Pretrained CNN Architectures: A Detailed Analysis Using Bell Pepper Image Datasets

Midhun P Mathew (midhunpmathew@cusat.ac.in)
SOE CUSAT Kochi
Sudheep Elayidom .M
SOE CUSAT Kochi
Jagathyraj VP
SMS CUSAT
Therese Yamuna Mahesh
AmalJyothi college

Research Article

Keywords: CNN Architecture, Disease classification, deep learning, Resnet 18, Resnet 50, Resnet 101, Alex Net, Google Net, Vgg 16, Vgg19.

Posted Date: July 13th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3146418/v1

Additional Declarations: No competing interests reported.

Pretrained CNN Architectures: A Detailed Analysis Using Bell Pepper Image Datasets.

Midhun P Mathew [1] Sudheep Elayidom .M[2] JagathyrajVP[3] Therese Yamuna Mahesh[4]
[1]Reserch scholar SOE, CUSAT [2] Professor, CS division, SOE CUSAT, [3]Senior Professor SMS CUSAT,
[4]Associate Professor, Dept. ECE, Amal Jyothi College

Corresponding Author:
Midhun P Mathew
Research Scholar
CS Division, SOE CUSAT
Kochi, Kerala, India.

E-mail: midhunpmathew@cusat.ac.in

Running Title: **Pretrained CNN Architectures: A Detailed Analysis Using Bell Pepper Image Datasets.**

Abstract

In the era of artificial intelligence, automation is becoming popular in every sector. The primary sector includes the agriculture sector. Farmers are facing problems such as the identification of diseases in their plants, lack of proper treatment for the disease, climatic changes that affect their yield, and low price for their crops. In this paper, we are mainly focusing on the disease identification of bell pepper plants using deep learning architectures such as Alex Net, google net, ResNet (18,50,101), and Vgg (16,19). We also focus on the detailed study of different pre-trained CNN architectures to analyze their performance and identify which architecture is more suitable for disease classification in bell pepper. This paper also helps bell pepper farmers to identify the disease with high accuracy compared to the traditional methods of disease identification. The new automation concept helps bell pepper framers to identify diseases with less time and effort, which makes their work easier. The identification of disease at an early stage with less effort will help the farmer to increase their yield. The paper will help to understand the performance of different pre-trained convolutional neural network architectures with and without augmentation of images and also compare the performance of the architectures. Based on these comparisons, it could find out that google net is more suitable for the classification of images in bell pepper as compared to other architectures with augmentation, and vgg19 was observed to be best for the classification of images without augmentation.

Keywords

CNN Architecture, Disease classification, deep learning, Resnet 18, Resnet 50, Resnet 101, Alex Net, Google Net, Vgg 16, Vgg19.

I. INTRODUCTION

Artificial intelligence can serve to perform several practical applications with automation. Machine learning is the subset of artificial intelligence. In Machine learning, the most important part is data. It acts as a statistical tool for analyzing the data and prediction. There are different techniques available in machine learning such as supervised, unsupervised, and semi-supervised. In Supervised learning, there are two types of techniques, one is regression and other is classification. In the case of regression, the values are continuous and in the case of classification, the values are discrete. In Unsupervised machine learning, the main tasks include clustering, segmentation, dimension reduction, etc. In semi-supervised machine learning, we have labeled data, then we train the model and use the model for testing data. While interacting with the model, it analyses the data and then gives the output. Deep learning is a subset of machine learning. In deep learning the main component is a neural network. The emergence of deep learning is based on the concept of the human brain. According to the researcher, Humans learn things based on the thinking capacity of the brain. So, the researcher uses this concept in deep learning and defining the main aim of deep learning is to mimic the human brain, which means machine should learn as similar way as that of human brain. In deep learning there are different techniques such as Artificial neural network, convolutional neural network, and recurrent neural network. In this paper we focus on the CNN concept in which input is either video or image. Application of CNN include image classification, object detection, object segmentation, tracking, so on [4,7]. This paper explains about the transfer learning concept of CNN architecture and also explains the Application of the CNN mainly the image classification. The paper explains about the usage of CNN architecture in diseases classification in bell pepper based on different architecture of convolutional neural network such as ResNet, Vgg, google net, and Alex net. The Paper explains about the performance of different architecture in the noise environment.

Paper is divided into different section as follows, first section explains about the introduction of Deep learning, and its different techniques. Second section explains about the previous study which explains about the importance of our work in the current time. Third section explains about the Pretrained deep neural network. The fourth section explains about the Main components of the CNN architecture. The fifth section explains about the Pretrained CNN Architecture such as ResNet and its different version, Vgg google net and Alex net. Sixth section deals with the noise of an image, explains about different type of noise in an image such as salt and pepper noise, gaussian noise and speckle noise. The seventh section explains about the methodology of the work and the results obtained and finally last section mentions about the conclusion

II. LITERATURE REVIEW

To understand more about the concept of deep learning and its techniques, first we have to understand the basis of the deep learning and its application. Since we are focusing on classification of diseases in bell pepper using pretrained CNN architecture, our literature review mainly focuses on CNN architecture and the application of CNN architecture such as object classification. The paper [1] explains about deep learning concept in a detail way. Paper explains about the basic concept of deep learning and its different techniques. Analyzing of paper [1] make us to under the beginning of the convolutional neural network. The main achievement of deep learning is the feature extraction. In Deep learning the feature extraction is done automatically, so the human effort is reduced compare to machine learning. It also explains about the architecture of deep learning. The architecture of deep learning consists of multi-layer structure in which first layer extract the lower features and the last layer extract the higher features. Main technology in the deep learning network is the convolutional neural network. Paper [2],[3] mentions the convolutional neural network and its application. In this paper the author tries to give the detail explanations about the CNN structure. The deep learning concept was most popular only because of the CNN model. The Main advantage of the CNN model is that the features are extracted automatically. This paper will give more detail analysis of the CNN architecture and its different architectures such as Alex net, Resnet, Google Net and so on. Our Paper focus more on the CNN architecture and the image classification, so, our literature part is pointed toward this area. Since the aims of the research work is to identify the diseases detection in bell pepper. Paper [4],[5] explains about the importance of computer vision in agriculture. This paper provides the ideas of the image processing and the machine learning in diseases classification. The papers show the importance of large amount of image for diseases classification. Paper [6],[7] mentions about the diseases identification in plant using artificial neural network and machine learning algorithm. Paper [6] mentions about the disease's classification with the help of segmentation, means the background image and diseases part of the leaves are separated, forming a variable parameter. With the help of GLCM the features of diseases are evaluated. The diseases are classified using tricolor mean values. Paper mentions about the usage of back Propagation for classification of different type of disease. Paper [7] mentions about the classification of disease using machine learning algorithm. Disease is identified in the leaves based on the color changes and shape. Paper [8],[9],[10] gives the detail ideas about the classification of diseases using ANN. In paper [8],[9] the input image is converted into HIS color space. After this conversion, paper [8] mentions about the clustering of image using squared Euclidean distance, thus the images are clustered into different group and evaluate the color and features of the image. Then it is statically evaluated and best features are obtained. Using the neural network, the image is classified. Paper [9] mention the usage of threshold method for feature extraction of the image. Classification of image is done by using self-organizing mapping, the author mentions that the purposed work is simple and giving average classification result but the purposed work does not provide a better result from the original image when the image is converted into frequency domain. Paper [10] deals with the classification of disease image, the image is classified using multi-layer perception. Here the color features are extracted and calculate its region of interest. The purposed model shows better result for some images but not for all. Paper [11], [12] mentions about the Navie based classification, here the author extracted the color features using normalized histogram and shape of the image using SURF and SIFT algorithm. The image is classified based on KNN, Navie base and SVM. The paper conclude that the Navie based classification shows better classification results as compare to other classification algorithm. This Navie based classification algorithm is used in the construction of google Net CNN architecture. Paper [13] mentions about the usage of fuzzy logic in disease classification. The mean and standard deviation of the image are identified and based on the fizzy rule the image is classified as disease or not. Paper [14] mentions the importance of combined classifier for diseases identification, here the author uses three different types of SVM classifier and threshold method is used for segmentation of the image. The color, shape and texture of the image are extracted and used as input for training. Three SVM are used in different stage of the work, first SVM is used in the low-level features and followed by the midlevel features and the last the high-level features. This method shows high success rate as compare to previous method. Paper [15],[16] mentions about the CNN and its applications of image detection. Paper mentions about the importance of CNN in plant disease detection and conclude that deep learning algorithm shows better prediction as compare to machine learning algorithm. Paper [17],[18] uses different CNN architecture for the disease's classification in plant leaves. Paper [18] compare the disease identification in Apple using multi-layer perception and Alex net and found out that Alex Net shows better performance as compare to another model. Same way the paper [17] uses Lenet CNN architecture and conclude that the LeNet shows better performance. While we analyzing this paper, we can conclude that the CNN architecture performs better as compare to Machine learning algorithm also while comparing Lenet with Alex Net we conclude that Alex Net has better performance. Paper [19] explains about the pretrained architecture of CNN such as Alex Net and Vgg16. It explains the importance of pretrained model and also compare the performance of pretrained model with large amount of image. Paper concludes that model with large number of images have high accuracy and perform well as compare to small number of images. Paper [20] explains about the different architectures of CNN and also mentions about the emergence of deep learning and its application.

III. PRETRAINED DEEP LEARNING NETWORK

[21,22] explains about the usage of transfer learning. Pre trained network in deep learning is defined as the networks that are trained to extract most powerful features from the images and use these features to learn new tasks. In most of the cases, the network is trained with ImageNet database which is used in ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [23]. The networks is trained with large amount of images having a class of thousand. Emerging trends of pretrained models helps to make the process faster and easier as compared to the model that are created from the scratch. The pretrained networks are generally used for classification, feature extraction and transfer learning. In classification problems pretrained network are directly used for classification of an image. Each pretrained network has different characteristics. The most important characteristics of pretrained models are network accuracy, size, and speed. The network is selected on these characteristics. A network is considered to be good if it has high accuracy and speed. The accuracy of the model is measured based on the classification accuracy. A network with good classification accuracy will function properly on a new task because of its learning capability. Feature extraction is done when we are using a low-cost system, means if we have no GPU facilities. In feature extraction, there is need for training all networks. The features of the image are extracted using pretrained model and the extracted features are given to train the Machine learning classifier. The feature extraction is used when we have small data set which are similar to pretrained network. Last purpose of pretrained deep neural network is the transfer learning. Transfer learning is the process in which a trained network which is used for one task can be used for another task. By the concept of transfer learning, training process in deep learning become faster and easier as compared to the architecture that is developed from scratch. Transfer learning concept is mainly used for object detection, image classification, video and audio detection and classification. Choose of transfer learning model is basically depend on the size, efficiency and simple nature. Selection of pretrained model includes the selection of simple model such as google net, Alex net and so on. To Classify the new task, some layers in the transfer learning model are replaced. This means that for new purpose, the fully connected layer and out classifier layer of the CNN architecture are replaced in the new task. This replacement means number of classes in the new task. Optically freezing the weight include the freezing the weight of previous layer, this is done by assigning the learning rate to zero. Freezing of the weight also helps to prevent overfitting issue that are caused due to small amount of data set. Retrained model includes the updated model with new features of the image. In most of the cases, the retrained model requires less amount of data than training a model from scratch. Final stage of work flow includes the predict and asses the new network accuracy. It includes the performance of network with new image. Figure 1 shows the work flow of transfer learning model.

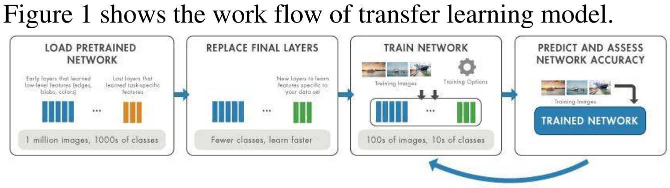


Fig 1. Work flow of transfer learning, taken from MATLAB

IV. MAIN COMPONENTS OF CNN ARCHITECTURE

Paper [20,23,25] mentions about the important componence in CNN architecture. It includes convolutional layer, pooling layer, activation layer, fully connected layer and loss function. In addition to this it also consists of filters, stride, optimization, padding and hyper parameter tuning. In this section we will explain about these CNN components. Before we begin with CNN, let us take the example of human brain. Human brain is divided into three different parts such as mid brain, hind brain, forebrain. Fore brain consist of cerebral cortex which consists of visual cortex, the visual cortex consists of different layer such as V1, V2, V3, V4 and V5. V1 layer is responsible for finding the edges of the images. V2 layer responsible for gather additional information than V1. Similarly, V3, V4 and V5 also gather some information and perform operation and thus we classify the image. In the case of convolutional neural network, filters are used to perform these operations.

A. Convolutional

Image is generally represented as pixel. Suppose an image is 6*6 pixel, if the image is grey scale, then values of the pixels range from 0 to 255. In the case of color image, image is represented as three channels and is denoted as 6*6*3. Values of pixels in the color images are between 0 to 255. To normalize the pixel, we use min-max scaler and thus the pixel values of the image is reduced. Consider an image which is represented as 6*6 matrix as follows

$$\begin{bmatrix} 0 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$$

Here 0 denotes white and 1 denotes black. Then there is a filter, which perform the function similar to the different layers of visual cortex of the human brain. 3*3 filters are represented as follows

$$\begin{matrix} +1 & 0 & -1 \\ +3 & 0 & -3 \\ +1 & 0 & -1 \end{matrix}$$

The above mention filter is a vertical edge filter. When we apply convolution operation to above matrix, we will get a matrix with 4 *4. In convolutional operation, filter is placed above image and multiplication operation is performed. Here we use a stride of one, which means after one multiplication operation we move one step right and this process is repeated until the operation is completed. Here the result is express as following matrix.

$$\begin{bmatrix} 0 - 9 - 9 \ 0 \\ 0 - 9 - 9 \ 0 \\ 0 - 9 - 9 \ 0 \\ 0 - 9 \ - 9 \ \ 0 \end{bmatrix}$$

From the above operation, we can generate a formula for the pixel value of an output. Formula is

$$O = N- F+1 \qquad (1)$$

Were

O = output size

N= pixel value of an image

F= filter size

Here, we can see that for 6*6 image size we get an output of 4*4, means there is a loss of information, to avoid this we can use padding [24] . Padding is the process of adding a layer to the image. For example, in 6*6 image if we use padding of size one, then there occurs an addition of one row upward and one row downward, one column left and one column right. Now the image pixel size is changes to 8*8. Then applying a filter, we get an output of size 6*6. Equation (1) is changed to

$$O = N+2P-F+1 \qquad (2)$$

Were

O = output size

P= Padding

N= pixel value of an image

F= filter size

In CNN architecture, there are many filters used with different size, it can be either vertical filter or horizontal filter, and also, in CNN values of filters are updated based on the back propagation. Initially the values of the filters are selected based on random ways. Activation function is applied to each value of the output. The generated result is the output of the convolutional layers.

B. Pooling Layer

Paper [25] mentions about the pooling layers. In CNN architecture the pooling layer is place after the convolutional layer, which helps to down sample the output of the convolutional layer. It is done by sliding the filter and calculate its average or maximum. There are different types of pooling, such as Max-pooling, Min Pooling, and average pooling. Max Pooling and average pooling concept can be explained with an example. Suppose if we have an output feature of 3*3 and filter of size 2*2, when we place filter to output feature of convolution operation, in Max-pooling for every operation we take the maximum value. Output feature of convolution operation can be represented as

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

Here, we take stride of two, in Max-pooling during the first operation we take the maximum value from the feature map, here it is 5. Since we choose the stride as two, so it shifts to two position, same previous operation is done and maximum value is found out. final output is as follows

$$\begin{matrix} 5 & 6 \\ 8 & 9 \end{matrix}$$

Thus, in Max pooling operation will take maximum intensity value from the output of the convolutional layer. So, in case of images, after Max pooling, features of the image is detected properly. In the case of Min pooling and average pooling only change is that the final value will be minimum value for min pooling and average value for average pooling. The value of the filter will be updated after the back propagation and this process will continues until we get the correct result.

C. Fully connected layer

The output of the convolutional layer or in some case pooling layer will be connected to the fully connected layer or dense layer. Here every input vector influences every output vector by learnable weight. In CNN architecture there are one or more fully connected layer. Final fully connected layer in the CNN architecture consists of nodes which is equal to the number of classes.

D. Optimizers

Paper [26] mentions about the optimizers in deep learning. The Optimizer is defined as the algorithm or a method that is used to change the neural network attributes such as weight and learning parameter so that loss in the network can be reduced and provides accurate result. Different optimizers used in CNN includes Gradient descent, Stochastic gradient descent, Mini batch, Momentum, Adagard, Adadelta, and Adam. Gradient descent is the basic optimizer in deep learning and it is used in classification and problem regression. It depends on first order derivate of loss function. Here the weight is adjusted to minimized the loss. Main advantage of gradient descent is its implantation, computational easy and simple nature. At the same time, it also faces some drawback such as it requires more memory and takes long time. Stochastics gradient is a variant of gradient descent and the parameter of the models are changed after computing the loss at each training. As the parameters are updated frequently it converges in less time. Also, this optimizer requires less amount of memory. Stochastic optimizer faces some issues such as high variance in model parameter. In Mini batch the parameter is updated at every batch so the model has less variations. In this optimization algorithm dataset is dived into batches. When the learning rate is too small it takes lot of time to converge. Momentum optimizer was developed to solve the problem of high variance and soften the convergence. Momentum optimizer accelerate towards relevant direction and reduces fluctuation toward irrelevant direction. Adam Optimizer combine momentum and RMS PROP. RMS Prop helps to change learning rate in an efficient manner. The Adam optimizer will converge rapidly and thus solve the issue of vanishing learning rate

E. Activation Function

Paper [27,28,29,30,31] mentions about different activation functions used in deep learning architecture. It defines as the function that is used to transform the input signal to output signal and is used as an input for next layer. A Neural network consist of input layer, hidden layer and output layer. All these layers have nodes and each nodes have weight and these are used when processing a network. Suppose, while processing a neural network, an activation function is not used, then the output will be a linear equation which is either a simple or complicated. This liner equation does not have the capability of learning a complex mapping from the image or data. A neural network without an Activation function will perform as a linear regression network or model with limited performance. While designing a neural network, the main aim of the developer is to develop a network which can able to perform complex task with help of activation function. Activation function can be divided into two class, linear and nonlinear. In the CNN architecture we use the nonlinear activation functions . There are different type of activation function such as sigmoid, tanh, Sigmoid-Weighted Linear Units, Derivative of Sigmoid-Weighted Linear Units (dSiLU), Rectified Linear Unit (ReLu) Function, SoftMax function. Here we will discuss more about ReLu and SoftMax. Paper [30] mention about the ReLu activation function. It was invented in the year 2010, and widely used in CNN architecture as the activation function. It performs better as compare to other activation function, shows a linear function, faster computation and sparsity in hidden unit. Thus, ReLu activation function is easy to optimize with gradient decent method. The Activation function ReLu is given as

$$F(Y) = \max.(0, Y) \qquad (3)$$

The activation function also faces some issues such as overfitting issue. In training, the ReLu become fragile and thus neuron will not be active as a result there will not be any weight updating in future data. To solve this problem the leaky ReLu was introduced. In the case of leaky ReLu a small negative slope is introduced to the ReLu structure such a way that the weight updating is active throughout the training process [31]. The alpha parameter was introduced in the ReLu activation function to overcome the issue of dead neuron. The function of Leaky Relu is given as

$$\begin{aligned} f(y) &= 0.01y, & y &< 0 \\ f(y) &= y, & y &>= 0 \end{aligned} \qquad (4)$$

where $\alpha$is the alpha, which is constant value.

Parametric Rectified Linear Units (PReLU) is another type of ReLu activation function. Here negative slop is not fixed and varies with respect to y equation of PReLU and is given as

$$\begin{aligned} f(y) &= \alpha y & y &< 0 \\ f(x) &= y & y &>= 0 \end{aligned} \qquad (5)$$

This activation function will overcome the problem of gradient of ReLu equal to zero in case of negative values. The alpha value is learnable while training with back propagation. Exponential Linear Unit (ELU) is a type of ReLu. In ELU a parameter slope for the negative values of x is introduced and uses a log curve for showing negative value. The mathematical expression for ELU is given as

$$\begin{aligned} f(y) &= y & y &>= 0 \\ f(x) &= \alpha\,(e^y - 1), & y &< 0 \end{aligned} \qquad (6)$$

Soft max function is another type of activation function. It is the combination of sigmoid activation function and used in multi classification problem. It is generally used when the classes are mutually exclusive. The formula for soft max is given as

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad (7)$$

Table 1 shows the Activation function that are used in different CNN architecture

Table 1. Activation function

CNN Architecture	Activation Function
Vgg16	ReLu
Vgg 19	ReLu
Resnet 18	ReLu
Resnet50	ReLu
Resnet 101	ReLu
Alex Net	ReLu
Google Net	ReLu

F. Loss Function

It is used to check the performance of algorithm on the data that we choose. In Mathematical terms the loss function is defined as the difference between predicted value and the true value. It is mainly used in training process to check the fitting of training data. The output of the loss function is used as a feedback signal for weight adjustment. This makes the loss function more important for the optimization process. While modelling an architecture it is very important to choose a proper loss function for the better performance of the model. There is different type of loss functions available based on the problem that we choose. In Regression type problem loss functions are Mean square error, Mean absolute error, Huber loss. In classification type problem, loss function used here is

Binary cross entropy, categorical cross entropy, hinge loss. Similarly for Auto encoder we choose KL divergence as the loss function and lastly for the object detection we choose focal loss and triple loss. Our work is to classify diseases presented in the bell pepper plant,paper explains about some of the loss function that are used in the classification problem. Binary cross entropy loss function is used in binary classification problem where the main aim of the loss function is to predict two possibilities of outcome. It calculates the difference between predicted probability and true labels. Mathematical expression for binary cross entropy is given as

$$L(Y_{predicted}, Y_{true}) = -Y_{true} * \log(Y_{predicted}) - (1- Y_{true}) *\log(1- Y_{predicted}) \quad (8)$$

Binary cross entropy is differentiable and easy to interpret. It also faces some drawback which include, cannot be used for multi class classification problem and highly dependent on outliers. In the case of Binary cross entropy loss function, activation function should be sigmoid. Categorical cross entropy loss function is mainly used in multi classification problem. It measures the difference between predicted class probability and true labels. Mathematical expression of Categorical cross entropy loss function is given as

$$L (Y_{predicted}, Y_{true}) = \text{sum over all class } (Y_{true} * \log (Y_{predicted})) \quad (9)$$

Here the targeted vector is represented as one hot encoded vector. Each class is represented as binary, in case of correct class it is represented as 1 and for other it is represented as 0 Hinge loss function is another type of loss function that is used in the classification problem where SVM model is used. It is used in binary classification problem.

V. PRETRAINED CNN ARCHITECTURE

This section explains about the pretrained CNN architecture such as Alex Net, Google Net, Resnet 18, Resnet 50, Resnet 101, Vgg16, and Vgg19

A. Alex Net

[32-36] mentions about the Alex Net architecture. It consists of five convolutional layer and three fully connected layer. Alex Net architecture mark a significant way to the convolutional neural network as it introduces ReLu activation function in CNN architecture. Introduction of ReLu activation function helps to make CNN architecture faster. It also gives solution to overfitting issues in the CNN architecture, by introducing the dropout layer. Figure 2 shows the general architecture of Alex Net

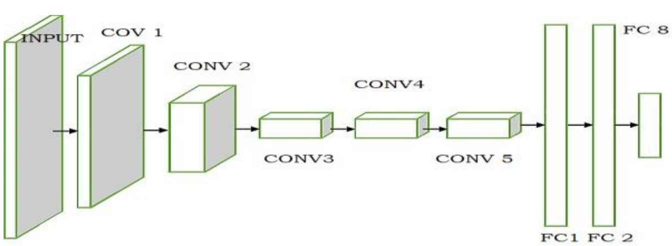


Fig.2 Alex Net Architecture

Alex Net uses an input image of size 227. Input image pass through the convolutional layer plus ReLu and produce an output, which is pass through Max pooling layer and moves to another convolutional layers plus ReLu. Each convolutional layer used in the Alex Net is of different filter size. Max pool layer has a stride of two. From the above process we can say that, as the filter number increases the feature extraction capability also increases and vice versa. Drop out layer used in the Alex Net activate fifty percentage of neurons. Depend on the size of the image the fully connected layer extract the features. Main advantage of Alex Net architecture over other convolutional network is the feature extraction. The convolutional layers extract the edge of the image or data and fully connected layer will learn the image or data. Same way the Alex Net also faces limitations such as depth issue, use of large CNN filters. Compare to other architecture depth of Alex Net is less.

B. Vgg 16

[32,35-37] explains about the Vgg 16. Vgg 16 consist of 13 convolutional layer and 3 fully connected layer, followed by Maxpooling layer and a SoftMax layer. Vgg 16 has large size of architecture (515 MB) and has large number (138 M) of learnable parameter. Vgg 16 provides a stable and high accuracy result in classification problems. Figure 3 shows the general architecture of Vgg 16.

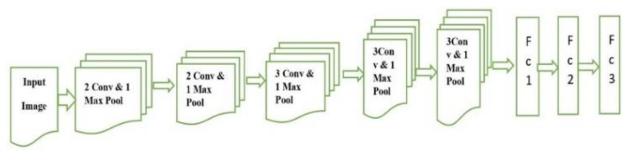


Fig 3. Vgg 16 Architecture

In Vgg 16 architecture input is of size 224*244 and the filter used here is of fixed size 3*3. The output of the convolutional layer will pass through a max pool layer having a stride of two. There are two set of filters with size 512, output of these convolutional layer will pass through a Maxpooling layer of stride two, fully connected layer will extract the feature and last it will classify the image. The main draw backs include, slow training time and large network size of Vgg 16 is large so it requires large amount of space.

C. Vgg19

[38,39] mentions about the Vgg 19 architecture. Vgg 19 architecture is similar to Vgg 16 architecture with addition of three CNN layers. Figure 4 shows the architecture of Vgg 19. Image size used in Vgg 19 is of 224*224. The convolutional layer used here is of same size of 3*3. Each output from the convolutional layer will go to the Maxpooling layer and finally go to fully connected layers. The additional part of this convolutional layer made the architecture bigger in size and increases the learning parameters as compare to Vgg 16.

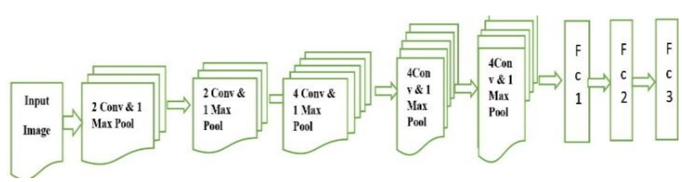


Fig.4 Vgg 19 architecture

Compare to Vgg 16, Vgg 19 increase the depth but still faces the issue of large network size and slow training time.

D.Resnet 18

[40-43] gives a detail idea about the Resnet 18 architecture. It consists of 17 convolutional layer and one soft max layer, which is shown in figure 5. Similar to Vgg 19, Resnet 18 uses the filter one of size 3*3. The down sampling process in the Resnet 18 is done by the convolutional layer of stride two. Pooling layer used in CNN structure is average pooling and a fully connected SoftMax layer is also present. There are two connection presents in the architecture. One connection is used when input dimensions is equal to output dimensions similarly the other connection is used when the dimensions of the image is increased [42].

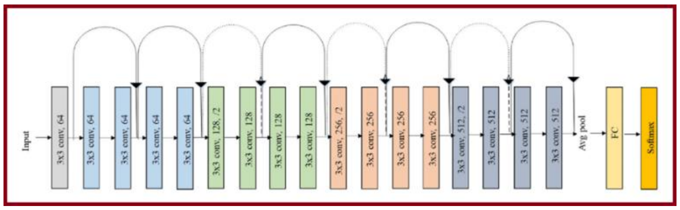


Fig. 5 Resnet 18 architecture

E. Resnet 50

Resnet 50 is another version of residual network, [44,45] which consist of 50 layers. It uses convolutional layers of 1 *1, 3*3 and 1*1. It has five convolution stages which includes stage 1 consists of single block of layer followed by stage 2 consist of block of size three, stage 3 consist of block of four, stage 4 consist of block of six and stage 5 consist of block of three. Block of each stage consist of convolutional layers of size conv (1*1), conv (3*3), and conv (1*1). This architecture uses average pooling. With Average pooling and down sampling, size of the feature map is reduced. In addition to above layer there is a fully connected layer which helps to classify the image. Including these fully connected layers, total of 50 layers is present which is called as ResNet 50. Figure 6 shows the general architecture of Resnet 50

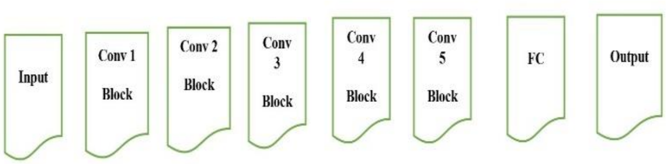


Fig.6 Resnet 50 Architecture

F. Resnet 101

[46-48] mentions about the detail study of ResNet 101 architecture. Resnet 101 consist of 99 convolution layers and average pooling layer. For the development of building block convolutional of size 1*1 is added. It has five convolution stages, stage one consists of single block of layer followed by stage two consist of block of size three, stage three consists of block of four, stage four consist of block of twenty-three and stage five consist of block of three. Block of each convolutional stage consist of convolutional layers of size conv (1*1), conv (3*3), and conv (1*1). Figure 7 shows the general architecture of ResNet 101

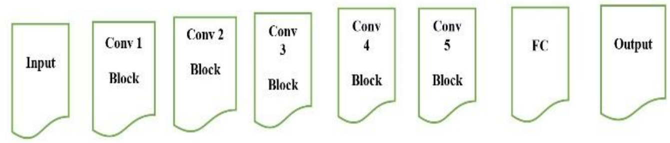


Fig.7 Resnet 101 Architecture

When we compare the architecture of Resnet 101 and Resnet 50, the only difference is the conv 4 block where it consists of 23 convolutional layers in 101 ResNet, where as in case of ResNet 50 it consists of only six convolutional layers. Resnet 101 has a depth of 101 whereas the Resnet 50 has a depth of 50.

G.Google Net

[49-53] mentions about the google Net architecture, which was developed by the researcher of google. It consists of depth 22, having a pooling layer of twenty-seven along with inception module of nine. These modules are connected with average pooling layers. Max pooling layers are also used here. Activation function used here is ReLu and the architecture consist of forty percentage of dropout. Figure 8 shows the general structure of google Net. Initial convolutional layer is used for resizing an input image, followed by the second convolutional layer which is used for dimensionality reduction. The inception module contains the max pooling layer which down sample the image. Average pooling layer presented in the architecture calculate the mean average from the output of the last inception module. The drop out layer helps the architecture to prevent from overfitting.

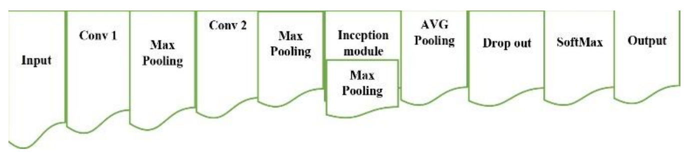


Fig.8 Google Net architecture

VI. NOISE IN IMAGE

Noise in an image is defined as the variation in the color or pixel value of an image. It is caused due to the external factors of the system or the environment. The noise can be added to the image due to several factors such as transferring of image to electronics device, some factors of heat while capturing an image, and the variation in specification of camara. There is different type of noise such as gaussian noise, Salt and pepper noise, Poisson noise and speckle noise. Our Model mainly focus on Gaussian, salt and pepper and speckle noise.

A. Gaussian Noise

It is also known as electrical noise and is defined as the noise having the probability density function which is similar or equal to normal distribution. Noise is implemented in the image by using random gaussian function. Figure 9 shows the gaussian noise in an image

(a) *(b)*

Fig.9 (a) Original image (b) gaussian noise

B. Salt and pepper Noise

It is a type of impulse noise, which is added to the image by the addition of random dark and bright values to the image. It also is known as data drop noise. Fig 10 shows the salt and pepper noise along with original image

(a) *(b)*

Fig.10 (a) Original image (b) salt & pepper noise

C. Speckle Noise

It is a granular noise that affect the quality of the image. It is generated in an image by multiple random value of pixel with different pixels. Figure 11 shows the speckle noise in an image.

(a) *(b)*

Fig.11 (a) Original image (b) speckle noise

VII. PROCEDURE OF THE WORK

The Procedure of the work is explained in Figure 12. It includes the collection of images, preprocessing, training with different CNN architecture, addition of noise to testing image, testing of CNN architecture based on noise and without noise and finally analyzing the performance of different CNN architecture based on the performance, while testing and training.

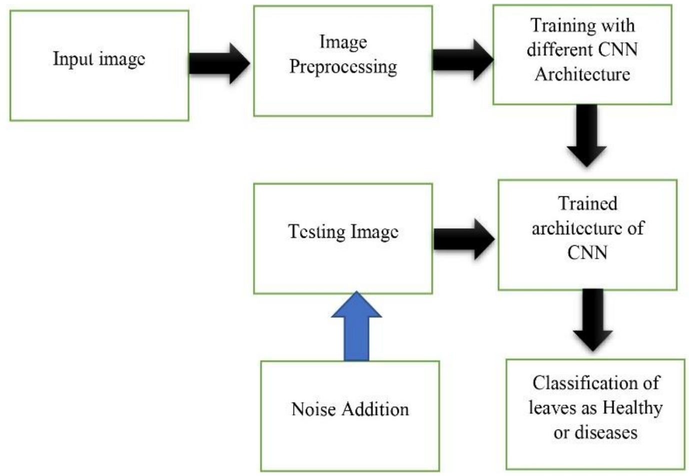


Fig.12 Block diagram of purposed model

A. Input image or data set collection.

The major challenge that we face while doing the project is the collection of images. Collection of large amounts of image from the farm with camara is a time-consuming task. So, we consider secondary data set which is also known as bench mark dataset which is available in plant village on Kaggle site. Figure 13 shows the sample of the dataset which are used for training.

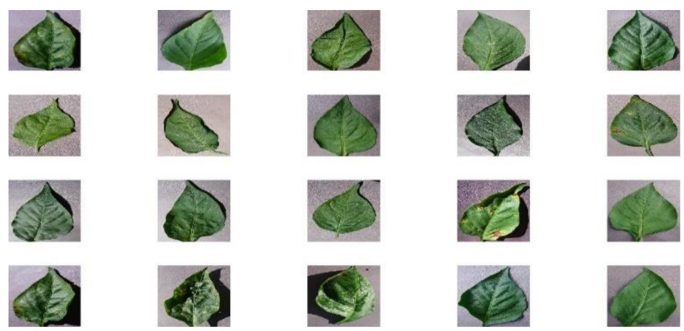


Fig. 13 Sample image for Training CNN Architecture

B. Image Preprocessing

In Image preprocessing, collected images are analyzed and changes are made if needed. Image that we collected from plant village data set is of size 256*256.In our project work we use folder wise labelling. We create two folder and name it as bacterial spot and healthy and fill this folder with bacterial spot diseases and healthy leaves images. Also, we make sure that the data set is balanced. Table 2 shows the class number that are used for classification of diseases. The data set is divided into training and validation, for our work we choose 75% of data for training and 25 % for validation

Table 2. Classes used in the Project work

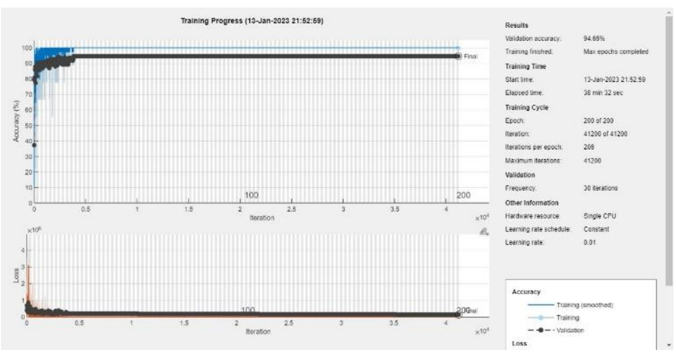
Class No	Class Name
0	Bacterial spot
1	Healthy

C. Training of Model

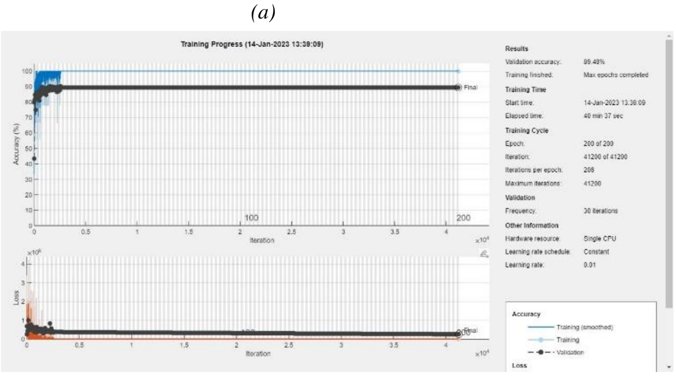
We done our project work on MATLAB 2022b version in single core CPU hardware and single core GPU hardware. The Project work was done on two different phases, one is training of image without augmented and other is with augmented. To identify which architecture is more suitable for the classification of disease in bell pepper plant, we trained the dataset with different CNN model such as google net, Alex net, ResNet 18, 50,101 and Vgg 16, vgg19 with above mention phases. For training of different architecture, we choose the hyperparameter as follows.

- Choose the optimizer as stochastic gradient
- Set minibatch size as 9
- Make the verbose false
- Make validation frequency as 30 for without augmentation and 206 for augmented.
- Make initial learning rate as 0.01 for without augmented and 1e* 10^{-4} for augmented
- Training data is shuffle at each epoch

Training process is done with above parameters and also with augmented data and without augmented data and the performance of different CNN architecture is analyzed. Based on the analysis we found out that the Vgg 19 and google Net are the best architecture for the classification of diseases in bell pepper. Figure 14 shows the training process of the best architecture without augmentation.



(a)



(b)

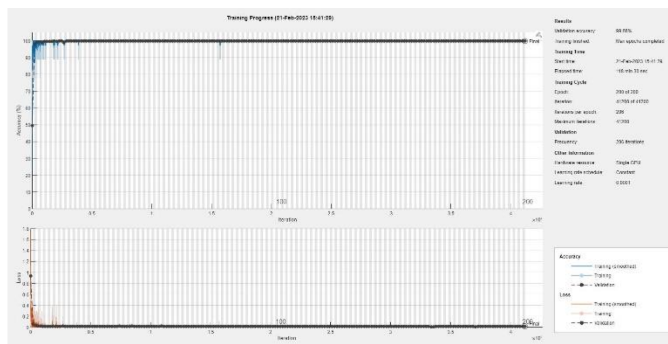
Fig.14 Training process of (a) Vgg 19 (b) Google Net architecture without augmented

Table 3 shows the validation accuracy and time for training different CNN architecture without Augmentation. Here we use maximum iteration of 41200 along with iteration per epoch 206 and constant learning rate schedule. Hardware is single core CPU

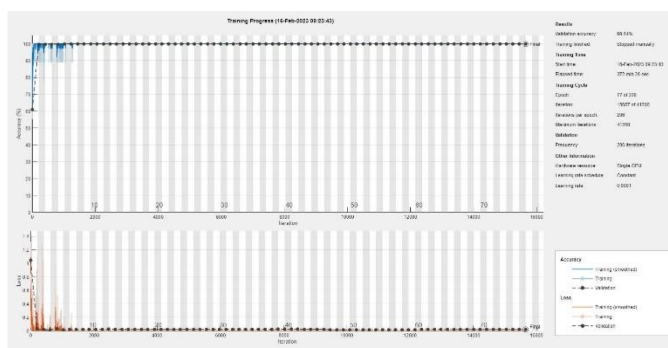
Table 3. validation accuracy of different pretrained model

CNN Arc.	Vali Acc.	Train. Time (min)	Hardware	Max. Iteration	Iteration/epoch
Alex Net	91.42	40.57	CPU	41200	206
Google Net	89.48	40.37	CPU	41200	206
Resnet 18	91.10	39.38	CPU	41200	206
Resnet 50	91.26	36.51	CPU	41200	206
Resnet 101	91.10	36.51	CPU	41200	206
Vgg 16	90.13	37.40	CPU	41200	206
Vgg19	94.66	38.32	CPU	41200	206

Similarly, figure15 shows the training process of google net and Vgg 19 architecture with augmentation. In augmented training process we use hardware of single core GPU.



(a)



(b)

Fig.15 Training process of (a) google net (b) Vgg 19 architecture with augmented

Table 4 shows the validation accuracy and time for training different CNN architecture with Augmentation. Here we use maximum iteration of 41200 along with iteration per epoch 206 and constant learning rate schedule. Hardware is single core GPU

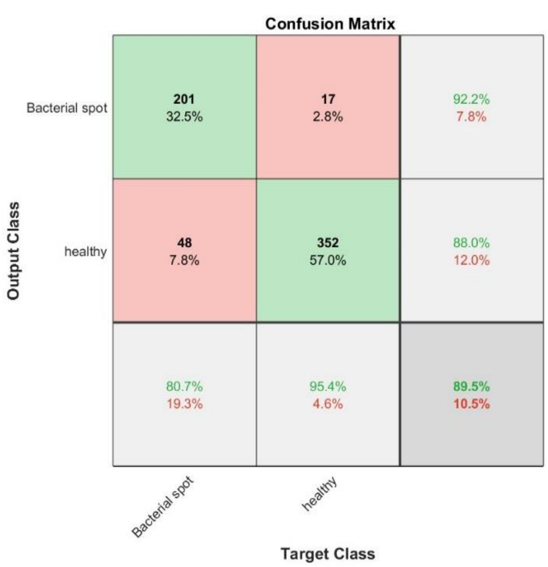
Table 4. validation accuracy of different pretrained model

CNN Arc.	Vali Acc.	Train. Time (min)	Hardware	Max. Iteration	Iteration/epoch
Alex Net	99.84	83.57	GPU	41200	206
Google Net	99.68	118.30	GPU	41200	206
Resnet 18	100	106.17	GPU	41200	206
Resnet 50	99.84	291.41	GPU	41200	206
Resnet 101	100	559.32	GPU	41200	206
**Vgg 16	99.84	1443.4	GPU	41200	206
**Vgg19	99.84	372.26	GPU	41200	206

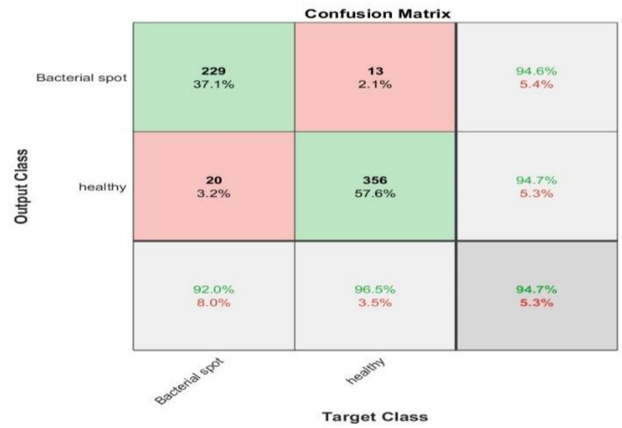
** Comparing with other network Vgg 16, 19 takes large time to finish. Here we manually stop the training process. In case of Vgg 16 the training process is stopped at the epoch of 138 and performance is evaluated. Similarly, the Vgg 19 the training process is stopped at the epoch of 77. In both the cases the architecture reaches the saturation level.

D.Testing of CNN Architecture and Result analysis

To know the performance of the architecture in identifying the disease, testing is important. We have taken the images which is available in google images for testing the CNN architecture. Firstly, we select images for testing, after that we change the size of the image based on the requirement of the CNN architecture. After resize, the image we evaluated the image using different CNN architecture and the performance is noted. It was done for both augmented and without augmented. Performance of the CNN architecture is evaluated by plotting the confusion matrix and classification capability. Figure 16 shows the confusion matrix of google Net and Vgg 19 architecture that are used for classification of diseases without augmentation.



(a)



(b)

Fig16 Confusion matrix of (a) Google Net (b) Vgg 19 without augmentation

Figure 17 shows the confusion matrix of google Net and Vgg 19 architecture that are used for classification of diseases with augmentation.

(a)

(b)

Fig17 Confusion matrix of (a) Google Net (b) Vgg 19 with augmentation

From the confusion Matrix we can calculate the accuracy, precision, recall, specificity and F1 score of different CNN architecture. This can be calculated by the following mathematical equation.

$$Accuracy = (TP+TN)/ TP+TN+FP+FN \quad (10)$$
$$Recall = TP / (TP+FN) \quad (11)$$
$$Specificity = TN/(FP+TN) \quad (12)$$
$$Precision = TP/(TP+FP) \quad (13)$$
$$F1\ score = 2*TP/(2*TP+FP+FN) \quad (14)$$

Were

TP = True Positive

TN = True Negative

FP = False positive

FN = False Negative

Figure 18 shows the accuracy, precision, recall, specificity, F1 score of different CNN architecture, without augmentation.

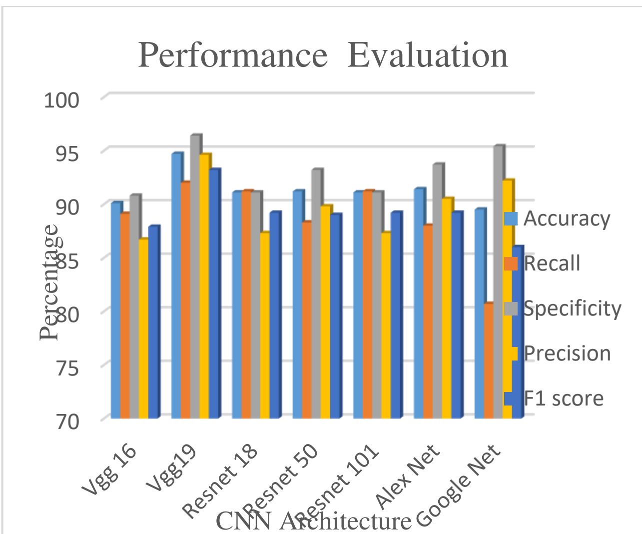


Fig 18. Performance evaluation of different CNN architectures without augmented

Figure 19 shows the accuracy, precision, recall, specificity, F1 score of different CNN architecture with augmentation.

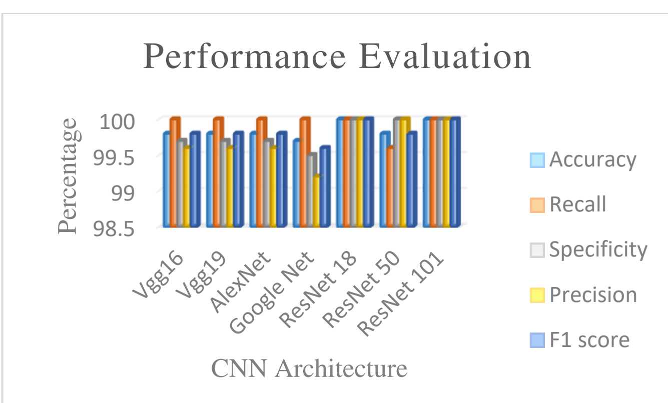
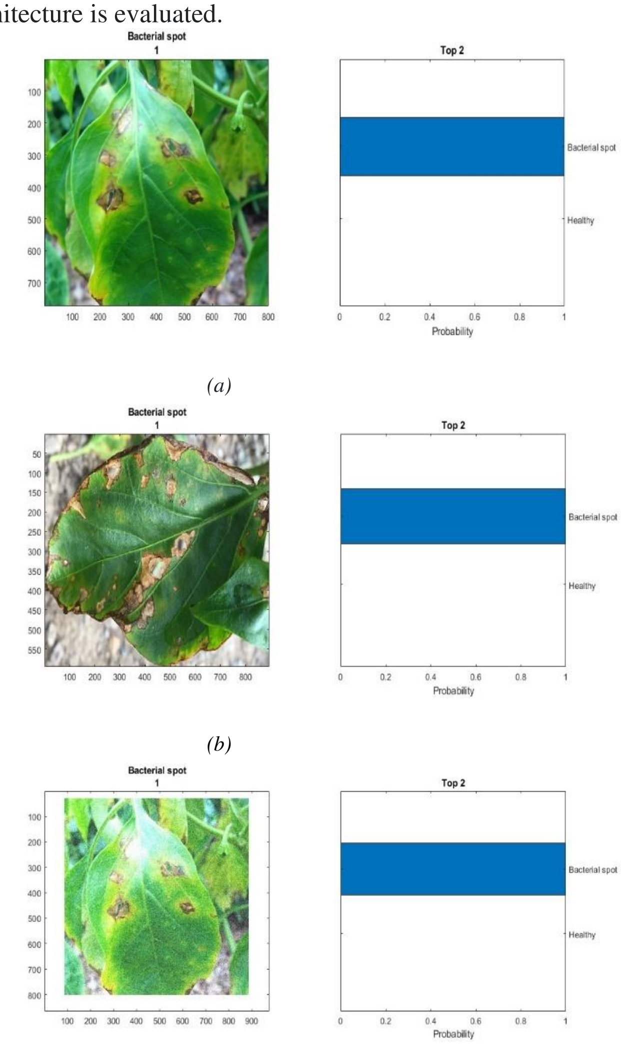


Fig 19. Performance evaluation of different CNN architecture with augmented

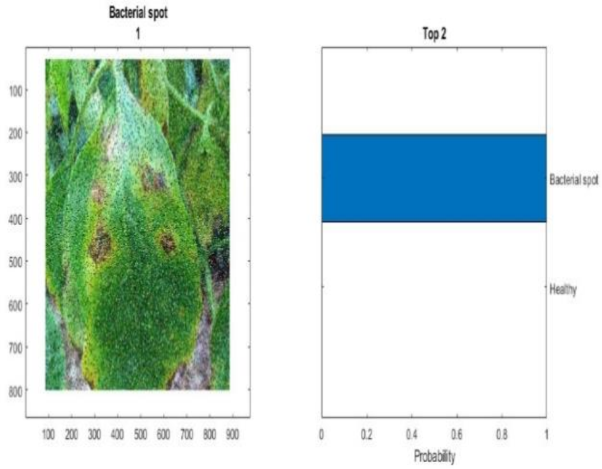
When we evaluated the architecture with augmentation, we found out that Resnet network undergoes the fitting issues and the images are classified wrongly but the accuracy and the other parameter of the ResNet architecture is very high as compare to other architecture. Performance of the architecture are in the range of 99 to 100 percentage and the difference is the classification capability of the CNN architecture. Based on our study we found out that the google Net and Vgg 19 shows the better performance, still these architectures also face some problems. Main problem associated with the Vgg 19 is the training time. In case of augmented data the training time is very large which is mentions in the table4 and the size of the architecture is very high. This problem should be considered in the future for the development of the CNN architecture. In case of google Net the training time is less as compared to Vgg 19. Google Net architecture is lightweight, so it requires less amount of space for the implementation and the computational cost is also less. But it faces the issues of accuracy, recall, precision, specificity and F1 score. Compare with Vgg 19 it has less performance which should be consider in future. Figure 20 shows the tested image result that we get in the augmented image. Here images are taken in random way and the gaussian, salt and pepper and speckle noise is added and evaluated the classification capability of the CNN architecture is evaluated.



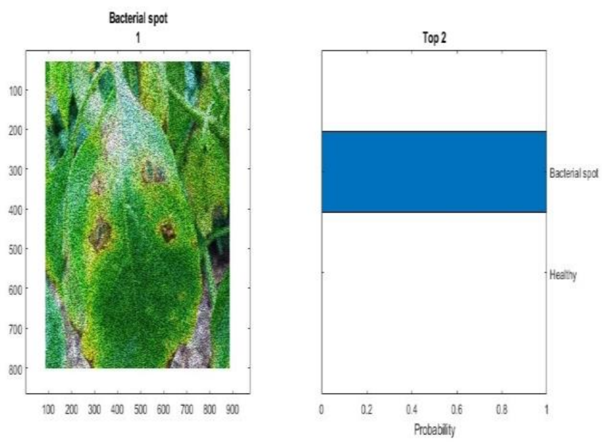
(a)

(b)

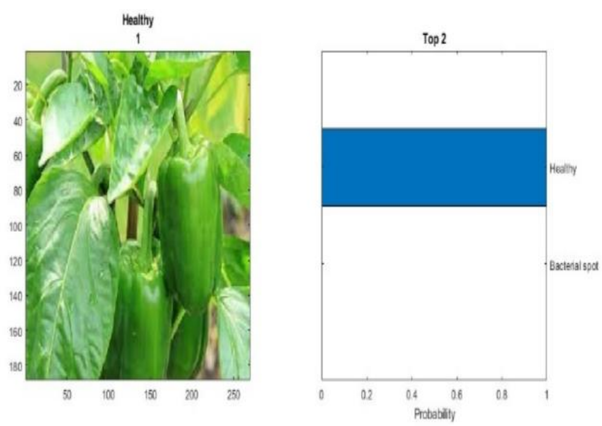
(c)



(d)



(e)



(f)

Fig. 20 Tested result that we obtained in google Net architecture

(a), (f) is the image tested in a normal condition, (b) is the image with low quality and (c), (d) and (e) is the tested image in noise condition. Similarly, the images are also tested without augmented condition. In low quality and in Gaussian noise condition capability of the Vgg 19 and google Net for classification is low which means the architecture was classified wrongly. In this case the performance evaluation parameter is also low. Among this architecture we analyzed, we came to a conclusion that the Vgg 19 and Google Net performs well as compare to other CNN architecture. Even though Vgg 19 and Google Net architecture performs well, but their performance evaluation parameter such as accuracy precision, recall, and specificity is low and it should be addressed in future.

VIII. Conclusion

In this project, we analyzed different pre-trained CNN architectures such as Vgg 16, Vgg 19, Resnet 18 Resnet 50, Resnet 101, Google Net, and Alex Net in different conditions such as augmented condition, without augmented condition, Noise condition, and low-resolution condition. Based on our study, we concluded that Google Net and Vgg 19 show better performance. Thus, we concluded that Google Net and Vgg 19 are better for the classification of diseases in bell pepper. The purposed work will help the bell pepper farmers to identify diseases at an early stage and will help the farmers to provide solutions to the diseases. The identification of diseases at an early stage will help the framers to provide proper treatment and thereby reduce the further spreading of the diseases. The model provides a pathway for automation, which will help the bell pepper farmers and agriculture departments which are related to bell pepper and make their work easy. This project will help the researcher to analyze the different pre-trained models and identify the issues related to each pre-trained architecture and develop an architecture that can overcome the issue that is presented in the existing pre-trained architecture. The future development of the work includes the development of an architecture that has high accuracy, precision, recall, specificity and F1 score as compare to the Vgg 19 and Google Net in both augmented data and without augmented data and have high classification capability and should be lightweight. The development of an application for diseases identification will make a huge impact on the life of bell pepper farmers.

Declarations

A. Ethical Approval

It is not applicable to this work

B. Competing interest

On behalf of all authors, the corresponding author states that there is no conflict of interest

C. Authors Contributions

On behalf of all authors, the corresponding author states that all authors contributed to the study of concept and design.

D. Funding

On behalf of all authors, the corresponding author states that there is no fund received for this work

E. Availability of Data and materials

The dataset used in the work can be accessed from the Kaggle site and Google image

REFERENCE

[1] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44

[2] Yao G, Lei T, Zhong J. A review of convolutional-neural-network-based action recognition. Pattern Recogn Lett. 2019; 118:14–22.

[3] Dhillon A, Verma GK. Convolutional neural network: a review of models, methodologies and applications to object detection. Prog Artif Intell. 2020;9(2):85–112.

[4] S. S. Kumar and B. K. Raghavendra, "Diseases Detection of Various Plant Leaf Using Image Processing Techniques: A Review," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 313-316, doi: 10.1109/ICACCS.2019.8728325

[5] Dhingra, G., Kumar, V. & Joshi, H.D. Study of digital image processing techniques for leaf disease detection and classification. *Multimed Tools Appl* 77, 19951–20000 (2018). https://doi.org/10.1007/s11042-017-5445-8

[6] Huang K-Y (2007) Application of artificial neural network for detecting Phalaenopsis seedling diseases using color and texture features. Comput Electron Agric 57:3–11

[7] Aduwo JR, Mwebaze E, Quinn JA (2010) Automated vision-based diagnosis of cassava mosaic disease. In: Industrial Conference on Data Mining-W9orkshops, pp. 114–122

[8] Bashish DA, Braik M, Bani-Ahmad S (2010) A framework for detection and classification of plant leaf and stem diseases. In: IEEE International Conference on Signal and Image Processing, pp.113–118

[9] Abdullah NE, Rahim AA, Hashim H, Kamal K (2007) Classification of rubber tree leaf diseases using multilayer perceptron neural network. In: Fifth student conference on research and development (SCORed), Selangor, 11-12 December. pp. 1–6

[10] Phadikar S, Sil J (2008) Rice disease identification using pattern recognition techniques. In: 11th international conference

[11] Aduwo JR, Mwebaze E, Quinn JA (2010) Automated vision-based diagnosis of cassava mosaic disease. In: Industrial Conference on Data Mining-Workshops, pp. 114–122

[12] Dhingra, G., Kumar, V. & Joshi, H.D. Study of digital image processing techniques for leaf disease detection and classification. *Multimed Tools Appl* 77, 19951–20000 (2018). https://doi.org/10.1007/s11042-017-5445-8

[13] K, Latha P (2014) Fuzzy inference system based unhealthy region classification in plant leaf image. Int J Comput Inf Eng 8(11):2103–2107

[14] Tian Y, Zhao C, Lu S, Guo X (2012) SVM-based multiple classifier system for recognition of wheat leaf diseases. In: IEEE World Automation Conference, pp.189–193

[15] Kaya, A., Keceli, A. S., Catal, C., Yalic, H. Y., Temucin, H., and Tekinerdogan, B. Analysis of transfer learning for deep neural network-based plant classification models. Computers and electronics in agriculture 158 (2019), 20-29.

[16] Brahimi M, Boukhalfa K, Moussaoui A (2017) Deep Learning for Tomato Diseases: Classification and Symptoms Visualization. Applied Artificial Intelligence 31(4):1–17.

[17] Prajwala TM, Alla Pranathi, Kandiraju Sai Ashritha, et al. "Tomato Leaf Disease Detection using Convolutional Neural Networks". in: 2018 Eleventh International Conference on Contemporary Computing (IC3)

[18] Nachtigall LG, Araujo RM, Nachtigall GR (2016) Classification of Apple Tree Disorders Using Convolutional Neural Networks. In: 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), pp 472–476, DOI 10.1109/ICTAI.2016.0078

[19] Aravind Krishnaswamy Rangarajan, Raja Purushothaman,Aniirudh Ramesh, " Tomato crop disease classification using pre-trained deep learning algorithm", ELSEVIER, International Conference on Robotics and Smart Manufacturing (RoSMa2018).

[20] Alzubaidi, L., Zhang, J., Humaidi, A.J. *et al.* Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* **8**, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8

[21] Kaur, T., Gandhi, T.K. Deep convolutional neural networks with transfer learning for automated brain image classification. *Machine Vision and Applications* **31**, 20 (2020). https://doi.org/10.1007/s00138-020-01069-2

[22] Russakovsky, O., Deng, J., Su, H., et al. "ImageNet Large Scale Visual Recognition Challenge." *International Journal of Computer Vision (IJCV)*. Vol 115, Issue 3, 2015, pp. 211–252

[23] Muhammad Asif Saleem, Norhalina Senan, Fazli Wahid, Muhammad Aamir, Ali Samad, Mukhtaj Khan, "Comparative Analysis of Recent Architecture of Convolutional Neural Network", *Mathematical Problems in Engineering*, vol. 2022, Article ID 7313612, 9 pages, 2022. https://doi.org/10.1155/2022/7313612

[24] T. Haryanto, I. S. Sitanggang, M. A. Agmalaro and R. Rulaningtyas, "The Utilization of Padding Scheme on Convolutional Neural Network for Cervical Cell Images Classification," *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, Surabaya, Indonesia, 2020, pp. 34-38, doi: 10.1109/CENIM51130.2020.9297895.

[25] Yamashita, R., Nishio, M., Do, R.K.G. *et al.* Convolutional neural networks: an overview and application in radiology. *Insights Imaging* **9**, 611–629 (2018). https://doi.org/10.1007/s13244-018-0639-9

[26] Gousia Habib, Shaima Qureshi, Optimization and acceleration of convolutional neural networks: A survey, Journal of King Saud University - Computer and Information Sciences,Volume 34, Issue 7, 2022, Pages 4244-4268, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2020.10.004.

[27] Agostinelli, F., Hoffman, M., Sadowski, P., & Baldi, P. (2014). Learning Activation Functions To Improve Deep Neural Networks. Arxiv Preprint Arxiv:1412.6830

[28] D. Pedamonti, "Comparison of non-linear activation functions for deep neural networks on MNIST classification task,," arXiv, 2018. [Online]. Available: http://arxiv.org/abs/1804.02763

[29] Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall (2018). Activation Func-

tions: Comparison of Trends in Practice and Research for Deep Learning Arxiv Preprint Arxiv; arXiv:1811.03378v1

[30]] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for Activation Functions," ArXiv, 2017. [Online]. Available: 1710.05941;http: //arxiv.org/abs/1710.05941

[31] A. Maas, A. Hannun, and A. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," in International Conference on Machine Learning (icml), 2013.

[32] W. El-Shafai, N. El-Hag, A. Sedik, G. Elbanby, F. Abd El-Samie, an efcient medical image deep fusion model based on convolutional neural networks. Comput. Mater. Continua 74(2), 2905– 2925 (2023)

[33] Jadhav, S.B., Udupi, V.R. & Patil, S.B. Identification of plant diseases using convolutional neural networks. *Int. j. inf. tecnol.* **13**, 2461–2470 (2021). https://doi.org/10.1007/s41870-020-00437-5

[34] W. El-Shafai, M. Aly, A. Algarni, F. Abd El-Samie, N. Soliman, Secure and robust optical multi-stage medical image cryptosystem. CMC-Comput. Mater. Continua 70(1), 895–913 (2022)

[35] N. Soliman, N. Ali, M. Aly, A. Algarni, W. El-Shafai, F. Abd El Samie, An efcient breast cancer detection framework for medical diagnosis applications. CMC-Comput. Mater. Continua 70(1), 1315–1334 (2022)

[36] F. Alqahtani, M. Amoon, W. El-Shafai, A fractional fourier based medical image authentication approach. CMC-Comput. Mater. Continua 70(2), 3133–31504 (2022)

[37] W. El-Shafai, F. Khallaf, E. El-Rabaie, F. El-Samie, Robust medical image encryption based on DNA-chaos cryptosystem for secure telemedicine and healthcare applications. J. Ambient. Intell. Humaniz. Comput. 12(10), 9007–9035 (2021)

[38] W. El-Shafai, S. El-Nabi, S. El-Rabaie, A. Ali, N. Soliman, Efficient deep-learning-based autoencoder denoising approach for medical image diagnosis. CMC-Comput. Mater. Continua 70(3), 6107–6125 (2022)

[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," https://doi.org/10.48550/arXiv.1409.1556

[40] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016

[41] Korfiatis, P., Kline, T.L., Lachance, D.H. *et al.* Residual Deep Convolutional Neural Network Predicts MGMT Methylation Status. *J Digit Imaging* **30**, 622–628 (2017). https://doi.org/10.1007/s10278-017-0009-z

[42] Ramzan, F., Khan, M.U.G., Rehmat, A. *et al.* A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer's Disease Stages Using Resting-State fMRI and Residual Neural Networks. *J Med Syst* **44**, 37 (2020). https://doi.org/10.1007/s10916-019-1475-2

[43] He, K., Zhang, X., Ren, S., and Sun, J., Identity mappings in deep residual networks. Cham: Springer, 2016, 630–645.

[44] Polat, Ö., Güngen, C. Classification of brain tumors from MR images using deep transfer learning. *J Supercomput* **77**, 7236–7252 (2021). https://doi.org/10.1007/s11227-020-03572-9

[45] Kaiming H, Xiangyu Z, Shaoqing R, Jian S (2015) Deep residual learning for image recognition. arXiv:1512.03385

[46] He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 770-778 (2016)

[47] Buvaneswari, P.R., Gayathri, R. Deep Learning-Based Segmentation in Classification of Alzheimer's Disease. *Arab J SciEng* **46**, 5373–5383 (2021). https://doi.org/10.1007/s13369-020-05193-z

[48] Gordo, A., Almazán, J., Revaud, J. *et al.* End-to-End Learning of Deep Visual Representations for Image Retrieval. *Int J Comput Vis* **124**, 237–254 (2017). https://doi.org/10.1007/s11263-017-1016-8

[49] Khan, R.U., Zhang, X. & Kumar, R. Analysis of ResNet and GoogleNet models for malware detection. *J Comput Virol Hack Tech* **15**, 29–37 (2019). https://doi.org/10.1007/s11416-018-0324-z

[50] Le Yang, Xiaoyun Yu, Shaoping Zhang, Huibin Long, Huanhuan Zhang, Shuang Xu, Yuanjun Liao,GoogLeNet based on residual network and attention mechanism identificat ion of rice leaf diseases, Computers and Electronics in Agriculture, Volume 204, 2023, 107543, ISSN 0168-1699, https://doi.org/10.1016/j.compag.2022.107543.

[51] Çınar, A., Tuncer, S.A. Classification of lymphocytes, monocytes, eosinophils, and neutrophils on white blood cells using hybrid Alexnet-GoogleNet-SVM. *SN Appl. Sci.* **3**, 503 (2021). https://doi.org/10.1007/s42452-021-04485-9

[53] Yang, N., Zhang, Z., Yang, J. *et al.* A Convolutional Neural Network of GoogLeNet Applied in Mineral Prospectivity Prediction Based on Multi-source Geoinformation. *Nat ResourRes* **30**,3905–3923(2021) https://doi.org/10.1007/s11053-021-09934-1