A Survey: Plant Disease Detection Using Deep Learning

Anshul Tripathi[1], Uday Chourasia[1], Priyanka Dixit[1] and Victor Chang[2]

1. Rajiv Gandhi Proudyogiki Vishwavidyalaya (RGPV), BHOPAL, Madhya Pradesh 462033, India
2. Department of Operations and Information Management, Aston Business School, Aston University, Birmingham, UK.

Emails: anshultripathi731@gmail.com; udaychourasia61@gmail.com; dixitpriyanka162@gmail.com and ic.victor.chang@gmail.com

Abstract

Agriculture occupation is the prime occupation in India since the primeval era. Nowadays, our country has been ranking second in these prime occupations threatening global warming. Apart from this, diseases in plants are challenging to this prime source of livelihood. The present research can help in recognition of different diseases among plants and held out helping hand to find out the solution or remedy that we can call as a defense mechanism in counter to the diseases. Finding diseases among plant DL is considered to the most perfect and exact paradigms. Four labels are classified as "Bacterial Spot", "Yellow Leaf Curl Virus", "Late Blight" and "Healthy Leaf".

Keywords**:** Plant Disease, Deep Learning, Primeval.

1. Introduction

A large majority of the population in India depends on agricultural production. India ranks second in the agriculture output worldwide. In India, farmers cultivate a great diversity of crops. The food welfare and plants full of life and lively are interpersonally linked. The(FAO) displays that illness among plants and pests and diseases paves the way to become the cause of food production loss worldwide, approximately 20% - 40%, which is a big threat to food security. The use of pests is a way to protect crops to this law makes way for danger in the security of food, but applying them on crops leads to a negative impact on biodiversity and a to solve the requirement of expanding population. The situation is further complicated as nowadays of danger to humanity intense and persistent. Diseases can spread freely than in old times. The symptom of infected plants disease become visible superficially on leaves there become visible resulted in rotten leaves or rotten fruits. The disease was created due to pathogens such as Fungi, Bacteria, and Viruses. In Bhopal, the public observed that vegetables are being irrigated through sewage water flowing in a big drain in some areas. We can only imagine the impact of these vegetables appearing in our kitchens and resulting in health hazards. A report of FAQ says that there is a requirement to boost food production up to 65% to 70% by 2050 to fulfill the necessity of food in the world. 40% of food production is destroyed by infected plants' use of pests. In this category, Maharashtra ranks on top where farmers' suicide due to major reasons is the failure of crops.

There is a need for some method to detect diseases on the plant in the early stages. The identification and tabulation or classification of disease in the leaf is the key to prevent agriculture loss. For this purpose, Deep Learning is used. Deep Learning, in many cases, has been proved as a game-changer in solving complicated problems in a short span of time.

A variety of different methods and a lot of algorithms may be implemented for the mentioned purpose. For example, these include Linear Regression, Logistic Regression, K-Nearest Neighbours (KNN), Naive Bayes (NB), Gaussian Models, Decision Trees, +78+---Random Forest, Clustering and Support Vector Machines (SVM). In comparison to conventional methods, the above-said methods give precise forecasting helping in the flow of making a decision. Now in solving complicated problems, Deep Learning methods are being used and Capable to do pretty soon. Deep Learning is can also be useful for many other assignments as it is already a State-of-the-art methodology for the land cover classification task. [1]

1.1 Deep Learning In Image Processing

Deep Learning is a subset of Machine Learning, which is one of the Machine Learning techniques. Deep Learning instructs computers to act in such a way as human beings perform inherently. Deep learning may be a great potential for generating a successful model, framework, and method for achieving sustainable computing[44]. It is the main technology back automated cars, capable of determining a sign of stop or discrimination between road and pedestrians. In Deep Learning machine is being trained for assortment directly from images, sounds, or text. The DL models can acquire progressive validity. A NN contains several layers and categorized data and Neural Network architecture used to train a model.
A large amount of categorized data is needed in DL. It also essential extraordinary computing power. High Geared GPUs have a parallel architecture that is adequate for DL. Utmost DL methods used Neural Network architectures, so that is why Deep Learning is also termed as Deep Neural Networks. DNNs can be classified into three main types: convolutional neural networks (CNNs), deep belief networks (DBNs), and recurrent neural networks (RNNs) [45].
The name "deep" commonly cite for NN hidden layers. Conventional NN contains 2-3 hidden layers, but DL can contain as 150. The most popular Deep Neural Network is a Convolutional Neural Network (CNN).
There are commonly two ways to process the image, i.e., using grayscale and using RGB values.

i) Gray Scale: In this image is converted into white to black, which grayscale and a range of shades. Every pixel is attached to a value that confers to how dark it is.

ii) RGB Values: The colors are represented as RGB values, then each pixel extracted and set the result for the array.

1.2 Plant Disease

In the southern part of India, Agriculture has started as a livelihood before the Indus valley civilization. As a result, its position is second in the world in farming occupation. The 2018 data shows that out of the total population in India, 50% got employment in the agriculture sector. It has good participation by increasing the GDP of India from 17% to 18%. In India, a large number of people are farmers and depend on agriculture production. Plants are a necessity and requirement in human life because they are the gifted resource of the energy of nature and through there, we can beat the damage of global warming. Plant Disease in the agriculture sector, plants and diseases are the main threat.
The disease results in irregular shaped black patches that are spotted superficially over the leaf or over those fruits which have grown up earlier and under moisture in patches there is the possibility of appearing fungus, but speedily these patches cover the complete leaf or fruit area and the last we get rotten leaves and fruits. The disease in plants destroys crops of fruits and destroys the farmers whose livelihood totally depends on agriculture.
.Hence, it is necessary to identify plant disease in a precise and timely way. It is a well-known fact that for identification, tabulation of infected plants, a lot of ML models have been used, but increasing advanced technology DL has made possible the research area more valid and exact.

There are various diseases of plants and their sign and symptoms shown in Table 1.

TABLE 1: Different types of plant diseases

Disease	**Host**	**Signs and Symptoms**
Coffee Rust	Coffee	Symptoms appear on the lower side of leaves as Orange-Yellow powdery spots, while brown colors are shown in the centers.
Anthracnose	Various Category of Trees, grasses,	Infected leaves turn yellow, appear large round-black canker on leaves.

	vegetables, and shrubs	
Late blight of Potato	Potato	Tissue suffered damage through disease shows canker that is dark green to blackish purple, followed by a green edge on lower leaves.
Brown Rot	Stone Fruits	Brown spots appear on fruits, small buried brown blisters.
Chestnut Blight	Chestnut Tree	The symptoms appear on the skin of the plant in the form of Yellowish to little red and brown patches.
Loose Smut	Wheat, Barley, Oats	The upper part is infected shows olive-green spores in large numbers.
Powdery Mildew	Various Category of Trees, grasses, vegetables, and shrubs	Powdery Mildew grows in spotted form and develops in leaves and other organs of plants.
Black Stem Rust Of Wheat	Various grasses; Wheat	Shows bronze-colored boils added with a nut; after that, develop black rust fungi.

2. Literature Survey

1) In paper [8], the process of developing a plant disease determination model of DCNN on leaf image, systemized, has appeared with an advanced approach.
Training is an innovative way and procedure used to simplify and make easy a fast and simple system implementation in practice with the capacity to differentiate plant leaves from their surroundings. The advanced developed model can determine thirteen various kinds of plant illnesses from healthy leaves.
To set up a database assessed by agricultural experts, every critical step necessary for using this illness model is completely elaborated in this paper, beginning from gathering images. Berkley Vision and Learning Center has developed a deep learning framework 'Caffe' is being used to accomplish deep (CNN) training. For the Separate class test, the experimental result on the developed model is between 91% to 98%, with an average of 96.3%.

2) In paper [16] to perform detection of plant diseases using plain leaves pictures of the fresh and infected plant via DL, approach CNN models were developed with the by using an exact index/database of 87,848 pictures full of 25 flora or plants, training of the models was executed, that is in a group 58 different group of amalgamation in which fresh and healthy leaves were included. In order to recognize the concerned amalgamation, some prototype pattern was developed and top achievement reached the 99.53% success rate. The top achievement makes the prototype or model a practical and competent learning device. This access might be extended to assist a unified plant illness recognition system in conducting in familiar farming circumstances in the near future.

3) In Paper [12], we can conclude that CNN can revive important features categorize illness of plant pictures taken through the natural environment. In order to perform the soybean plant disease classification model, the model is designed on the basis of LeNet architecture, from the plant village database, where 12,673 samples were obtained containing pictures of the leaf of four groups inclusive of fresh pictures of the leaf. These images were taken in unrestrained surroundings. The model implemented reached 99.32% classification accuracy.

4) In paper [25], a deep learning meta-architectures detector is used to detect tomato plant diseases and pests. We acknowledge three main families of detectors. They are known as (Faster R-CNN)-Faster Region-Based Convolutional Neural Network. (R-FCN)- Region-Based CNN and (SSD)- Single Shot multi-box Detector, each of deep unreal planning architecture is associated with deep feature expression, for example, VGG Net and Residual Network (ResNet). To increase the reliability, validity, and precision and minimize the number of false or false positives during training. We authenticate the presentation of deep meta-architectures and feature extractors by coming up with suggestions for a procedure for local and worldwide class elucidation and data enhancement. On large tomato diseases and pest data sheets, they train and test their system throughout and it has challenging images with diseases and pests, including several inter-and-extra class variations as infection status and location in the plant.

5) In paper [2] To recognize 14 crop categories and 26 diseases, edify a deep convolutional neural network with an accuracy of 99.35%. The focused model achieves on a held-out test set showing the possibility of the approach. The

model carries out attain validity of 31.4% still when tested on fixed pictures gathered from trusted and reliable online sources, for example, drawn different from the pictures used for practice.
A more divergence set of exercise data is important to improve common and general precision, although the above-mentioned exactness is much higher than the one based on random selection (2.6%). In a nutshell, we can propound that exercise or practice DL model on growing worldwide reaching and publicly accessible images dataset come up with a crystal clear way towards smartphone-assisted crop disease investigation on a huge worldwide scale.

6) In paper [5], the aim is to identify plant disease is come up with a model with feature extraction, segmentation, and the classification of patterns of captured leaves. In this model, he has used four classifier labels - one is Bacterial Spot, the second is the Yellow Leaf Curl Virus, the third is Late Blight and the fourth is Fresh Leaf. The drawn-out components are fit into the neural network with 20 epochs. In recognizing the plant disease, few artificial neural network architectures are implemented with the best result of 98.59% validity and accuracy.

7) In paper [11], when dealing with real-life images, the fabricated type of compilation may badly influence the reliability and accuracy of the neural model. The problem with plant village image collection was noticed during research. A distinctive index of the small images of tiny pictures of grape leaves was gathered, comprising four sets of images to solve the problem of the small image database. A deep Siamese Convolutional Network was generated. Reliability and validity were 90% in identifying Black rot and Chlorosis diseases on the grape leaves by comparing findings of different prototypes or models and plants. Siamese Neural network is a viewpoint research field. For PDDP, we get to apply them as a basic deep learning architecture. We are ready to add more classes to the trained data early because their energy explores dissimilarities between classes. To develop the process of fulfilling DB with images of correctly identified disease, the quality of the database is fully necessary for the result.

8) In paper [13], it shows two Deep Learning-based architectures, namely AlexNet and VGG 16 Net, provided as input to study images of tomato leaves obtained from plant village datasets.
Significance of Hyperparameters as mini-batch size weight and bias learning are very few in the tabulation reliability. AlexNet and VGG 16 Net that is used to classify tomato crop disease. For VGG 16 Net, the classification validity was 97.29% and for AlexNet, it was 97.49% AlexNet immerge with fine precision with the least implementation time while deep VGG 16 Net is lagging behind.

9) In paper [17] To Identify and systemize the sigh of plant disease, a lot of developed DL planning is applied with a few measurement techniques. For the assessment of these techniques, some performance metrics are used. In order to conceptualize different plant diseases, this analysis gives a detailed explanation of DL models to reach greater transparency for identifying diseases in plants, while some research gaps are identified. The data set should point out the natural condition and pictures drawn in distinct field backgrounds and images taken in different field scenarios. For this purpose, DL planning must be systematic for many illustration Circumstances. In order to understand the factors affecting the identification of plant diseases, a deep study is necessary.

10) In paper [15] Banana, the most loved fruit known worldwide, is also detected with many diseases. Fast and innovative measures should be applied to control this. In crop detection, DCNN and transfer learning has been used and it has brought fruitful result in this field. It is necessary to develop an AI-Based Disease detection System to support banana farmers. Researchers show that for the banana disease detection system, DCNN was a strong strategy. To achieve a network that can make a correct prognosis, we applied DTL. For this, we used a pre-trained illness identification model. The success rate was very high. So in the future, we can establish a complete automatic mobile app to prevent banana disease.

11) In paper [18], the present paper uses Random Forest to detect amidst fresh and affected leaves from the generated datasets. This paper also contains a lot of execution entitled dataset creation, feature extraction, and classification. In order to classify the diseased and affected images, the generated datasets of affected and healthy leaves are trained all together under Random Forest. Histogram of an Oriented Gradient (HOG) is used to root out features of an image. Smoothly processing and managing the huge amount of datasets available gives a smooth and straightforward path to identify the disease in plants on an enormous scale.
The aim of this algorithm is to perceive irregularities appearing on plants in their natural environments. The algorithm was the opposite of other machine learning models for reliability and validity 160 pictures of papaya plant leaves were to train the model by applying the Random Forest Classifier. 70% correctness was estimated by the model

classification. This correctness may further be enhanced by training with many various pictures that used regional characteristics and global features such as SIFT, SURF and DENSE, and BOVW.

12) In Paper [1], a new dataset consisted of leaf images taken in different weather conditions during daytime in varying backgrounds where natural and practical situations were imitated conventional enhancement method and state-of-the-art style generative adversarial network were used to increase the number of images in the dataset. At the last final stage, plant disease classification was the center of attention on the natural environment and for this, an innovative neural network two-stage architecture was recommended. The correctness was 93.67% of the up-skilled model.
In the present paper, a new dataset originated in which pictures of leaves consisted of natural environments in dissimilar weather conditions with different angles. The little was classification and task detection, so the said dataset is thorough and global, improving the correctness of classification and the actual appropriateness of the model.

13) In paper [10], a survey of 19 studies was done, which was based on CNN to self-executing the recognition of diseases of the crop. In order to rate the problems of the network for approaches, main points, and limitations of work used CNNs to self-execute recognition, crop diseases have been identified. Few guidelines and lines of action were handled over and afforded to follow to escalate the probability of CNN's placed in the living world of flora and fau na.

14) In paper [3], a smartphone camera was used to collect the identified and systematized images of the maize leaf diseases by the CNN network. For this purpose, Neuroph was used to accomplish the training of CNN. The progressive model identified three dissimilar kinds of maize leaf diseases out of healthy leaves.
The northern leaf blight, common rust, and grey leaf spot diseases. These diseases had affected a large part of Southern Africa's maize field, and that is why they are chosen for the study. To setup, a more simplified Deep CNN Neuroph Studio Framework was used as an IDE, whereas the convolutional and pooling features extractions were implanted in the Neuroph library. Plant Village online website was used for datasets for trained, tested, and recommended CNN proving its probability. The accuracy was 92.85%. Suggestions were given to the researchers who were willing to use the recommended CNN in this study, that they use the resolution settings of 10 x 20 x 3 (height x width x RGB). Future research is propounded to stimulate achievement by CNN when the training and testing are effectuated black and white or monochrome images.

15) In paper [4], all the leaves are like texture and other similarities that describe or identify the type of disease. So computer vision and perception utilized with deep learning show the path to come out from this difficulty. By using a public dataset with images of healthy and defective crop leaves, the present paper propound a trained deep learning model by tabulating images of leaves into an affected category, which depends on the defect of the template. The model meets its goal and aim. The present paper reveals that the approach and application of Deep Convolutional Neural Network have been put together to classify both crop species and identify disease on images. Three types of crop disease and five classes of crops were taken on the test by the suggested methodology. In terms of correctness and failure of validity and reliability, the experimental result display that the Inception V3 model achieves better than the Mobile Net model.

We probe the multi-user resource allocation problem in the MEC system to maximize the long-term performance of the whole system. As a solution, we make computing and communication resource location architecture for different users. After that, we formulate an infinite discounted continuous state in DP. At last, the front-end and back-end queues are used for each mobile user. See Table 2.

TABLE 2: The comparison of various paper's models and accuracy.

Sr. No	**Paper Reference**	**Dataset**	**Model & Technique Used**	**Result In Accuracy**
1.	[1]	Plant Village	PlantDiseaseNet- (PDnet-1 & PDnet-2) A Novel Two Stage Architecture	93.67%
2.	[12]	Plant Village	LeNet Architecture Based CNN Model	99.32%
3	[8]	Open Database	CNN Deep Training using Berkley Vision and Learning Centre	96.3%

4	[10]	Plant Village	Transfer Learning On CNN by Inception-V3, Squeeze Net, VGG-16, ResNet-101, ResNet-50, DenseNet-121	N/A
5	[9]	Plant Village & Own Dataset	CNN	78%
6	[2]	Plant Village	Transfer Learning from Scratch with Alex Net, Google Net in Deep CNN	99.35%
7	[13]	Plant Village	Transfer Learning with Alex Net and VGG-16	AlexNet:- 97.29% VGG-16:- 97.49%
8	[3]	Own Dataset	CNN using Neuroph Framework	92.85%
9	[5]	Plant Village	Artificial Neural Network Architecture	98.59%
10	[4]	Plant Villa	Transfer Learning Using CNN with Mobile Net and Inception-V3	Mobile Net:- 99.62% Inception-V3 :- 99.74%
11	[16]	Open Database	CNN Models Developed With Deep Learning Methodologies	99.53%
12	[25]	Own Dataset	DL meta-architecture using Faster R-CNN, R-FCN, SSD	85.98%
13	[18]	Own Dataset	Machine Learning Using Random Forest Classifier	~70%
14	[21]	Own Dataset	Machine Learning Using SVM, KNN	KNN:-98.56% SVM:- 97.6%
15	[20]	Plant Village and Malaya New Dataset	CNN	87%
16	[23]	Plant Village	CNN, Tensor Flow, Android Application, Java Web Services	89.67%
17	[22]	Own Dataset	API applied for CNN, Python	96.5%
18	[24]	Apple Leaf Disease Dataset (ALDD)	INAR-SSD Using DeepCNN	78.80

3. Methodology

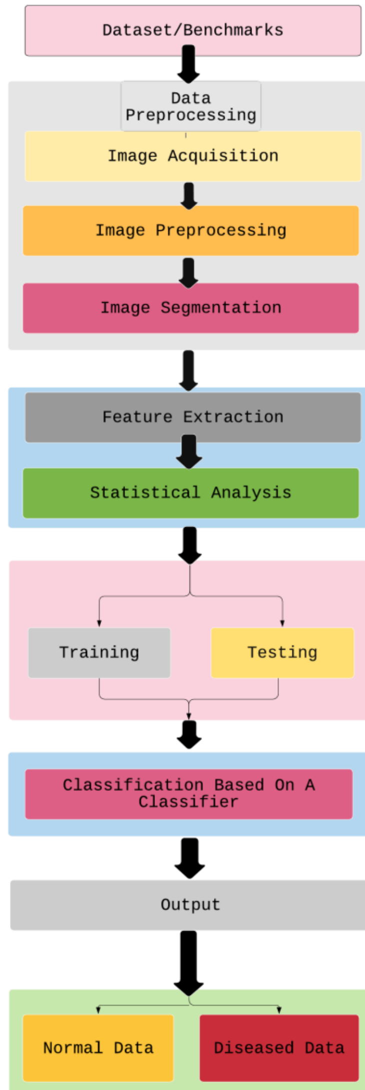

Figure 1. Methodology

The Plant Disease Detection Methodology as shown in Figure 1 Steps Are:-
1). Dataset
2). Image Acquisition
3). Image Preprocessing
4). Image Segmentation
5). Feature Extraction
6). Statistical Analysis
7). Classification Based On A Classifier

1). Dataset – The DataSet class represents a data transform with input and output data. In the context of neural networks, a dataset represents a mapping of input features to an output vector (e.g., outcomes). The outcomes are specifically for neural network encoding such that any labels that are considered true are 1s. The
rest are zero.

2). Image Acquisition - It is one of the hidden ideas of DL through which the pieces of Images are named, particularly for preparing the models.

3). Image Preprocessing – In this, the RGB images collected are converted into HSI color structure in the Preprocessing step. The aim is to improve the image data that suppresses unwanted distortions or enhance some image features important for further processing. There are four categories of image preprocessing –

1). Pixel Brightness Transformations
2). Geometric Transformations
3). Local Neighborhood Processed Pixel
4). Image Restoration

4). Image Segmentation – In this, a Visual input is being partitioned into segments for image analysis. Segments are the part of the object and compose a set of pixels, as showing in Figure 2.

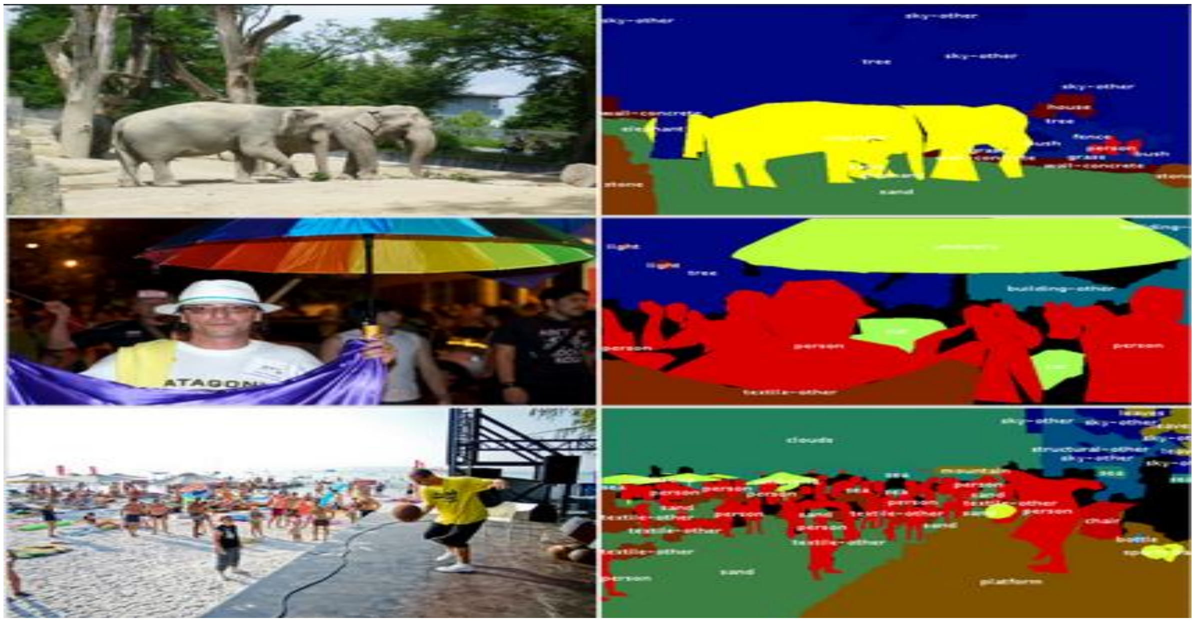

Figure 2. RGB Image Converted to Segmented Image

5). Feature Extraction - Feature extraction is the process of mapping a given feature set from a higher dimensional space to a lower-dimensional space. The process of feature extraction is performed to reduce the complexity and increase the efficiency of the classification process. Feature extraction makes use of a projection matrix that contains values that are mapped to the lower-dimensional space using a function.

6). Statistical Analysis – In this, the extracted features are being taken in and then processes to identify the target result.

7). Classification Based on the classifier- The DL Network will obtain the particular image to be targeted and classify the output.

4. Deep Learning Algorithms For Plant Disease Detection

1) Convolutional Neural Network (CNN)
2) Radial Basis Function (RBF)
3) Single Layer Perceptron (SLP)
4) Multilayer Perceptron (MLP)

4.1) Convolution Neural Network

The Technique is used in Convolutional Neural Network (CNN), the subpart of Deep Learning (DL).
Convolutional Neural Network is just like a Neural Network designed for neurons with trainable weights and biases. Each and every one neuron intake several inputs, take a weighted aggregate of them, and pass it over an activation function and counter with an output.

CNN is the Regularized version of Multi-Layer Perceptron (MLP). The goal of CNN is to learn high-dimensional features in the data through convolutions. These networks are also used in object recognition, image classification, text recognition, etc. They can determine most of the objects with effectiveness and clarity like faces, individual person, street signs, and many other image data. CNN can very effectively structure the input data (image, audio, or video).

CNN also has been used in other tasks such as natural language translation and sentiment analysis. Convolution is a powerful concept for helping to build a more robust feature space based on a signal.

Biological Inspiration Of CNN

The visual protective layer (cortex) in animals is the biological inspiration for CNN. The visual field in which the cells are in the visual protective layer is sensitive to a small sub-located input area. These small areas are connected jointly to seize the whole visual field. The cells are effective for utilizing the powerful vastly local association establishes in the types of resemblances and our nervous system procedure and functions as local filters over the intake range and sphere widely; there are two main categories of cells in this area of the brain. When simple cells discovered an edge-like design/pattern, they activate and turn on. When these simple cells have a larger visual field and are undeviating to the position of the pattern, they are more complex.

The CNN Architecture

CNN transforms all sets of the input data that passes through different connections (all intermediate layers) into a collection of class scores provided by the output layer. There are many different CNN architectures, but based on the pattern of layer or model are as shown in Figure 3.

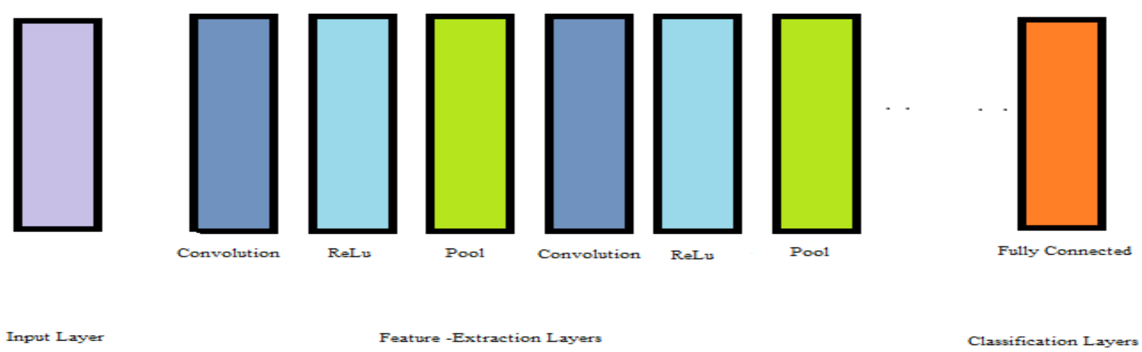

Figure 3. CNN Architecture

Three major components –

1) **Input Layer**
2) **Feature Selection Layers**
3) **Fully Connected Layers**

The neuron layer is connected to the set of neurons that act as small processing units of a limited small area, just like a traditional multilayer network. CNN also holds on to a layer-oriented architecture but has a distinct one. Every layer changes the 3D input volume from the preceding layer to output volume in 3D of activation neuron along with few differentiable functions that may or may not contain parameters—shown in Figure 4.

4.1.1). Input Layer - Normally, the input layer receives three-dimensional input in the form vastly of size (width * height) of the image and has a bottom illustrating the color channel.
We call input layers those in which raw figures data of the image is loaded and stored for the advance procedure. The width, height, and number of channels are mostly defined by the Input data. The number of channels for the RGB values for each pixel is three.
.

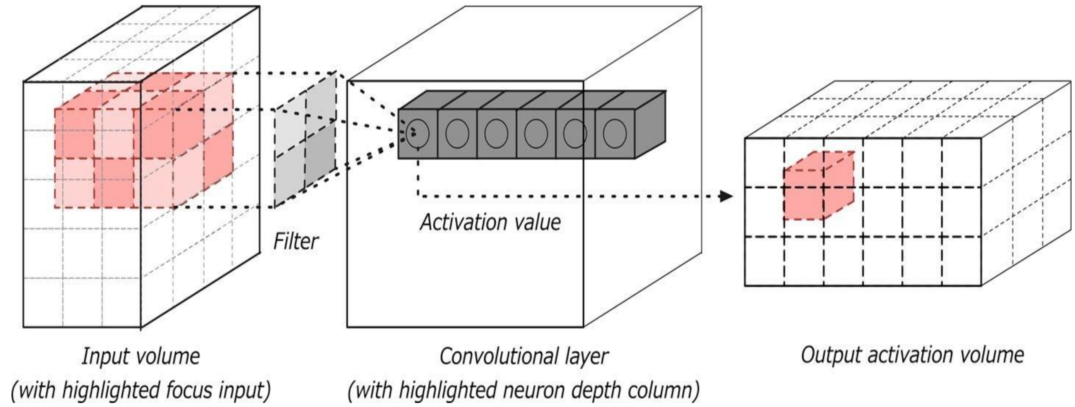

Figure 4. Input and Output volumes In Convolution Layer. [27]

4.1.2). Feature Extraction - The feature-extraction layers have a generally repeating pattern of the sequence:

1. Convolution layer-- Convolution layers are the main building blocks of CNN architectures. The ReLU activation work as a layer in the present diagram to go up to other literature has been elaborated here. By using a mark of locally connecting neurons from the preceding layer, mostly the CNN layer survey set inputs to change the input data. A dot product will be calculated by the layer between the region of neurons in the input layer to the weights of the output layer connected as locally. The result is the same smaller spatial dimension, but the number of components is enlarged at some times in the third dimension. So it is necessary to have an extensive review of the main concept in these layers, which are called Convolution. We can define convolution as a different model with a mathematical operation that elaborates a protocol to combine on pair of information. It is significant that mathematics and physics act as a bridge between frequency and time/space FT transformation. It accepts a set of inputs, applies a convolution kernel, and mapping features as output. The set of operations in it are called a feature detector.
The input to a convolution can be defined as raw data or a feature map output from another convolution. Most of the time, it is elaborated and expressed as a filler in which the kernel filters input data for determining kinds of information. Figure 5 clarifies, the function of the kernel is to glide across the input data and generate output data the convoluted feature. The kernel is multiplied step by step by the input data within its hurdles or bounds. It set up a single entry in the output feature map. If we are seeking the feature that is disclosed in the input, the output is wide-ranging. Often, we mention the set of weights in a convolutional layer as a filter that is the kernel. We get the outcome as a feature map when the kernel is convoluted with the output.

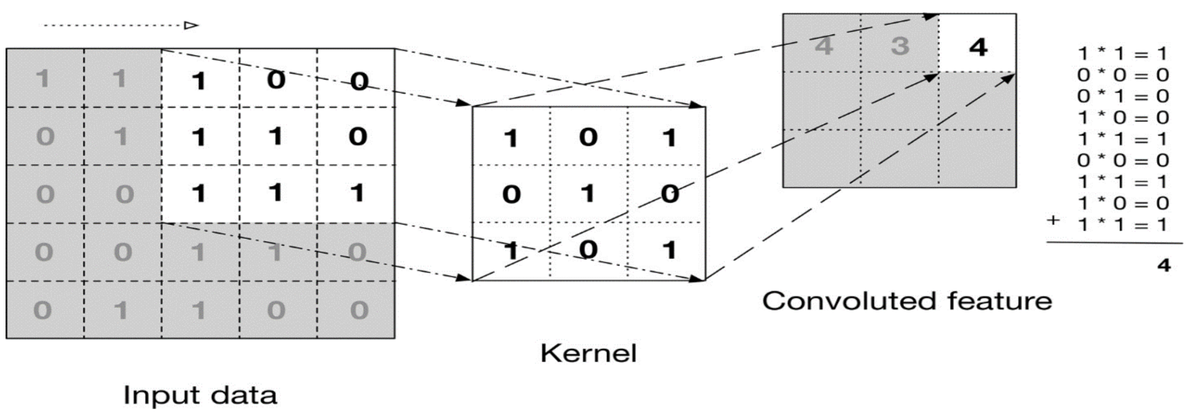

Figure 5. Convolution Operation [27]

The alteration and modification of the input data volume are presented by the convolution layer, which is the role of activations in the input volume and the framework parameters (weight and biases of the neurons). To generate the 3D output volume, the activation map for each kernel is pile up together with the depth dimension.

There are frameworks and parameters for the layer and additional hyperparameters convolutional layers. To direct and aim, the framework parameter in this layer gradient descent is applied in the same way as scores of the class are compatible with the labels in the training set.

The main units and parts of the convolutional layer are:-

I). Filters
II). Activation Maps
III). Parameter Sharing
IV). Layer – Specific Hyper-Parameters

I). Filters - The arrangement and designing of the layer's set of filters is presented by the parameters for a convolutional layer. We can define filters as functions with a width and height smaller than the width and height of the input volume. As we slide the window pane, filters are used across the width and height of the input volume in the same manner. We also use filters for every depth of the input volume. By producing the dot product of the filter and the input region, the output of the filter is calculated and determined.
The learned filters generate the strongest activation to extensive local input design specimens by planning, construction, and architecture. When the design specimen or pattern takes place in the training data in their field, only then learned filters activate on pattern or features. We come across filters that can accept non-linear ombinations of features and are growing worldwide in how they can discover patterns as we start ahead along in layers in a CNN Network depth to be an important component in CNN; it is clearly displayed by high performing convolutional architecture.

II). Activation Maps - If a neuron is determined to allow information to pass through, then activation is a numerical result. The activation function, the weights on the connection, is the function of the inputs to the activation function. Which filter permits information to move through it from the input volumes into the output volumes, then the filter activates. We skate or glide the filters from one side to another during the forward movement of data or facts through CNN. The spatial dimensions are the width and height of the input volume. This procedure generates a two-dimensional output and is named as an activation map for particular filters.

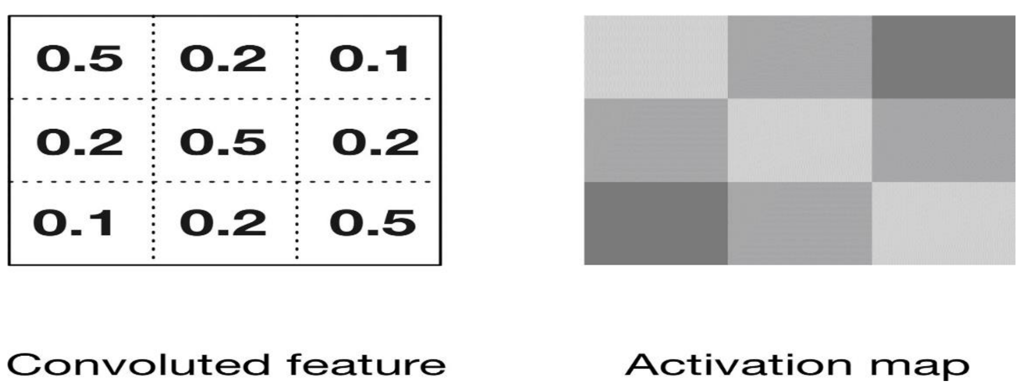

Figure 6. Convolutions and Activation Maps [27]

How convolutional activation maps are regularly furnished and provided in the literature, in figure 6 activation map, is furnished differently to demonstrate and exemplify. We put the filter from one side to another side of the input volume depth to calculate the activation map. Between the entries in the filter and the input volume, the dot product is calculated. The weights are multiplied by the moving window of activation and a filter represents these. In a particular fixed spatial position, the network learns filters turn on when they see recognized features in the input data. The three-dimensional output volume is generated for the convolution layer by cumulating these activation maps along the depth dimension in the output. The output of a neuron looks at only a tiny window of the input volume. The output volume has entries.

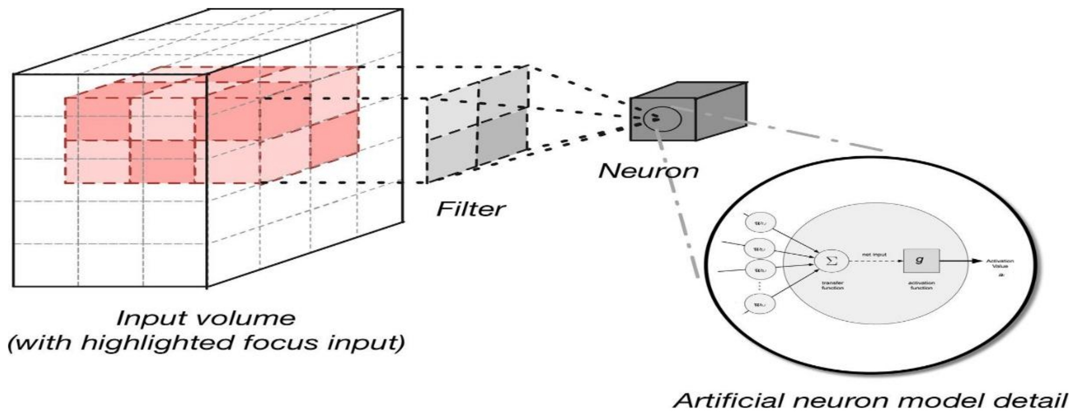

Figure 7. An Activation Output Volume Generation [27]

In some activation maps, this output will be the result of a framework or parameters shared with neurons from time to time. A local region of the input volume to which each neuron is attached, creating the output volume. The receptive field that is called by the name of hyper-parameter this process is managed by local connectivity. The receptive field manages the quantity of width and height of the input volume compared to the filter map. To create an activation map from the input volumes, Filters specify and interpret a tinier bounded region. Previously, it was illustrated that filters are joined to a subset of the input volume. They are connected through the dynamics of local connectivity. When we minimize the number of parameters or framework per layer to train the procedure mentioned above permits quality feature separation or extraction at an advanced stage, the Parameter sharing technique is used to minimize parameter count by convolutional layers.

III). Parameter Sharing - To manage the total parameter count, a parameter framework sharing plan of action is adopted by CNN. The plan mentioned above helps in training time to use a few means to learn the training dataset. In the beginning, we represent and identify a single two-dimensional slice of depth as a 'depth slice' to execute framework or parameter sharing on CNN. After this, we force the neurons in each depth slice to use the same weight and bias. So, for a given convolutional layer, this approach provides us remarkably fewer parameters. Unless the input image has a particular centered structure, we can't take the lead of parameter sharing when we suppose a particular feature to look in a particular place. The impact can be seen in faces. In all likelihood, parameter sharing is not used. Whatever CNN provides unchangeably and even to translation/position is called parameter sharing.

ReLU activation functions as layers-

Frequently we observe that the ReLU layer is used with CNN. A component-wise activation function over the input data thresholding will be applied by the ReLU layer. The input volume will change pixel values if we flow this function over it but the spatial dimensions of the input data in the output will not change. ReLU layers don't have parameters and additional hyper-parameters.

.

IV). Convolutional Layers Hyper-parameters - Following are the hyper-parameters that dictate the spatial arrangement and size of the output volume from a convolutional layer are:

a) Filter Size
b) Output Depth
c) Stride

d) Zero-Padding

a) Filter Size - If we compare the size of the filter with respect to the width and height, every filter small spatially. The three-sized filter 5x5x5 is the first convolutional layer. We can take up input image in three-channel RGB color, So we can explain that the filter represents three color channels: 5 pixels wide and by five pixels tall.

b) Output Depth- To collect the depth of the output volume, it requires human and physical effort. In the convolutional layer, the neuron count joined to the same region of the input volume is directed and managed by hyper-parameter depth. We study and examine a set of neurons as a depth column that looks at the same region of the input volume.

c) Stride - The answer to the question that per application of the filter function, how far our sliding or gliding filter window will move when we produce and design a new depth column in the output volume we put in the filter function to the input column. In the output volume, a lower setting for stride will allot and design more depth columns. This setting will lead to larger output volumes by giving more heavily overlapping receptive fields between the columns. When we define higher stride values, the contrast is true. We get and receive less overlap and smaller output volumes in computative graphics by these higher stride values.

d) Zero Padding - We can command the computative graphics as the spatial size of the output volumes by the zero paddings, which is the last hyper-parameter. When we want to keep up the spatial size of the input volume in the output volume, apply it in such cases.

2. Pooling layer-- Pooling layers are the layers that discover and notice a lot of properties and characteristics in the images. These increasingly set up higher-order properties and characteristics are spontaneously learned as a contrast to normally hand-engineered. This communicates straight to the ongoing theme in deep learning. We have more than one completely joined layers to take higher-order properties and generate class probabilities or scores. As the title of these layers tells us, these layers are completely joined and connected to all neurons in the prior layer. The result of these layers generates a two-dimensional output of the dimensions [b x N] here, b denotes the number of examples in the mini-batch and N implies the number of classes in which we take an interest in scoring. Normally, pooling layers are fitted between successive and consecutive convolutional layers. We follow convolutional layers with pooling layers to increasingly minimize the spatial size (width and height) of the data representation. These pooling layers minimize the data representation increasingly over the network aid in controlling over-fitting. On the entry depth slice of the input, these pooling layers work autonomously. To resize the input data spatially (width, height), the max operation process is used by the pooling. This procedure is mentioned as max pooling. The max operation process is taking the largest of four numbers in the filter area with a 2 x 2 filter size. The depth dimension is not influenced by this operation. To present the downsampling process on the input volume filters are used by pooling layers, with the computative graphics or spatial dimension of the input data, these pooling layers represent executing a downsampling operation. This means that if the input image were 32 pixels wide by 32 pixels tall, the output image would be smaller in width and height (e.g., 16 pixels wide by 16 pixels tall). To apply 2 x 2 filters with a stride of 2 is the most common setup for a pooling layer. This will downsample each depth slice in the input volume by a factor of two on the spatial dimensions (width and height). When 75% of activation is given up, the downsampling operation will be output in it. Pooling layers have extra and additional hyper-parameters in spite of parameters for the layer. This layer calculates a steady function of the input volume, so it does not involve parameters. To use zero paddings for pooling layers is not common.

3). Fully Connected Layers- Fully connected layers are used to calculate class scores used as the output of the network. The output volume dimensions are [$1 \times 1 \times N$], where N is the number of output classes we're evaluating. *N* would be 10 for the ten classes of objects in the dataset. Every neuron is connected with the prior layer and this layer is connected to all of its neurons. For the layer and hyper-parameters, fully connected layers have normal parameters. The input data volumes are a function of the activation in the input volume and the parameters, Fully connected layers present transformation on this input data.

Additional Applications of CNN

The position-invariant nature of CNNs has proven useful in these domains because we're not limited to hand-coding our features to appear in certain "spots" in the feature vector. Three-dimensional (3D) scene understanding is of great

significance to several robotic applications. With the large development of the deep learning methods, especially the convolutional neural network (CNN) [43].
Beyond normal two-dimensional image data, we also see CNNs applied to three-dimensional datasets. Here are some examples of these alternative uses:

- Graph Data
- MRI Data
- NLP Applications
- 3D Shape Data

Some Popular Architectures of CNN

1) LeNet

- Originally Used to Read Digits In Images
- One of the initial lucrative architectures of CNN
- Developer - Yann Lecun
- Shown in Figure 8.

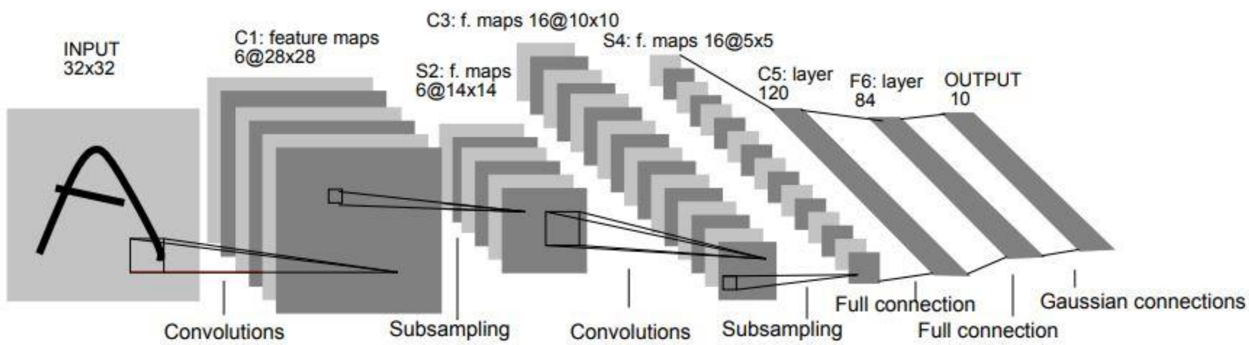

Figure 8. The LeNet Architecture [29]

2) AlexNet

- Helped popularize CNNs in computer perception
- Won the ILSVRC 2012
- Developer- Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton
- Shown in Figure 9.

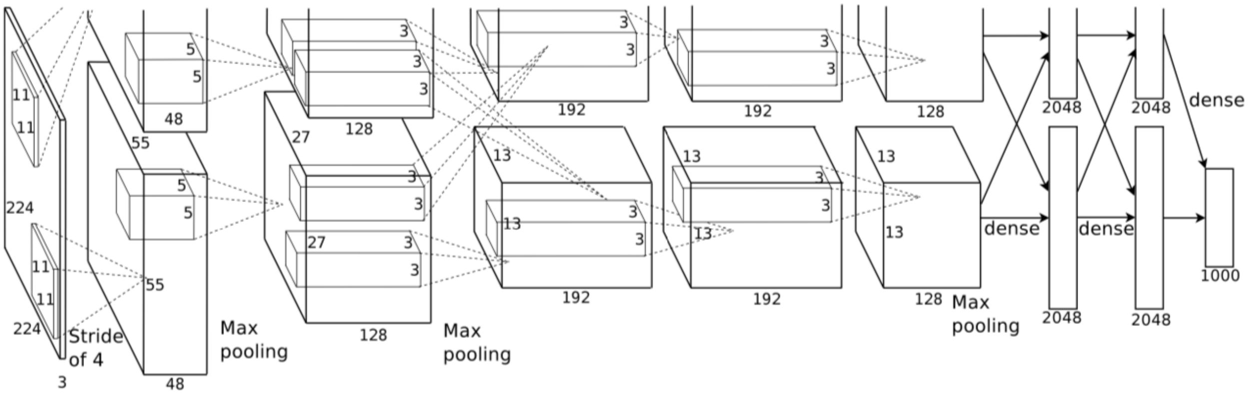

Figure 9. The Alexnet Architecture [30]

3) ZF Net

- Proposed the visualization approach of the De-convolutional Network
- Triumphed the ILSVRC 2013
- Developer- Matthew Zeiler and Rob Fergus
- Shown in Figure 10.

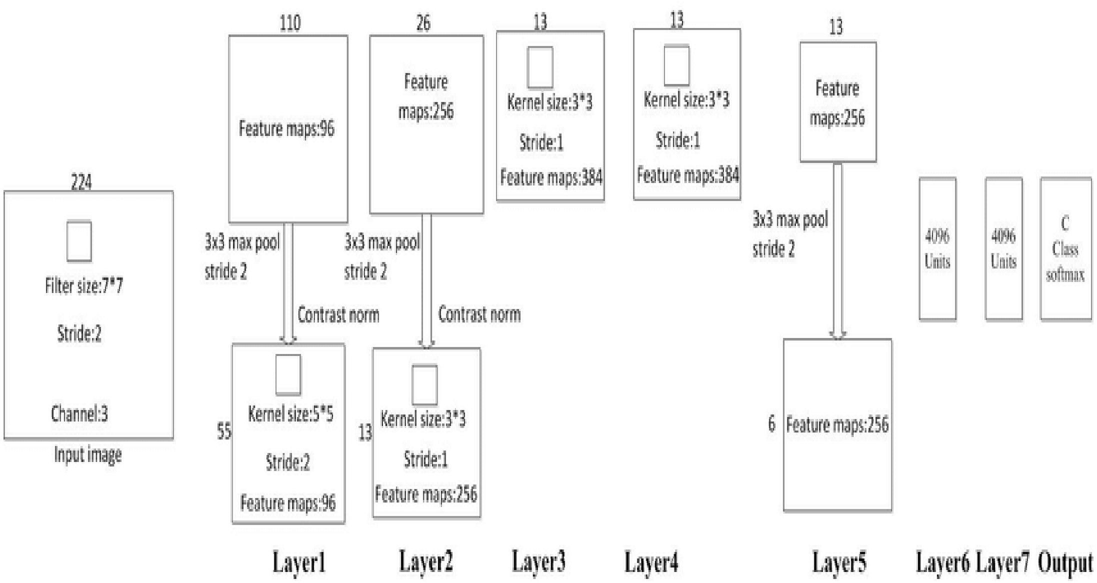

Figure 10. The ZF Net Architecture

4) GoogLeNet

- Codenamed "Inception," one variation has 22 layers.
- Triumphed the ILSVRC 2014
- Developer - Christian Szegedy and his team at Google
- Shown in Figure 11.

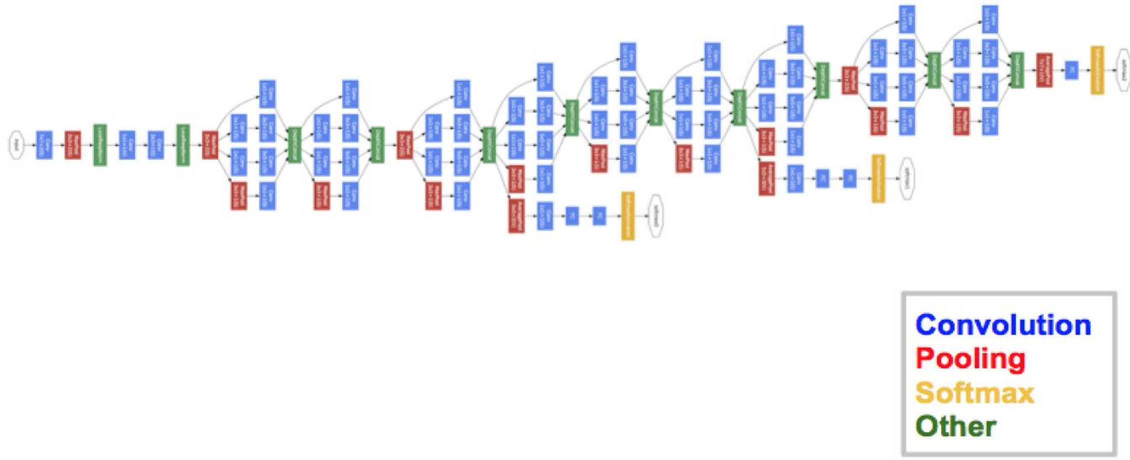

Figure 11. The GoogLeNet Architecture [31]

5) VGG Net

- Showed that depth of network was a critical factor in a good performance
- Second in the ILSVRC 2014
- Developer- Karen Simonyan and Andrew Zisserman
- Shown in Figure 12.

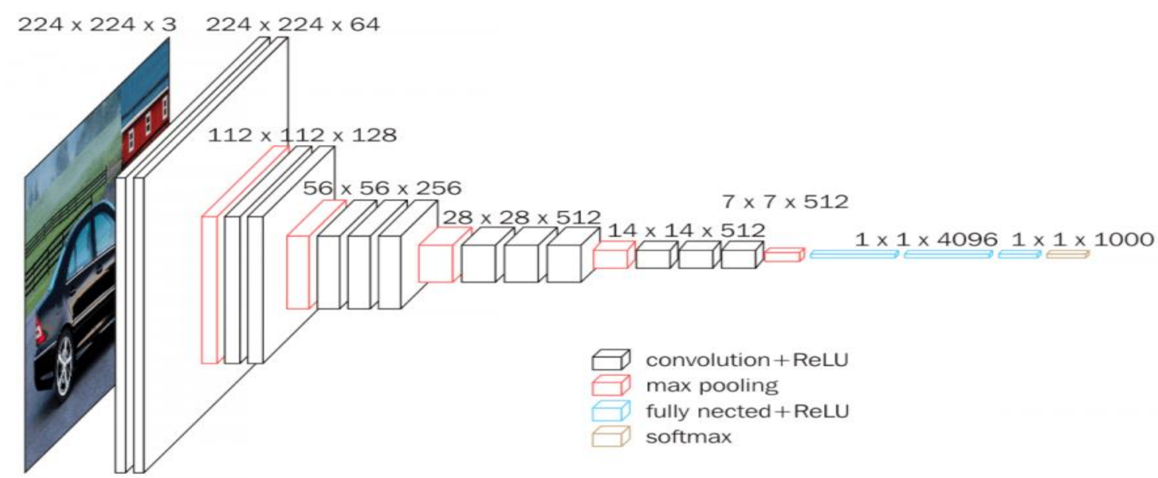

Figure 12. The VGG Net Architecture [32]

6) ResNet

- Equipped with very deep networks
- Triumphed in the ILSVRC 2015
- Shown in Figure 13.

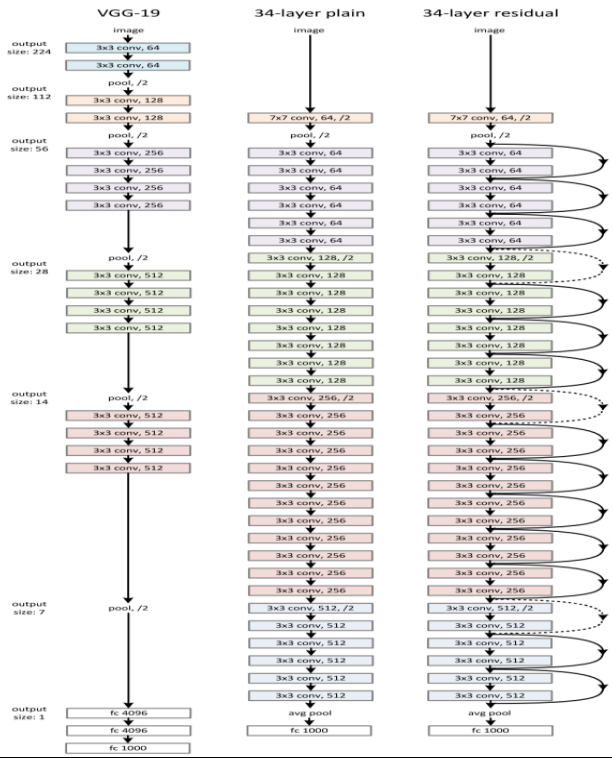

Figure 13. The ResNet Architecture [33]

TABLE 4 Comparison of CNN Architectures

CNN Architecture	Year	Salient Feature	Top -5 Accuracy	Number of Parameters
LeNet	1998	Simplest Architecture	N/A	60 Thousand
AlexNet	2012	Deeper	84.70%	62 Million
ZF Net	2013	Improved Image Classification Rate	85.20%	N/A
GoogLeNet	2014	Wider -Parallel Kernels	93.30%	6.4 Million
VGG Net	2014	Fixed – Size Kernels	92.30%	138 Million
ResNet	2015	Shortcut Connections	95.51%	60.3 Million

4.2) Radial Basis Function (RBF)

Radial basis function networks (RBF networks) are a paradigm of neural networks, which was developed considerably later than that of perceptrons. Like perceptrons, the RBF networks are built in layers. But in this case, they have exactly three layers, i.e., only one single layer of hidden neurons, as shown in Figure 14. Like perceptrons, the networks have a feed-forward structure and their layers are completely linked. Here, the input layer again does not participate in information processing. The RBF networks are -like MLPs - universal function approximators. Although all things in common, the difference lies in the information processing itself and in the computational rules within the neurons outside of the input layer. So, in a moment, we will define a so far unknown type of neurons.
In Radial Basis Functions, we have three layers :
i) Input Layer

ii)Hidden Layer
iii) Output Layer

i) Input layer: It subsists of m_o nodes of source, where m_o is the ambit of input vector x.

ii) Hidden Layer: It subsists of the equivalent count of computation units as compared to the training fragment, which is *N*. Every entity analytically characterized by a radial basis function

$\varphi(x) = \varphi(\| x - xj \|), \quad j = 1,2,\ldots, N$ [1]

In equation 1,The *xj* which is the *jth* point of data intake, construes the center of the radial-basis function and in the input layer in the form of signal vector x is enforced. So, dissimilarly a multilayer perceptron, links interlacing the source nodes to hidden entities are continuous affiliations with *no* weights.[28]

iii) Output Layer: In this, the architecture of the RBF individual computational entity. There is no stipulation on the size of the output layer. Apart from typically, the output layer size is considerably shorter than the hidden layer. Hereafter, In equation 2 we concentrate on applying of Gaussian Function in the character of radial basis function, in which every computational entity in the hidden layer of the network [28].

$\varphi i(x) = \varphi(x - xj)$ [2]

$= exp(- 1/2\sigma j\text{^}2 \; \| x- xj\| \text{^}2), \quad j = 1, 2, ..., N$ [3]

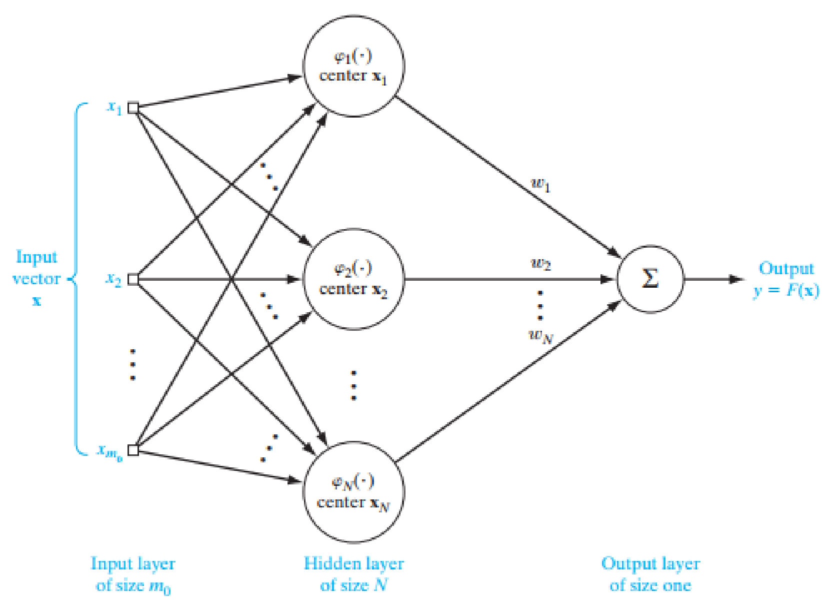

Figure 14. RBF Structure [28]

where σj the width, which is evaluated of the *j*th Gaussian function with center *xj*,In Equation 2. Consistently, but not repeatedly, σ is the assigned common width by Gaussian hidden entities. Now *xj* is the center of one hidden entity from other of the figure-out parameter. The justification responsible for selecting Gaussian function in the character of the radial basis function in the RBF network is fascinating properties.[28]

4.3) Single Layer Perceptron (SLP)

The perceptron is a linear model used for binary classification. In the field of neural networks, the perceptron is considered an artificial neuron using the Heaviside step function for the activation function. The perceptron training algorithm is considered a supervised learning algorithm. The binary classifier with a consecutive model shows summing *n* number of inputs times their associated weights and then sending this "net input" to a step function with a defined threshold. Typically with perceptrons, this is a Heaviside step function with a threshold value of 0.5. This function will output a real-valued single binary value (0 or a 1), As shown in Figure 15, depending on the input.[27]

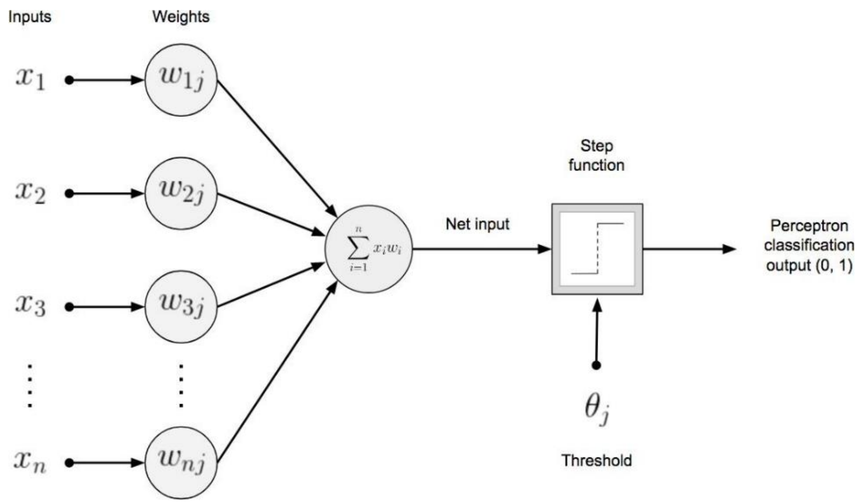

Figure 15. Single Layer Perceptron [27]

Model the decision boundary and the classification output in the Heaviside step function equation, as shown in Equation 4:

f(x) = {0 x<0 ; 1 x > = 0} [4]

To produce the net input to the activation function (here, the Heaviside step function), we take the dot product of the input and the connection weights.
A basic Linear Classifier, we consider Single Layer Perceptron to be the elementary form of feed-forward neural networks.[27]

4.3.a). The Perceptron Learning Algorithm

The perceptron learning algorithm changes the perceptron model weights until all input records are correctly classified. If learning input is non-linear separable, then termination of the algorithm will not be complete. A linearly separable dataset is one for which we can find the values of a hyperplane that will cleanly divide the two classes of the dataset. The perceptron learning algorithm initializes the weight vector with small random values or 0.0s at the beginning of training. The perceptron learning algorithm takes each input record and computes the output classification to check against the actual classification label. In order to produce the classification, the columns (features) are matched up to weights where *n* is the number of dimensions in both our input and weights. The first input value is the bias input, which is always 1.0 because we don't affect the bias input. The first weight is our bias term in this diagram. The dot product of the input vector and the weight vector give us the input to our activation function. If the classification is correct, no weight changes are made. If the classification is incorrect, the weights are adjusted accordingly. Weights are updated between individual training examples in an "online learning" fashion. This loop continues until all of the input examples are correctly classified. If the dataset is non-linear separable, then the training algorithm will not terminate.

4.3.b). Limitations of Single Layer Perceptron

The perceptron was found to be limited in the types of patterns it could recognize. The initial inability to solve non-linear (e.g., datasets that are not linearly separable) problems were seen as a failure for the field of neural networks. So this is the drawback of a single-layer perceptron. However, the general industry did not widely realize that a multilayer perceptron could indeed solve the XOR problem, among many other non-linear problems.

4.4) Multi-Layer Perceptron (MLP)

A perceptron is a neural network with a single layer of input linear neurons, shown in Figure 16, followed by an output unit based on the *sign(•)* function (alternatively, it's possible to consider a bipolar unit whose output is -1 and 1). The main limitation of a perceptron is its linearity. How is it possible to exploit this kind of architecture by removing such a constraint? The solution is easier than any speculation. Adding at least a non-linear layer between input and output leads to a highly non-linear combination, parametrized with a larger number of variables. The resulting architecture is called Multilayer Perceptron (MLP) and containing a single (only for simplicity) Hidden Layer. This is a so-called feed-forward network, meaning that the flow of information begins in the first layer, always proceeds in the same direction, and ends at the output layer. Architectures that allow partial feedback (for example, in order to implement a local memory) are called recurrent networks.[26]

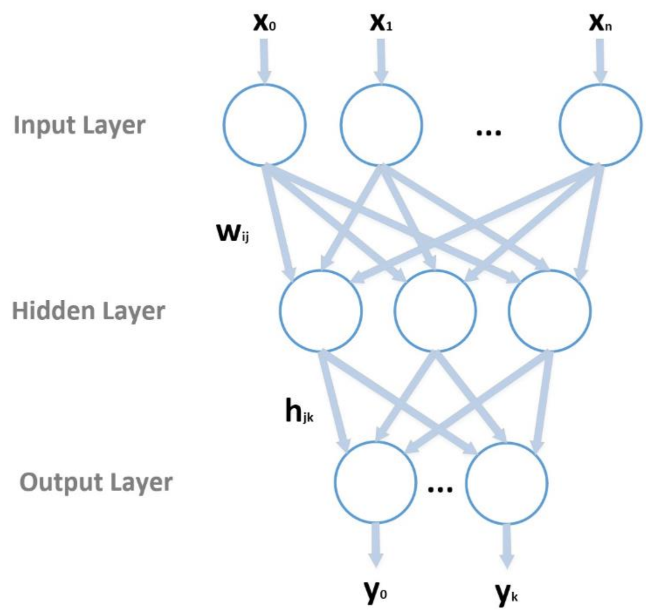

Figure 16. The Multilayer Perceptron Architecture [26]

There are two weight matrices, *W* and *H*, and two corresponding bias vectors, *b*, and *c*.

5. Conclusion

The present paper has access to identifying plant illness and elaborates and explains the said issue. Adding to this a lot of anticipation, skill and approaches have been sum up to identify the system of illness or diseases. The present paper also identified some crucial and concern points and limitations of work used by CNN to detect naturally crop diseases to pursue to escalate the potentials of CNN placed in natural world utilization and operation and the present paper provided ground rules, instructions, and approaches. Naturally, the tabulation and detection of plant diseases from pictures of the leaf by a novel way of using the DL technique were analyzed through this paper. The detection of the presence of leaf and discrimination between fresh leaves and 13 different diseases can be identified visibly with the help of a said model. In few yesteryears, the achievement and attainment of CNN in object perception and picture and image tabulation has made remarkable pace, growth, and progression. In the early years, the conventional way for picture tabulation was based on hand-designed features such as SIFT HOG SUPF. In this work, DL based detector for tomato plant disease and pest have been proposed. This paper also reveals real-time detection based on improved CNN for apple leaf diseases. The DL-based approach can automatically extract the discriminative features of the diseased apple images and detect the five common types of apple leaf diseases with high accuracy in real-time. This present study, CNN is used to identify and detect and tabulate soybean plant disease. The network is trained using the images taken in the natural environment and achieved 99/32% classification ability. Our aim was to find the more suitable DL Architecture for our task. The experimental results and comparison between various deep-meta-architectures with feature extractors displayed how our deep-learning-based detector can successfully recognize and identify nine different categories of diseases and pests, including complex intra-and interclass variations. We expect and look forward that our proposed system will significantly contribute to the agriculture research area. Future work will focus on improving the current results and a promising application will extend the idea of diseases and pest recognition to other crops.

Nomenclatures

m_o	Ambit
N	Training Fragment
φ	Gaussian Function
x	Input Vector
xj	jth point of data intake
σj	Width
σ	Common Width
W,H	Weight Matrices
b,c	Bias Vectors

Abbreviations

CNN	Convolutional Neural Network
DL	Deep Learning
RGB	Red Green Blue
ReLU	Rectified Linear Unit
2D	Two Dimension
3D	Three dimension
NN	Neural Network

References

1. Arsenovic, M.; Karanovic, M.; Sladojevic, S.; Anderla, A.; Stefanovic, D. Solving Current Limitations of Deep Learning Based Approaches for Plant Disease Detection. *Symmetry* **2019**, *11*, 939.

2. Mohanty SP, Hughes DP and Salathé M (2016) Using Deep Learning for Image-Based Plant Disease Detection. *Front. Plant Sci.* 7:1419. DOI: 10.3389/fpls.2016.01419

3. Sibiya, M.; Sumbwanyambe, M. A Computational Procedure for the Recognition and Classification of Maize Leaf Diseases Out of Healthy Leaves Using Convolutional Neural Networks. *AgriEngineering* **2019**, *1*, 119-131.

4. Kulkarni, O. (2018). Crop Disease Detection Using Deep Learning. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 1-4.

5. Ashwin Dhakal, Prof. Dr. Subarna Shakya, (2018). Image-Based Plant Disease Detection with Deep Learning. *International Journal of Computer Trends and Technology*, 61(1), 26-29.

6. Gandhi, Rutu et al. "Plant disease detection using CNNs and GANs as an augmentative approach." *2018 IEEE International Conference on Innovative Research and Development (ICIRD)* (2018): 1-5.

7. Rajashree Patil, Dr. Sampada Gulvani . "PLANT DISEASE DETECTION USING NEURAL NETWORK: A REVIEW".*Journal of Emerging Technologies and Innovative Research.*

8. Sladojevic, Srdjan & Arsenovic, Marko & Anderla, Andras & Stefanović, Darko. (2016). Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. *Computational Intelligence and Neuroscience. 2016. 1-11. 10.1155/2016/3289801.*

9. Plant Disease Detection using CNN & Remedy Adnan Mushtaq Ali Karol1 , Drushti Gulhane2 , Tejal Chandiwade3 BE Student, Dept. of EXTC, Rajiv Gandhi Institute of Technology, Mumbai, India1 B.E. Student, Dept. of EXTC, Rajiv Gandhi Institute of Technology, Mumbai, India2 B.E. Student, Dept. of EXTC, Rajiv Gandhi Institute of Technology, Mumbai, India3,

10. Boulent J, Foucher S, Théau J, St-Charles PL. Convolutional Neural Networks for the Automatic Identification of Plant Diseases. *Front Plant Sci*. 2019;10:941. Published 2019 Jul 23. doi:10.3389/fpls.2019.00941.

11. Goncharov P., Ososkov G., Nechaevskiy A., Uzhinskiy A., Nestsiarenia I. (2019) Disease Detection on the Plant Leaves by Deep Learning. In: Kryzhanovsky B., Dunin-Barkowski W., Redko V., Tiumentsev Y. (eds) Advances in Neural Computation, Machine Learning, and Cognitive Research II. *NEUROINFORMATICS 2018. Studies in Computational Intelligence, vol 799. Springer, Cham.* https://doi.org/10.1007/978-3-030-01328-8_16

12. Serawork Wallelign, Mihai Polceanu, Cédric Buche. Soybean Plant Disease Identification Using Convolutional Neural Network. *FLAIRS-31*, May 2018, Melbourne, United States. pp.146--151. ⟨hal-01807760⟩.

13. Aravind Krishnaswamy Rangarajan, Raja Purushothaman, Aniirudh Ramesh, Tomato crop disease classification using pre-trained deep learning algorithm, *Procedia Computer Science*, Volume 133, 2018, Pages 1040-1047, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.07.070.

14. Hanson, A.M.J.; Joy, A.; Francis, J. Plant leaf disease detection using deep learning and convolutional neural network. Int. J. Eng. Sci. Comput. 2017, 7, 5324–5328

15. Selvaraj, M.G., Vergara, A., Ruiz, H. *et al.* AI-powered banana diseases and pest detection. *Plant Methods* **15,** 92 (2019). https://doi.org/10.1186/s13007-019-0475-z.

16. Ferentinos, Konstantinos. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*. 145. 311-318. 10.1016/j.compag.2018.01.009.

17. Saleem MH, Potgieter J, Arif KM. Plant Disease Detection and Classification by Deep Learning. *Plants*. 2019; 8(11):468.

18. S. Ramesh et al., "Plant Disease Detection Using Machine Learning," *2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C)*, Bangalore, 2018, pp. 41-45, DOI: 10.1109/ICDI3C.2018.00017.

19. Tlhobogang, B., & Wannous, M. (2018). Design of plant disease detection system: A transfer learning approach work in progress. *2018 IEEE International Conference on Applied System Invention (ICASI)*, 158-161.

20. Francis, Mercelin and C. Deisy. "Disease Detection and Classification in Agricultural Plants Using Convolutional Neural Networks — A Visual Understanding." *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)* (2019): 1063-1068.

21. Tulshan, Amrita S. and Nataasha Raul. "Plant Leaf Disease Detection using Machine Learning." *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (2019): 1-6.

22. Militante, Sammy V., Bobby D. Gerardo and Nanette V. Dionisio. "Plant Leaf Detection and Disease Recognition using Deep Learning." *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)* (2019): 579-582.

23. P. K. Kosamkar, V. Y. Kulkarni, K. Mantri, S. Rudrawar, S. Salmpuria and N. Gadekar, "Leaf Disease Detection and Recommendation of Pesticides Using Convolution Neural Network," *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, India, 2018, pp. 1-4, DOI: 10.1109/ICCUBEA.2018.8697504.

24. P. Jiang, Y. Chen, B. Liu, D. He and C. Liang, "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks," in *IEEE Access*, vol. 7, pp. 59069-59080, 2019, DOI: 10.1109/ACCESS.2019.2914929.

25. Fuentes A, Yoon S, Kim SC, Park DS. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors (Basel)*. 2017;17(9):2022. Published 2017 Sep 4. doi:10.3390/s17092022

26. GIUSEPPE BONACCORSO C. O. QUANDOO; (2018). *MASTERING MACHINE LEARNING ALGORITHMS* (1st ed.). Packt Publishing.

27. Josh Patterson,; Adam Gibson. (2017). *Deep Learning: A Practitioner's Approach*. O'Reilly Media; 1 edition.

28. Haykin, Simon S.. *Neural networks and learning machines*. Third Upper Saddle River, NJ: Pearson Education, 2009.

29. Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, DOI: 10.1109/5.726791.

30. Prashant Anand (2020). Retrived on August 12, 2020 from https://iq.opengenus.org/architecture-and-use-of-alexnet/

31. Siddharth Das (2020). Retrived on August 12, 2020 from https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and- more-666091488df5

32. Muneeb ul Hassan (2020). Retrived on August 12, 2020 from https://neurohive.io/en/popular-networks/vgg16/

33. Sik-Ho Tsang (2020). Retrived on August 12, 2020 from https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015-image-classification- localization-detection-e39402bfa5d8

34. Chen, C., Xiang, H., Qiu, T., Wang, C., Zhou, Y., & Chang, V. (2018). A rear-end collision prediction scheme based on deep learning in the Internet of Vehicles. *Journal of Parallel and Distributed Computing*. https://doi.org/10.1016/j.jpdc.2017.08.014 .

35. Huang, Binbin & Li, Zhongjin & Xu, Yunqiu & Pan, Linxuan & Wang, Shangguang & Hu, Haiyang & Chang, Victor. (2020). *Deep Reinforcement Learning for Performance-Aware Adaptive Resource Allocation in Mobile Edge Computing*. Wireless Communications and Mobile Computing. 2020. 1-17. 10.1155/2020/2765491.

36. Mahalakshmi, G. S., Selvi, G. M., Sendhilkumar, S., Vijayakumar, P., Zhu, Y., & Chang, V. (2018). Sustainable computing based deep learning framework for writing research manuscripts. *IEEE Transactions on Sustainable Computing*, 4(1), 4-16.

37. Kim, S., Moon, P., Jang, W., Kim, K. et al., *"An Experimental Investigation of a CW/CA System for Automobiles,"* SAE Technical Paper 1999-01-1238, 1999, https://doi.org/10.4271/1999-01-1238.

38. Tie Qiu, Aoyang Zhao, Feng Xia, Weisheng Si, Dapeng Oliver Wu, Tie Qiu, Aoyang Zhao, Feng Xia, Weisheng Si, and Dapeng Oliver Wu. 2017. *ROSE: Robustness Strategy for Scale-Free Wireless Sensor Networks. IEEE/ACM Trans.* Netw. 25, 5 (October 2017), 2944–2959. DOI:https://doi.org/10.1109/TNET.2017.2713530.

39. Li, P.-Y & Feng, Y.-L & Zhang, W.-M & Yang, C.-H. (2010). *System of obstacle avoidance used in deep-seabed vehicle based on BP neural network and genetic algorithm. Zhongnan Daxue Xuebao (Ziran Kexue Ban)/Journal of Central South University (Science and Technology)*. 41. 1374-1378.

40. Y. Mao, J. Zhang, S. H. Song and K. B. Letaief, "Stochastic Joint Radio and Computational Resource Management for Multi-User Mobile-Edge Computing Systems," in IEEE Transactions on Wireless Communications, vol. 16, no. 9, pp. 5994-6009, Sept. 2017, DOI: 10.1109/TWC.2017.2717986.

41. X. Wang, Y. Cui, Z. Liu, J. Guo, and M. Yang, "Optimal resource allocation for multi-user {MEC} with arbitrary task arrival times and deadlines," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, 2019.

42. Yi, Changyan & Cai, Jun & Su, Zhou. (2019). A Multi-User Mobile Computation Offloading and Transmission Scheduling Mechanism for Delay-Sensitive Applications. *IEEE Transactions on Mobile Computing*. PP. 1-1. 10.1109/TMC.2019.2891736.

43. Li, L., Ota, K., & Dong, M. (2019). Sustainable CNN for Robotic: An Offloading Game in the 3D Vision Computation. *IEEE Transactions on Sustainable Computing, 4*, 67-76.

44. *IEEE Transactions on Sustainable Computing*, Special Issue on Smart Data and Deep Learning in Sustainable Computing. *Qiu, M., +, TSUSC Jan.-March 2019 1-3*.

45. S. Liao, Y. Xie, X. Lin, Y. Wang, M. Zhang and B. Yuan, "Reduced-Complexity Deep Neural Networks Design Using Multi-Level Compression," in IEEE Transactions on Sustainable Computing, vol. 4, no. 2, pp. 245-251, 1 April-June 2019, DOI: 10.1109/TSUSC.2017.2710178.