

A Swarm Automated Farm-Bot

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

by

Ramya M (37130342)

Vimala C (37130436)



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

SATHYABAMA

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

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

MARCH 2021



DEPARTMENT OF ELECTRONICS AND COMMUNICATIONENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Ramya M** (Reg. No. 37130342) and **Vimala C** (Reg. No. 37130436) who carried out the project entitled “**A Swarm Automated Farm-Bot**” under our supervision from December 2020 to March 2021.

Ms. Yogalakshmi M.E., Ph.D.,

Internal Guide

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

Head of the Department

Submitted for Viva voce Examination held on **19-04-2021**

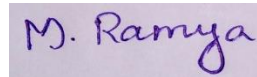
Internal Examiner

External Examiner

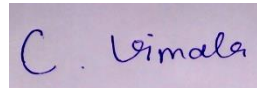
DECLARATION

We **Ramya M** (37130342) and **Vimala C** (37130436) hereby declare that the Project Report entitled “**A Swarm Automated Farm-Bot**” done by us under the guidance of **Ms. Yogalakshmi M.E., Ph.D.**, at Sathyabama Institute of Science and Technology, Chennai-600119 is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

1.

Handwritten signature of M. Ramya in purple ink on a light purple rectangular background.

2.

Handwritten signature of C. Vimala in purple ink on a light purple rectangular background.

Date: 19-04-2021

SIGNATURE OF THE CANDIDATES

Place:

ACKNOLEGEMENT

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. Nandhita, Dean, School of Electrical and Electronics Engineering** and **Dr. T. RAVI, Head of the Department, Dept. of Electrical and Electronics Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would also like to express our sincere and deep sense of gratitude to our internal guide **Ms. Yogalakshmi M.E., Ph.D.**, for her valuable guidance and suggestions during the progressive reviews for our project.

ABSTRACT

The project's goal is to design and build a swarm of robots with a symbiotic relationship. The goal of this robotic system is to demonstrate the concept of using a symbiotic relationship to minimize costs and improve versatility in the implementation of massive robot-assisted tasks.

This was accomplished by utilizing multiple configurations of robots with varying responsibilities, one of which, the platform's navigation and mobility were handled by the pilot, while the other was a sensory node that collected data and controlled the environment.

Swarm robots were used in this project to locate unhealthy plants and spray them with pesticides. The automatic detection of plant diseases is needed for food protection, yield loss estimation, and disease management. This research, in particular, has deep learning since agriculture is such a large sector to work in, we're using deep learning and CNN to detect plant diseases.

To put this concept into practice, we chose "tomato" as the core vegetable whose leaf diseases will be predicted using updated VGG model CNN algorithms. The average accuracy value of 90 test images using VGG19 is 66 percent, but it is 67.91 percent when using updated VGG19.

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1	Dataset Infected	11
2	Dataset healthy	12
3	Architecture of CNN	19
4	Layer of cnn	19
5	Convolution Layer	20
6	Pooling layer	21
7	Architecture of Alexnet	24
8	Architecture of Vgg16 Module	25
9	Architecture of Vgg19 Module	26
10	Comparison of Alexnet, Vgg16 and Vgg19	27
11	Output Graph of Original Vgg19	30
12	Output Graph of Modified Vgg19	31

TABLE OF CONTENT

CHAPER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	vi
1	INTRODUCTION	1
2	LITERATURE SURVEY	2-4
3	AIM AND SCOPE OF THE PROJECT	5-6
4	MATERIAL AND METHOD 4.1 METHOD 4.2 MATERIAL 4.3 MATLAB 4.4 ALGORITHM USED 4.4.1 CONVOLUTIONAL NEURAL NETWORK 4.4.2 ALEXNET 4.4.3 VGG16 4.4.4 VGG19	7-28
5	RESULT AND DISCUSSION 5.1 CONFUSION MATRIX 5.2 ORIGINAL VGG19 OUTPUT 5.3 MODIFIED VGG19 OUTPUT	29-31
6	CONCLUSION	32-34

CHAPTER 1

INTRODUCTION

Tomatoes are one of the best vitamin sources A and C among the 15 most common vegetables in the United States. It is one of the most adaptable vegetables, since it can be prepared in a variety of ways. In our region, the tomato is a seasonal crop that is only grown in the winter. However, with the aid of the Ministry of Agriculture and a few dedicated tomato farmers, the tomato is now an adult and sold throughout the year. Farmers have been helped by technology to make this possible. Even during the off-season, there is a high demand for tomatoes in our region. Throughout the year, however, there is a small in consistency, quantity, demand, and price in the tomato industry. Insect attacks, bad weather, the climate, soil, contaminated water, and toxic insecticides can all cause diseases in tomatoes.

Tomato plants are killed well before they reach maturity due to insect and disease attacks. Tomato leaf diseases are also dangerous. Plant disease is an unhealthy condition of a plant that prevents it from growing normally. Plant diseases come in a variety of forms and may result in significant losses in crop production. The presence of the pathogen is dependent on favourable environmental conditions and crop varieties, which is why plant diseases occur and are so common. There are a variety of plant disease control systems available to help minimize grain and yield quality losses. The current system for disease detection in such plants relies on visual observation, which is a time-consuming process.

In agriculture, image processing has been used for sorting, grading, and detecting defects such as dark spots, cracks, on new vegetables, as well as bruises and seeds, among other things. Deep Convolution Neural Networks (CNN) have evolved as a result of recent developments in hardware technology, and they now have a large number of applications, including complex tasks like object recognition and image processing, as well as smart phone-based applications for shape and disease detection in plant leaves. In this scenario, the goal of the research is to collect disease data from tomato plants, train a model for disease detection, and identify infected leaves.

CHAPTER 2

LITERATURE SURVEY

[1] TITLE : "Diseases Detection of Various Plant Leaf Using Image Processing Techniques : A Review" published in international conference on advanced computing and communication system(ICACCS).

AUTHORS : Santhosh Kumar.S, B.K.Raghavendra(2019).

CONTENT : Survey on various types of plants disease and different types of techniques to detect these diseases. Here they were discussed common diseases in Maize, arecanut, coconut trees, papaya, cotton.

[2] TITLE : "A study on the method of image pre-processing for recognition of crop diseases" published in international conference on advanced computer control.

AUTHORS : Geng Ying, Li Miao, Yuan Yuan, Hu Zelin.

CONTENT: Using edge detection Disease spots were detected and Snake model used in contour extraction. This paper compared the effect of simple and median filter.

[3] TITLE : "Leaves diseases detection of tomato using image processing" published in 8th international conference on system modeling and advancement in research trends.

AUTHORS : Tahmina Tashrif Mim, Md. Helal Sheikh, Roksana Akter Shampa, Md. Shamim Reza, Md. Sanzidul Islam(2019).

CONTENT : They chosen "Tomato" as the core vegetable which's leaf diseases are to be predicted by using the algorithms of Artificial Intelligence, CNN and computer science. Adam algorithm used to reduce the error and value loss and the adam value is 0.001.

[4]TITLE : "Tomato plant disease detection using image processing" published in international journal of advanced research in computer and communicationengineering.
AUTHORS : Sagar Vetal, R.S. Khule(2017).

CONTENT : In this paper four key diseases are identified using image segmentation and Multi-class SVM algorithm. For selecting of damaged area on leaves image segmentation is used and for classification of accurate disease Multi-class SVMalgorithm is used.

[5] TITLE : "Tomato Plant Disease Classification in Digital Images using Classification Tree" This paper was presented at an international conference on connectivity and signal processing.
AUTHORS : H. Sabrol and K. Satish(2016).

CONTENT : Five types of tomato diseases are classified in the study, including tomato late blight, Septoria spot, bacterial spot, bacterial canker, tomato leaf curl, and healthy tomato plant leaf and stem photos. Color, form, and texture features were extracted from safe and unhealthy tomato plant images to classify them. After the segmentation process, the feature extraction process begins. Features extracted from segmented images and fed to the classification tree

[6] TITLE : "Leaf Segmentation and Classification Deep Learning with a Complicated History".
AUTHORS : Kunlong Yang,Weizhen Zhong and Fengguo Li (2020).

CONTENT : In this paper, image segmentation done using Mask Region-based Convolutional Neural Network (Mask R-CNN) and classification is done using VGG-16 (CNN).

[7] TITLE : " Neural Networks basic using MATLAB Neural Network toolbox".
AUTHOR : Heikki N.Koivo(2008).

CONTENT : Study of Neural Network using MATLAB toolbox.

[8] TITLE : Deep neural networks based recognition of plant diseases by leaf image classification" published in hindawi publishing corporation computational intelligence and neuro science.

AUTHORS : Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk,sDarko Stefanovic(2016)."

CONTENT : The established model could detect the presence of leaves and differentiate between healthy and diseased leaves.

CHAPTER 3

AIM AND SCOPE OF THE PROJECT

The aim of this scenario is to gather disease data from tomato plants, train a disease detection model, and identify infected leaves. Plant disease is an unhealthy condition of a plant that prevents it from growing normally. Plant diseases come in a variety of forms and may result in significant losses in crop production. The presence of the pathogen is dependent on favourable environmental conditions and crop varieties, which is why plant diseases occur and are so common.

There are a variety of plant disease control systems available to help minimize grain and yield quality losses. Based on image processing techniques and methods proposed and studied, an automated identification and quantification of plant diseases was proposed and studied. The automated diagnosis method, which is focused on plant disease characteristics, eliminates the need for experts in the field. Using image processing techniques, the diseased region of the disease-affected images was used to obtain the characteristic function of plant disease.

Pattern recognition methods such as "neural networks" are used to identify disease- infected plants. The current research uses image processing techniques in MATLAB to detect tomato leaf disease. It entails image loading, image preprocessing such as image adjustment and normalization, feature extraction, and classification. The development of automated detection systems using advanced technologies such as image processing aids farmers in identifying diseases at an early or early stage and providing useful data for disease management.

This study proposes a groundbreaking system for detecting the impact of "tomato" leaves on tomato productivity in the agricultural industry around the world. With this proposed approach, we can now confidently state that the best mode for detecting tomato leaf diseases has been identified.

Expert acceptance of image processing and the CNN (convolution neural network) algorithm has inspired us to pursue a newly developed technology. In this scenario, the aim of the research is to collect disease data from tomato plants, train a model for disease detection, and identify infected leaves.

CHAPTER 4

METHOD AND MATERIALS

4.1 METHOD

A complete idea about the tomato disease diagnosis and detection process can be found in this research paper. CNN (convolutional neural networks) and image processing techniques have been used to create and train the system with tomato leaves dataset as input. CNN, proposed by LeCun, is a deep learning algorithm that exhibits the best performance in the field of image recognition. It can perform recognition even when the size and position of objects are changed, by mimicking the visual processing process of a leaf.

CNN is capable of re-learning the network to be applied to new recognition work through transfer learning, thereby enabling high accuracy and quick learning. Which can be divided into a feature extraction layer and classification layer, rectified linear unit (ReLU) layer, and pooling layer. The convolution layer activates image features by using a convolution filter on the input image. The ReLU layer eliminates the vanishing gradient of the network, which allows the network to learn quickly. The pooling layer performs a nonlinear down sampling of the image to simplify the output, and to reduce the number of network parameters.

In addition, the classification layer consists of nodes connected after passing through the feature detection layer. It is generally composed of a fully connected layer, and it provides the number of classifications that can be predicted by the network as a K-dimensional vector. In this study, we evaluated the performance after transfer learning using AlexNet, VGGNet (VGG16, VGG19), and GoogLeNet. After processing the given images, an accurate outcome is to be expected as output. The explanation for this is rural farmers having little knowledge on disease detection and advanced technologies can also use this system without any difficulty and hustle.

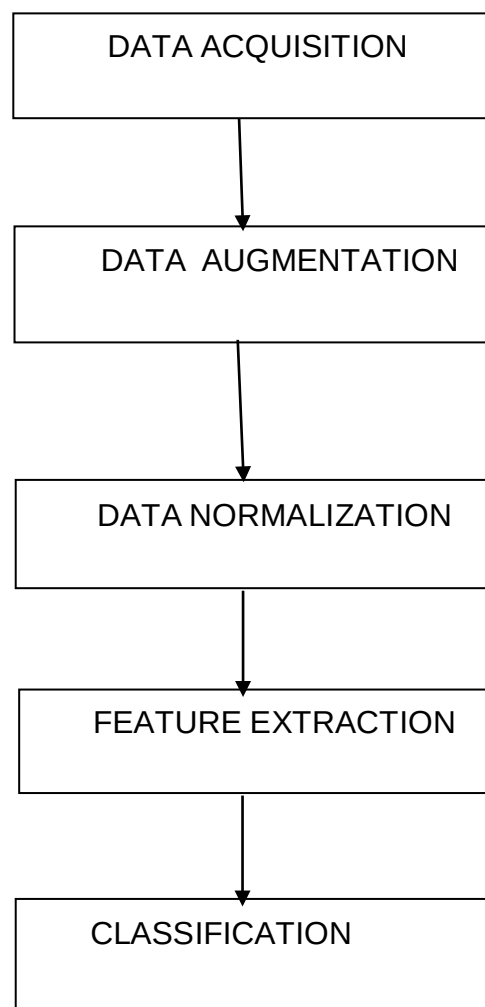
DETECTION AND CLASSIFICATION OF PLANT DISEASES

It comprises the following steps:

- A) Data Acquisition
- B) Data Preprocessing
- C) Feature Extraction
- D) Classification

Fig 1 shows the flowchart of this process.

FLOW CHART



A. DATA ACQUISITION

The data collection is made up of 4000 images that covers crops and is divided into 2 different classes. This data is derived from the original Plant village dataset. Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer.

B. DATA PREPROCESSING

It is done to data conversion in a form so that feature extraction method and subsequent steps can work properly. Data augmentation and Normalization are done in this step. Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing.

Data Augmentation is a popular regularization method which provides a sound solution to the over fitting problem. It results in better learning of the model. Images are rotated by 90 degrees where the probability of whether an image is rotated is 0.75. Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset. Training deep learning neural network models on more data can result in more skillful models, and the augmentation techniques can create variations of the images that can improve the ability of the fit models to generalize what they have learned to new images.

Normalization is done in order to make all the pixel values have the same mean and standard deviation. Normalization gives equal weights/importance to each variable so that no single variable steers model performance in one direction just because they are bigger numbers. As an example, clustering algorithms use distance measures to determine if an observation should belong to a certain cluster. "Euclidean distance" is

often used to measure those distances. If a variable has significantly higher values, it can dominate distance measures, suppressing other variables with small value.

C. FEATURE EXTRACTION

To solve the classification problem at hand, relevant features are extracted first. Features in images are color, shape and texture. Systems that detect diseases using leaf images focus more on the texture feature. Some examples of the techniques that may be used are Grey-level co-occurrence Matrix, auto-correlation, Gabor Transformation, 2D Gabor function etc. Gray-level co-occurrence matrix (GLCM) is a statistical method for determining how often pairs of pixels with certain values and in a specific spatial relationship appear in an image, which characterises the image texture. The degree of correlation between a given time series and its previous version over successive time periods is depicted by autocorrelation.

The Gabor transform is a type The phase content and sinusoidal frequency of local portions of a signal over time are evaluated using the short-time Fourier transform. Gabor functions can model the simple neurons of the brain visual cortex of mammals. 2-D Gabor function is used to simulate the space-based summation properties of simple cells (of the receptive field) in the visual cortex. Besides high accuracies, extracting features automatically has proved to be amongst the major advantages of using deep learning models. Since we modified VGG19, which is a deep learning model is used along with classification, it also handles automatic feature extraction. Thus, there is no need to use a separate feature extraction method in our proposed approach.

D. CLASSIFICATION

There are multiple choices available for classification. Some of the classifiers that maybe used in this step are Logistic Regression, Radial Basis Function, Linear Vector Quantization, ANN, Classification Trees, Support Vector Machines, CNN, K-NN etc. Vgg19 – a CNN architecture is used for classification purpose in the proposed method.

4.2 MATERIALS

DATASET

In this research we have chosen about 4000 image data to work with. Dataset of this research contains about 9 types of sample of tomato leaf diseases and one additional healthy class building 2 types of class as below:

Fig 1:-Tomato Leaf Infected :-





Fig 2:-Tomato Leaf Healthy :-



4.3 MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or Fortran.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

The MATLAB System :-

The MATLAB system consists of five main parts:

The MATLAB language:-

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

The MATLAB working environment. :-

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

Handle Graphics:-

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete Graphical User Interfaces on your MATLAB applications.

The MATLAB mathematical function library :-

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigen values, Bessel functions, and fast Fourier transforms.

DEEP NEURAL NETWORK:-

Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network. Deep learning is an AI function that mimics the workings of the human brain in processing data for use in detecting objects, recognizing speech, translating languages, and making decisions.

- Deep learning AI is able to learn without human supervision, drawing from data that is both unstructured and unlabeled.
- Deep learning, a form of machine learning, can be used to help detect fraud or money laundering, among other functions.

Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing.

However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support. Deep learning unravels huge amounts of unstructured data that would normally take humans decades to understand and process.

Deep Learning vs. Machine Learning :-

One of the most common AI techniques used for processing big data is machine learning, a self-adaptive algorithm that gets increasingly better analysis and patterns with experience or with newly added data. If a digital payments company wanted to detect the

occurrence or potential for fraud in its system, it could employ machine learning tools for this purpose. The computational algorithm built into a computer model will process all transactions happening on the digital platform, find patterns in the data set, and point out any anomaly detected by the pattern. Deep learning, a subset of machine learning, utilizes a hierarchical level of artificial neural networks to carry out the process of machine learning.

The artificial neural networks are built like the human brain, with neuron nodes connected together like a web. While traditional programs build analysis with data in a linear way, the hierarchical function of deep learning systems enables machines to process data with a nonlinear approach. *Electronics maker Panasonic has been working with universities and research centers to develop deep learning technologies related to computer vision.*¹

Special Considerations:-

A traditional approach to detecting fraud or money laundering might rely on the amount of transaction that ensues, while a deep learning nonlinear technique would include time, geographic location, IP address, type of retailer, and any other feature that is likely to point to fraudulent activity. The first layer of the neural network processes a raw data input like the amount of the transaction and passes it on to the next layer as output. The second layer processes the previous layer's information by including additional information like the user's IP address and passes on its result. The next layer takes the second layer's information and includes raw data like geographic location and makes the machine's pattern even better. This continues across all levels of the neuron network.

A Deep Learning Example :-

Using the fraud detection system mentioned above with machine learning, one can create a deep learning example. If the machine learning system created a model with parameters built around the number of dollars a user sends or receives, the deep-learning method can start building on the results offered by machine learning. Each layer of its neural network builds on its previous layer with added data like a retailer, sender, user, social media event, credit score, IP address, and a host of other features that may take years to connect together if processed by a human being. Deep learning algorithms are trained to not just create patterns from all transactions, but also know when a pattern is signaling the need for a fraudulent investigation. The final layer relays a signal to an analyst who may freeze the user's account until all pending investigations are finalized. Deep learning is used across all industries for a number of different tasks. Commercial apps that use image recognition, open-source platforms with consumer recommendation apps, and medical research tools that explore the possibility of reusing drugs for new ailments are a few of the examples of deep learning incorporation.

4.4 ALGORITHM USED

4.4.1 Convolutional neural network :

Inspiration from the workings of the human brain, led to the discovery of a class of algorithms, which falls in the sphere of deep learning. It involves the training of ANN to make predictions. ANN are networks of neurons arranged in a multilayer fashion. The output of one layer moves to the next layer as input. Features are automatically extracted by the deep learning models during training, therefore no separate method is required for the feature extraction step of the basic process that has been mentioned above. The initial layers of ANN learn the low level features (like edges) and as deeper, higher level features (like complete objects) are learned.

CNN is a unique form of neural organization. ANN that is customized for image processing. Researches have shown the capability of CNN to give high accuracy in image processing tasks. In conventional ANN, a single neuron in a layer takes its data from the previous layer's neurons. Thus, for image-based tasks, it creates a problem of learning quite a few parameters. CNN is preferred over ANN for image-related tasks due to less number of parameters involved in comparison to ANN. Parameter sharing is responsible for this reduction. Same parameters (filters in terminology of CNN) are used to get all the activations in the output volume from a volume of input of activations. This is called parameter sharing. The goal of a ConvNet is to decrease the size of an image without losing crucial features that helps in solving the problem.

Fig 3:- Architecture of cnn

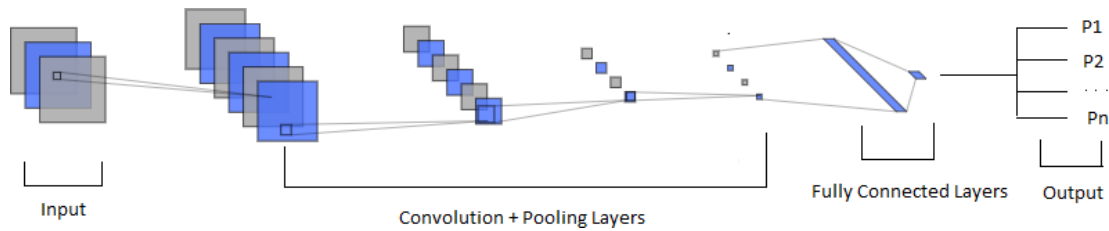
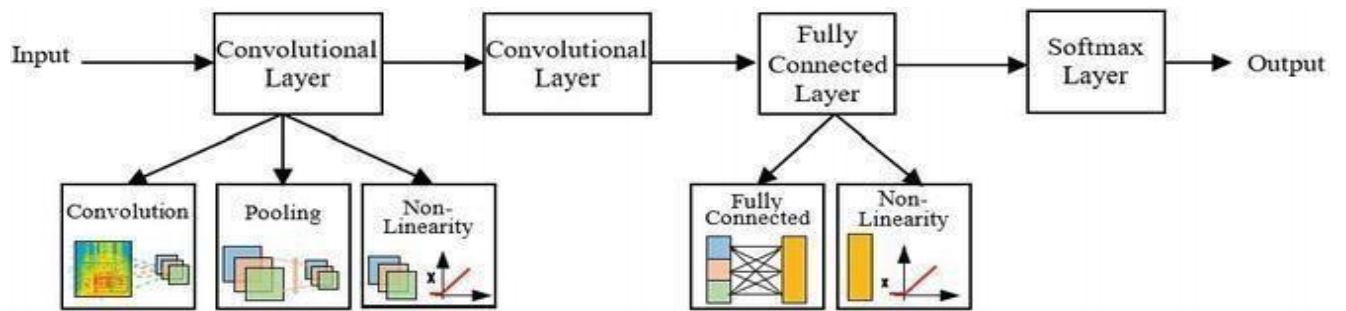


Fig 4:- Layers of cnn



LAYERS OF CNN

1] CONVOLUTIONAL LAYER

CNN was named after the convolutional layer. The dimensions of the image is reduced by applying a series of convolutional operations. A filter/kernel is first placed on the upper left corner of the image and is then shifted along the width of the image towards right by some stride value. After covering the entire width, the filter then hops down by the same stride value and starts again from the left in order to cover the entire width. This process is repeated till the whole image is

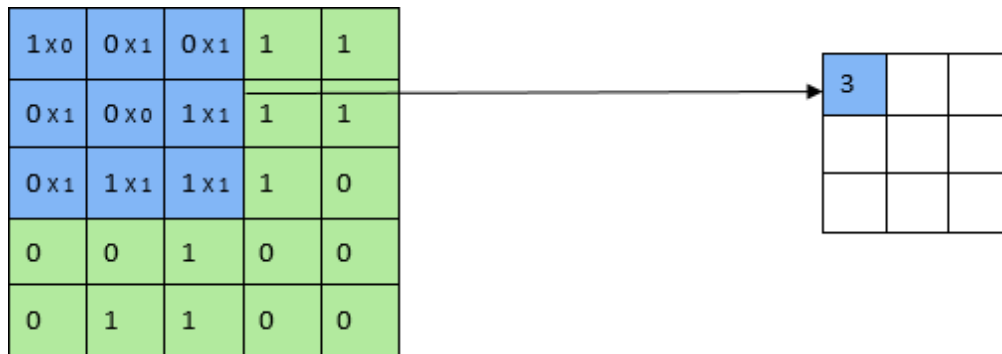


Fig. 4: Convolutional operator applied to 5 X 5 input and 3 X3 filter.

traversed [10]. The sum of product of corresponding values in the overlapped portion of the image and filter is evaluated at a single step. This gives rise to a new matrix (or volume) from the input matrix (or volume). Fig 3 shows convolution operation applied to a 5 X 5 image with 3 X 3 filter in a convolutional layer.

2] POOLING LAYER

This layer has the functions of reducing the image size and extracting the features that are dominant. A filter is placed and moved in the same manner as in case of a convolutional layer. A function is applied at a single step in this layer. This function can be a max function which finds the maximum of all the values in the overlapped portion of the input image and the filter (called Max-pooling) or it can be an average function (called avg-pooling). Generally, stride and filter size used is 2.

Max-pooling is preferred over avg-pooling.

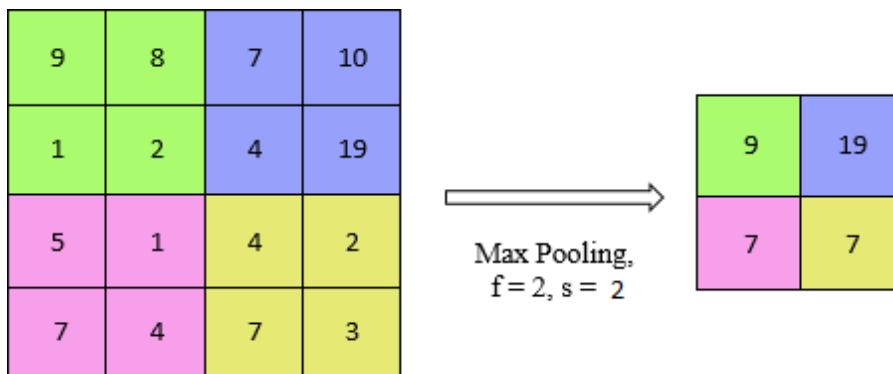


Fig 5 : maximum pooling layer

The addition of a pooling layer after the convolutional layer is a common pattern used for ordering layers within a convolutional neural network that may be repeated one or more times in a given model. The pooling layer operates upon each feature map separately to create a new set of the same number of pooled feature maps. Pooling involves selecting a pooling operation, much like a filter to be applied to feature maps. The size of the pooling operation or filter is smaller than the size of the feature map; specifically, it is almost always 2×2 pixels applied with a stride of 2 pixels. This means that the pooling layer will always reduce the size of each feature map by a factor of 2, e.g. each dimension is halved, reducing the number of pixels or values in each feature map to one quarter the size.

For example, a pooling layer applied to a feature map of 6×6 (36 pixels) will result in an output pooled feature map of 3×3 (9 pixels). The pooling operation is specified, rather than learned. Two common functions used in the pooling operation are:

Average Pooling: Calculate the average value for each patch on the feature map.

Maximum Pooling (or Max Pooling): Calculate the maximum value for each patch of the feature map.

The result of using a pooling layer and creating down sampled or pooled feature maps is a summarized version of the features detected in the input. They are useful as small changes in the location of the feature in the input detected by the convolutional layer will

result in a pooled feature map with the feature in the same location. This capability added by pooling is called the model's invariance to local translation. In all cases, pooling helps to make the representation become approximately invariant to small translations of the input. Invariance to translation means that if we translate the input by a small amount, the values of most of the pooled outputs do not change.

3] FULLY CONNECTED LAYER

The output of a series of convolutional plus pooling layers is a matrix (or volume). This is flattened and then a sequence of fully connected layers is used, similar to ANN. A single neuron in a layer takes its both neurons in the previous layer's input. FC Layers are used only when the size of image is reduced enough by the series of convolutional plus pooling layers so that the fully connected layers don't have large number of parameters to learn. Fully connected layers were unique to CNNs. However, they typically were included as the last few layers of most CNNs, appearing after several convolution and subsampling operations were performed. Fully connected layers were independent neural networks that possessed one or more hidden layers. Their operations involved multiplying their inputs by trainable *weight* vectors, with a trainable *bias* sometimes summed to those results. The output of these layers was traditionally sent through activation functions, similarly to convolution layers.

3] ACTIVATION LAYER

An The activation mechanism is used. to each element of the input matrix (or volume). So, the dimensions of input and output are the same for this layer. Linear activation functions can only help in approximating linear hypothesis functions. Since, a non-linear relationship generally occurs between input and output for complex problems, so nonlinear activation functions are commonly used. ReLU function is mostly used because it results in faster learning. ReLU function is the most widely used activation function in neural networks today. One of the greatest advantage ReLU has over other activation functions is that it does not activate all neurons at the same time. From the image for ReLU function above, we'll notice that it converts all negative inputs

to zero and the neuron does not get activated. This makes it very computationally efficient as few neurons are activated per time. It does not saturate at the positive region. In practice, ReLU converges six times faster than tanh and sigmoid activation functions. Some disadvantage ReLU presents is that it is saturated at the negative region, meaning that the gradient at that region is zero. With the gradient equal to zero, during backpropagation all the weights will not be updated, to fix this, we use Leaky ReLU. Also, ReLU functions are not zero-centered. This means that for it to get to its optimal point, it will have to use a zig-zag path which may be longer

4.4.2 ALEXNET

AlexNet was the first convolutional network to use the graphics processing unit (GPU) to improve performance. AlexNet has three max-pooling layers, five convolutional layers, two normalisation layers, two completely connected layers, and one softmax layer in its architecture. 1. AlexNet architecture consists of 5 convolutional layers, 3 max-pooling layers, 2 normalization layers, 2 fully connected layers, and 1 softmax layer. Each convolutional layer consists of convolutional filters and a nonlinear activation function ReLU. The pooling layers are used to perform max pooling. Input size is fixed due to the presence of fully connected layers The input size is mentioned at most of the places as $224 \times 224 \times 3$ but due to some padding which happens it works out to be $227 \times 227 \times 3$ AlexNet overall has 60 million parameters.

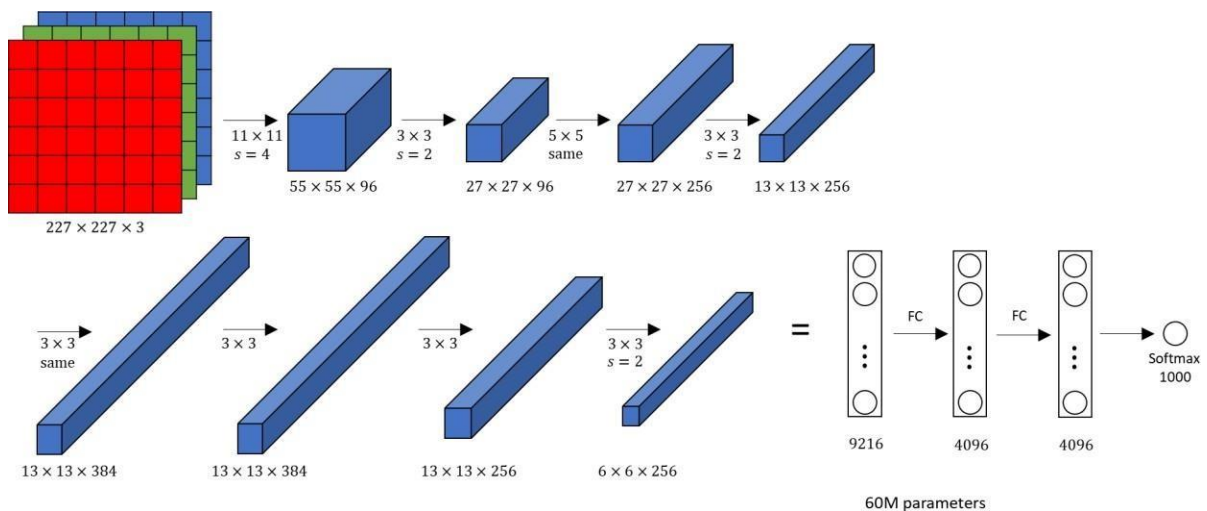


Fig 6:- Architecture of alexnet

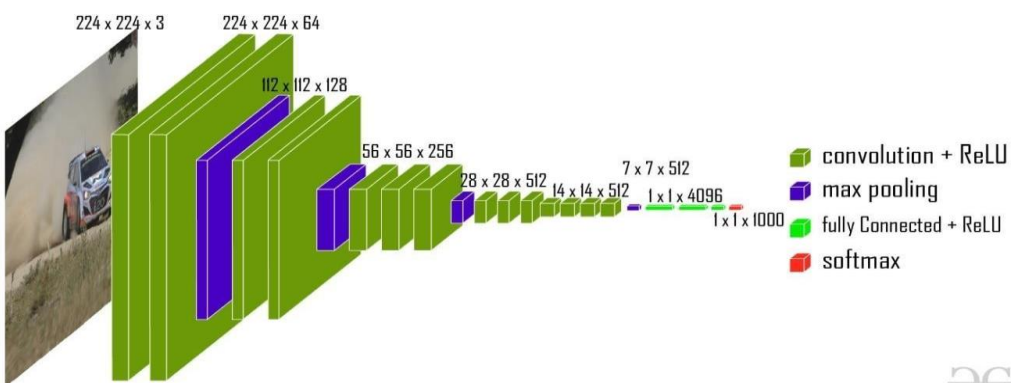
4.4.3 VGG 16

K suggested the VGG16 convolutional neural network model. It outperforms AlexNet by sequentially Wide kernel-sized filters (11 and 5 in the first and second convolutional layers, respectively) with multiple 33 kernel-sized filters. The input to conv1 layer is of fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations, it also utilizes 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity).

The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3×3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2×2 pixel window, with stride 2. Three Fully-Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks. All hidden layers are equipped with the rectification (ReLU) non-linearity.

It is also noted that none of the networks (except for one) contain Local Response Normalisation (LRN), such normalization does not improve the performance on the ILSVRC dataset, but leads to increased memory consumption and computation time.

Fig 7: architecture of vgg16



4.4.4 VGG19

The VGG19 model is a variation of the VGG model that has 19 layers in total (16 layers of convolution, 3 layers of completely connected, 5 layers of MaxPool, and 1 layer of SoftMax). VGG11, VGG16, and other VGG variants are among the others. The FLOP count of VGG19 is 19.6 billion.

A fixed size of (224 * 224) RGB image was given as input to this network which means that the matrix was of shape (224,224,3). The only preprocessing that was done is that they subtracted the mean RGB value from each pixel, computed over the whole training set. Used kernels of (3 * 3) size with a stride size of 1 pixel, this enabled them to cover the whole notion of the image. spatial padding was used to preserve the spatial resolution of the image. max pooling was performed over a 2 * 2 pixel windows with stride 2.

This was followed by Rectified linear unit (ReLU) to introduce non-linearity to make the model classify better and to improve computational time as the previous models used tanh or sigmoid functions this proved much better than those. implemented three fully connected layers from which first two were of size 4096 and after that a layer with 1000 channels for 1000-way *ILSVRC* classification and the final layer is a softmax function.

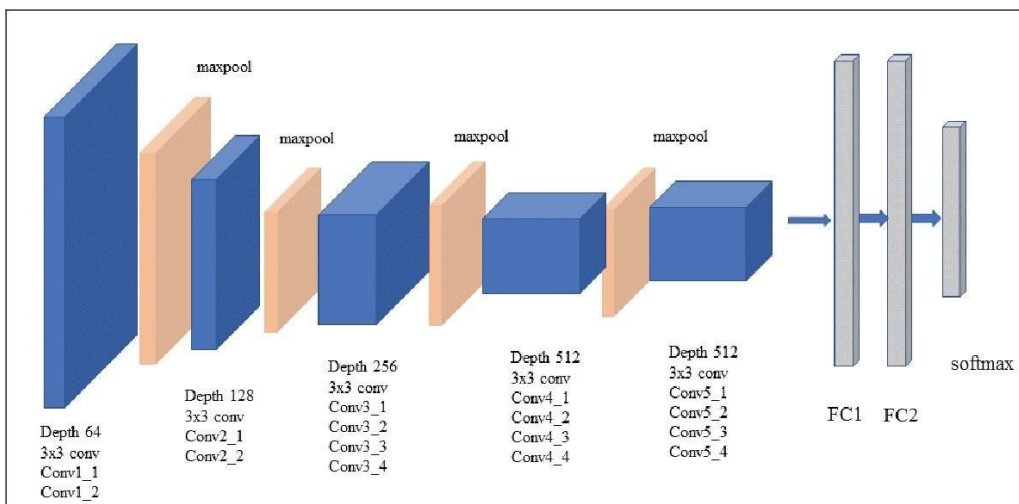


Fig 8:- Architecture of vgg19 model

The first two layers are convolutional layers with 3*3 filters, and first two layers use 64 filters that results in 224*224*64 volume as same convolutions are used. The filters are always 3*3 with stride of 1 After this, pooling layer was used with max-pool of 2*2 size and stride 2 which reduces height and width of a volume from 224*224*64 to 112*112*64 This is followed by 2 more convolution layers with 128 filters. This results in the new dimension of 112*112*128. After pooling layer is used, volume is reduced to 56*56*128. Two more convolution layers are added with 256 filters each followed by down sampling layer that reduces the size to 28*28*256. Two more stack each with 3 convolution layer is separated by a max-pool layer. After the final pooling layer, 7*7*512 volume is flattened into Fully Connected (FC) layer with 4096 channels and softmax output of 1000 classes.

Comparison of CNN :-

Fig 9: comparison of alexnet, vgg16 and vgg19.

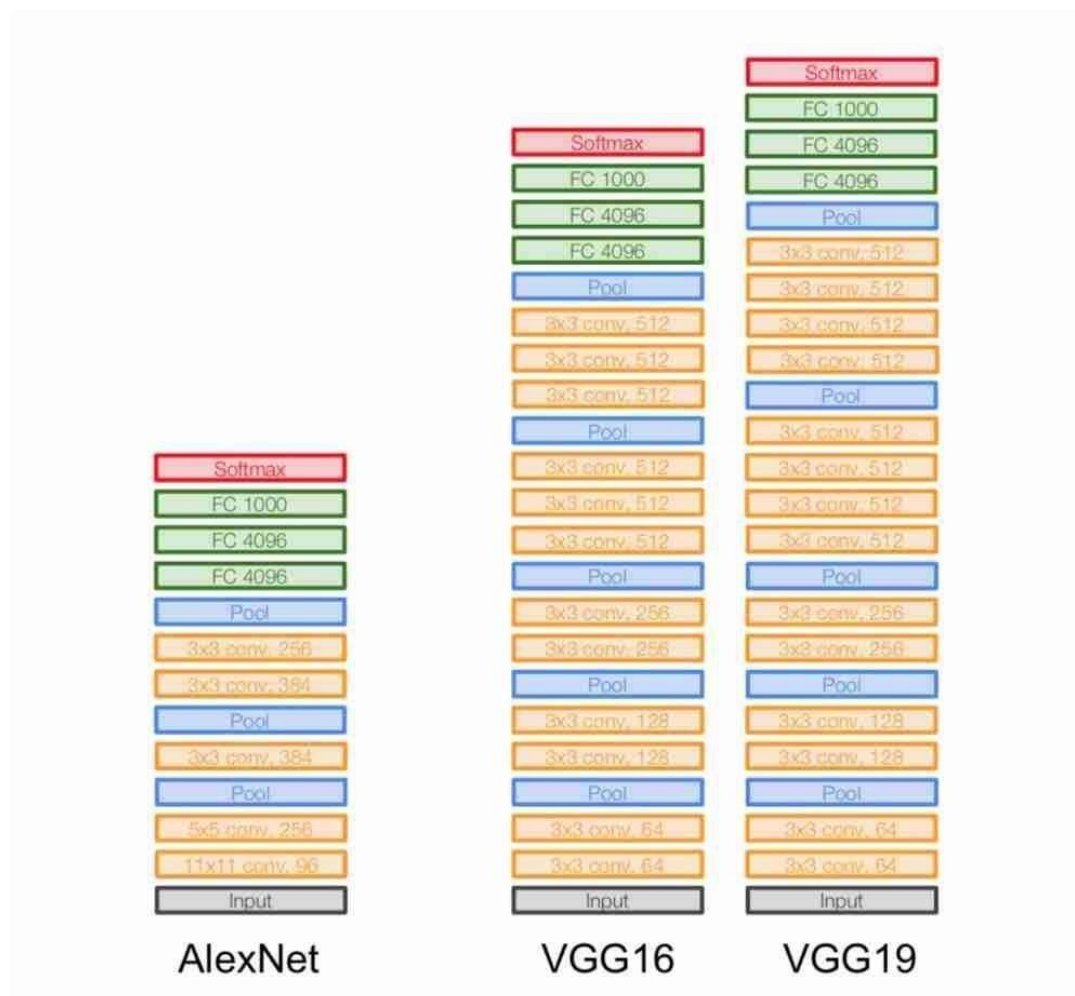


Figure 7 shows a comparison of alexnet, vgg16, and vgg19, as well as the architecture of all three network layers. AlexNet has three max-pooling layer, five convolutional layers, two normalisation layers, two completely connected layers, and one softmax layer in its architecture. K suggested the VGG16 convolutional neural network model. It outperforms AlexNet by sequentially Wide kernel-sized filters (11 and 5 in the first and second convolutional layers, replaced respectively) with multiple 3x3 kernel-sized filters. The VGG19 model is a variation of the VGG model that has 19 layers in total (16 layers of convolution, 3 layers of completely connected, 5 layers of MaxPool, and 1 layer of SoftMax).

CHAPTER 5

RESULT AND DISCUSSION

We tweaked the VGG 19 CNN to improve accuracy while lowering the loss. In comparison to the original vgg19, our modern convolutional neural network provides better accuracy. There are 19 layers in the original vgg19 cnn (16 convolution layers, 3 Fully connected layer, 5 MaxPool layers and 1 SoftMax layer). There are 21 layers in the updated vgg19 cnn (18 convolutional layers, 3 completely linked layers, and 5 MaxPool layers). For our dataset, the accuracy of the initial vgg19 is 66.22 percent. The accuracy of the adjusted vgg19 is 67.99 percent. The consistency and losses of the leaves are depicted in the graph. The loss value indicates how well or poorly a model performs after each optimization iteration. An interpretable accuracy metric is used to assess the algorithm's success. It's a metric for how accurate the model is compare to true data.

5.1 CONFUSION MATRIX

A confusion matrix is a way of summarising the results of a classification algorithm. Classification accuracy alone can be deceptive if you have an unequal number of observations in - class or if your dataset has more than two classes. Confusion Matrix Calculation will help you understand what your classification model gets correct and what types of errors it makes. It creates an uncertainty matrix that can be used to assess recall, precision, accuracy, and the AUC-ROC curve. An example of the terms Here are the definitions of True Positive, True Negative, False Negative, and True Negative. True Positive: You predicted something positive and it came true.

5.2 ORIGINAL VGG 19 OUTPUT

In this graphical representation, we display the original vgg19 cnn output and our updated net output. The graphical performance of our dataset is shown in Figure 11. The dataset's precision and losses are depicted in the graph below. The initial vgg19 net gives a 66.22 percent accuracy for our dataset. Fig 11:- output graph of original vgg19.

CONFUSION MATRIX OF ORIGINAL VGG19

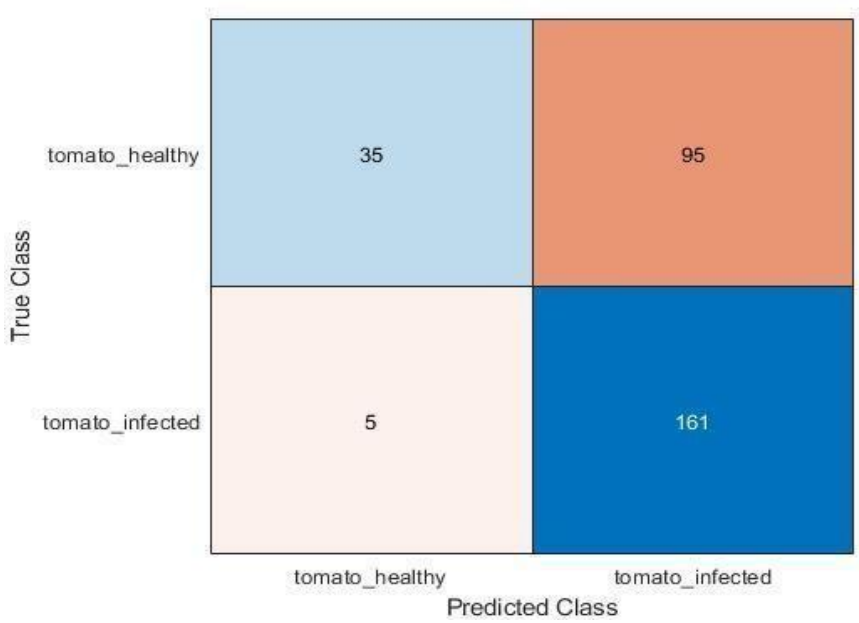


Fig 10: Output Graph of Original Vgg19

5.3 MODIFIED VGG19 OUTPUT

The graphical performance of our dataset is shown in Fig 12. The dataset's precision and losses are depicted in the graph below. The initial vgg19 net gives a 67.91 percent accuracy for our dataset.



Fig 11:- Output graph of modified vgg 19.

CONFUSION MATRIX OF MODIFIED VGG19

True Class	Predicted Class	
	tomato_healthy	tomato_infected
tomato_infected	35	95
tomato_healthy	31	166

CHAPTER - 6

CONCLUSION

This study proposes a groundbreaking method for detecting affected "tomato" leaves in order to reduce tomato productivity in the agricultural industry worldwide. We can now proudly state that the best model for detecting tomato leaf diseases has been developed using this proposed process. Expert acceptance of image processing and the CNN (convolution neural network) algorithm has inspired us to pursue a newly developed technology. This study looked into and examined affected tomato leaves extensively, and found no way to detect diseases more accurately than the initial vgg19 model. Our research has freed local farmers, as well as farmers all over the world, from the time-consuming and labor-intensive manual disease detection process.

6.1 REFERENCE

By Tahmina Tashrif Mim, Md. Helal Sheikh, Roksana Akter Shampa, Md. Shamim Reza, and Md. Sanzidul Islam, "Leaves diseases detection of tomato using image processing" (2019).

Kunlong Yang, Weizhen Zhong, and Fengguo Li's paper "Leaf Segmentation and Classification with a Complicated Context Using Deep Learning" (2020).

Heikki N.Koivo's "Neural Networks Simple Using MATLAB Neural Network Toolbox" (2008).

Santhosh Adhikari, Bikesh Shrestha, and Bibek Baiju developed a method for detecting tomato plant diseases using image processing.
<https://discovery.mathworks.com/convolutional-neural-network-matlab/> — neural network with convolutions

Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, Darko Stefanovic (2016) published "Deep neural networks based recognition of plant

diseases through leaf picture classification."

“Plant Diseases Encyclogry 2012,” by Arthur Kelmam. Plant diseases can be found at <http://www.britannica.com/science/plantdisease>. Ferentinos, K.P.. (2018). Plant disease identification and diagnosis using deep learning models. Agriculture and Computers and Electronics 145. 311-318 10.1016/j.compag.2018.01.009.

Adhikari, Santosh & Shrestha, Bikesh & Baiju, Bibek & K.C., Saban. (2018). "TOMATO PLANT DISEASES DETECTION SYSTEM USING IMAGE PROCESSING". 1st KEC Conference on Engineering and Technology "Agriculture Sector in India." [Online]. Available: [SrdjanSladojevic, Marko Arsenovic, AndrasAnderla, DubravkoCulibrk, and Darko /Stefanovic] SrdjanSladojevic, Marko Arsenovic, AndrasAnderla, DubravkoCulibrk, and Darko Stefanovic Hindawi Publishing Corporation Computational Intelligence and Neuroscience, Volume 2016, Article ID 3289801, 11 pages, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification." 10.1155/2016/3289801 (<http://dx.doi.org/10.1155/2016/3289801>) <https://github.com/PrajwalaTM/tomato-leaf-disease-detection>

M.R. Karim, M.S. Alam, M.R. Karim, M.R. Karim, M.R. Karim, M.R. Karim, M.R.
Karim, M.R. Karim, M.R. Karim, M.R. Karim, M.R. Karim, M.R.“CS231n Visual
Recognition with Convolutional Neural Networks.” github.com/cs231n On the 25th of
April, 2017, I was able to get a hold of some “Plant disease diagnosis,” by B. Riley,
M. R. Williamson, and O. Maloy. The P lant Healt h Instructor,” published in 2002.

K. A. Garrett, S. P. Dendy, E. E. Frank, M. N. Rouse, and S. E. Travers, "Climate change effects on plant disease: genomes to ecosystems," *Annual Review of Pathology*, vol. 44, pp. 489–509, 2006. "Climate change: potential effect on plant diseases," *Environmental Pollution*, vol. 108, no. 3, pp. 317–326, 2000. S. Chakraborty, A. V. Tiedemann, and P. S. Teng, "Climate change: potential impact on plant diseases," *Environmental Pollution*, vol. 108, no. 3, pp. 317–326, 2000.

Small-Scale Tomato (*Lycopersicon esculentum*) Production Profitability in a Few Selected Areas in Bangladesh, Mst. Samshunnahar, Romaza Khanum*, and M. Serajul Islam. *The Agriculturists*, vol. 14, no. 1, pp. 73-82 ISSN 2304-7321 (2016) (Online), 1729-5211 (Print) (Print) Krishi Foundation's Scientific Journal has an impact factor of 0.402. (GIF, 2014)

Detection of multi-tomato leaf diseases (late blight, aim, and bacterial spots) in different stages using a spectral-based sensor, Jinzhu Lu, Reza Ehsani, Yeyin Shi, Ana Isabel de Castro, and Shuang Wang. 1st of December 2018.
DOI:10.1038/s41598-018-21191-6 DOI:10.1038/s41598-018-21191-6
DOI:10.1038/s4

Deep Learning for Tomato Diseases: Classification and Symptom Visualization, Mohammed Brahimi, Kamel Boukhalfa, and Abdelouahab Moussaoui (2017) Visualization, Applied Artificial Intelligence, 31:4,299315, DOI: 10.1080/08839514.2017.1315516

K. A. Garrett, S. P. Dendy, E. E. Frank, M. N. Rouse, and S. E. Travers, "Climate change effects on plant disease: genomes to ecosystems," *Annual Review of Pathology*, vol. 44, pp. 489–509, 2006.

"Climate change: potential effect on plant diseases," *Environmental Pollution*, vol. 108, no. 3, pp. 317–326, 2000. S. Chakraborty, A. V. Tiedemann, and P. S. Teng, "Climate change: potential impact on plant diseases," *Environmental Pollution*, vol. 108, no. 3, pp. 317–326, 2000.

"tomato plant diseases detection method using image processing," by Santosh Adhikari, Bikesh Shrestha, Bibek Baiju, and Er. Saban Kumar K.C. 2018 KECC Conference Prashant Rothe and V. Kshirsagar, R. (2015).

1-6. 10.1109/PERVASIVE.2015.708698 Cotton leaf disease detection using pattern recognition techniques.