change the learning rate of the neural organization to 10^{-4} . For the arrangement of class #14 pictures, we need to change the quantity of epochs equivalents to 5 with 10^{-4} learning rate. By changing the previously mentioned boundaries regarding herbal grown plants, the efficiency for the arrangement of natural pictures can be expanded. This undertaking has halted at recreation level, equipment execution of this work should be possible in future too. Plausibility of other neural networks for the arrangement herbal grown pictures can be contemplated.

5.2 FUTURE NETWORK:

This project is stopped at simulation level. Hardware implementation of this work can be done in future as well. Feasibility of various neural networks for classification of plants can also be checked in future.

ORIGINALITY REPORT

4%

SIMILARITY INDEX

1%

INTERNET SOURCES

2%

PUBLICATIONS

2%

STUDENT PAPERS

PRIMARY SOURCES

1

Lau, Mian Mian, King Hann Lim, and Alpha Agape Gopalai. "Malaysia traffic sign recognition with convolutional neural network", 2015 IEEE International Conference on Digital Signal Processing (DSP), 2015.

Publication

<1 %

2

Submitted to National Institute Of Technology, Tiruchirappalli

Student Paper

<1 %

3

Raul Orus Perez. "Using TensorFlow-based Neural Network to estimate GNSS single frequency ionospheric delay (IONONet)", Advances in Space Research, 2019

Publication

<1 %

4

Submitted to Asia Pacific University College of Technology and Innovation (UCTI)

Student Paper

<1 %

5

Submitted to National Institute of Technology, Raipur

Student Paper

<1 %

6

Tatsuhiro Akiyama, Yosuke Kobayashi, Yuichi Sasaki, Kaori Sasaki, Takashi Kawaguchi, Jay Kishigami. "Mobile Leaf Identification System using CNN applied to plants in Hokkaido", 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE), 2019

Publication

<1 %

7

www.groundai.com

Internet Source

<1 %

8

ijeecs.iaescore.com

Internet Source

<1 %

9

Barkha M. Joshi, Hetal Bhavsar. "Plant leaf disease detection and control: A survey", Journal of Information and Optimization Sciences, 2020

Publication

<1 %

10

dblp.dagstuhl.de

Internet Source

<1 %

11

Ding Min Yue, Feng Qin. "Plant Leaf Recognition Based on Naive Bayesian Classification and Linear Discriminant Analysis Model", 2019 4th International Conference on Communication and Information Systems (ICCIS), 2019

Publication

<1 %

12

Nasibah Husna Mohd Kadir, Sharifah Nur Syafiqah Mohd Nur Hidayah, Norasiah

<1 %

Mohammad, Zaidah Ibrahim. "Comparison of convolutional neural network and bag of features for multi-font digit recognition", Indonesian Journal of Electrical Engineering and Computer Science, 2019

Publication

Exclude quotes On

Exclude matches Off

Exclude bibliography On