Leaf Spot Disease Image Classification for Groundnut Crop using Deep Convolutional Neural Network

S.Muthukumaran[a], P.Geetha[b], E.Ramaraj[c]

[a]Research Scholar, Department of Computer Science, Alagappa University, Karaikudi, India.

Email:muthumphil11@gmail.com.

[b]Associate Professor, Department of Computer Science, Dr. UmayalRamanathan College for Women, Karaikudi, India.

Email:geeth.ganesan@gmail.com.

[c]Professor and Head, Department of Computer Science, AlagappaUniveristy, Karaikudi, India.

Email:eramaraj@rediffmail.com

Abstract: Groundnut is an important cash crop cultivated in over 100 countries across the world and India is one of the largest countries in the world producing groundnut with an average of 745 kg/ha. The groundnut crop is prone to infection by pathogens and viruses that cause disease in the leaf, stem, and root which affects the yield. The Convolutional Neural Network (CNN) used for image classification takes a long time to build the neural network when the input images are in high resolution. CNN also have difficulties in identifying the patterns when the images have different background and is there any tilt or degree of rotation present in the input image. To overcome these difficulties, this paper proposed a **Convolutional Neural Network Algorithm for Groundnut Disease Prediction (CNN-GDP)** that is designed to diagnose early and late leaf spot disease in groundnut crops. The input image is properly pre-processed and compressed with image compression techniques such as **Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT),** and **Discrete Wavelet Transform (DWT)**. To study the performance of the CNN with different input image formats the collected input image is enhanced with **Thresholding Operations, Homomorphic Filters, and Contrast Stretching** techniques. The database is reconstructed with the output images of each image processing algorithm. Each database is trained with the CNN which is built using the ResNet-50 Architecture and the performance of the CNN for each analyzing the transformed database is measured with the performance metrics. The results show that the CNN trained using DFT and DCT classifies the leaf spot infected leaves correctly with higher accuracy and a low error rate. The computation time taken to build the Convolutional Neural Network with ResNet-50 architecture using the transformed image is extremely reduced when compared with the existing CNN.

Keywords: Convolutional Neural Network, Digital Image Processing, Image Compression, Image Enhancement, Leaf Spot Disease Prediction.

---

1. Introduction

Groundnut also called the peanut is one of the major oilseed crops cultivated in India for an average of 5.8 million hectares of land which have an annual productivity of 8.26 million tonnes. The groundnut belongs to the family of Fabaceae and it has the botanical name of ArachisHypogaea. The groundnut kernels are rich in nutrient content and it is used in food products such as Chikkis, butter, and groundnut milk. Groundnut has proteins, fats, and carbohydrate and its satisfies the major dietary requirement for rural women and children. The cake obtained in the extraction of oil from groundnut is used as organic manure and the shell, stem, and leaves of groundnut are used as a foodstuff for cattle. The shell of peanut is used for manufacturing fertilizer, insulating material for building, and also used as fuel in boilers. Groundnut crop is frequently attacked by nearly 55 pathogens and causes diseases such as Tikka, root rot, stem rot, and rust which result in severe mortality of the crop and reduce the yield of groundnut up to 40 percent. This study concentrates on Early and late leaf spot disease caused by CercosporaArachidicola, PhaeoisariopsisPesonatum fungus respectively. The early leaf spot disease starts after one month of sowing and it causes chlorotic spots on the leaf which are smaller in size at the time of infection. In course of time, the choloritcspots enlarge, and in the lower region of the leaf is turned to light brown. After a certain time, several lesions appear on the petioles, stems, stipules which results in premature senescence that affects the yield. The late leaf spot starts at 42-46 days of sowing and it turns the light lesions into rough which results in premature senescence and led to the shedding of the leaves that decreases the yield.

During 1969-71, the parameters such as daily minimum air temperature, duration of relative humidity, precipitation of rain on the groundnut leaf were measured and the data is analyzed with Jensen and Boyle method to detect the leaf spot diseases. In 1974, a computer program was developed by (Parvin et al [1]) and the computerized report was compared with the advice suggested by Weather Service Meteorologists and it is used for disease prediction. In 1976 electronic sensors and microprocessors were used by Phipps and Pavell for data acquisition and their proposed system uses hygrothermographs for data processing. The advisory report for leaf spot disease obtained by analyzing the data was used by groundnut growers for disease prediction. After 1984 Digital Image Processing techniques such as Masking, Color Space Transformation, and Threshold Techniques were used to detect the leaf spot infected regions of the groundnut crops. Nowadays leaf spot disease prediction is done with Digital Image Processing Techniques along with the combination of Machine Learning Techniques. Deep Learning which is the branch of Artificial Intelligence that deals with designing an intelligent computer system that works as same as the human brain is used for Image classification having large datasets and patterns are produced to assist the farmer in decision making. The leaf spot infected leaf image is collected and converted to HSV (Hue, Saturation, and Value). The HSV value is converted to binary values and machine learning algorithms were applied to the binary values and the percentage of infected regions was identified. In this paper, Convolutional Neural Network (CNN) is used for predicting the leaf spot disease of groundnut crops. CNN takes an image as input and extracts the features of the image without any human intervention and classifies the image to a particular target class. The major contribution of this research paper includes the following:

- The input images were properly pre-processed and converted to PNG format having common background using Photoshop CS 3 version.
- The dataset containing RGB image is treated with image transformation techniques such as Discrete Fourier Transform, Discrete Cosine Transform, and Discrete Wavelet Transform techniques and a new dataset was constructed for each technique.
- Image Enhancement Algorithms such as Thresholding with morphological operations, Contrast Stretching, and Homomorphic Filters were used and the images in the groundnut leaf dataset were enhanced and new dataset was constructed for each dataset.
- The Convolutional Neural Network is constructed using ResNet_50 Architecture and the input images are classified as Healthy and infected leaves based on the features extracted by the Convolutional Neural Network.
- The performance of the CNN in classifying the input images of different datasets constructed by various image processing techniques was analyzed with the performance metrics and the results were compared.

2. Literature Review

PrudhviThirumalairaju et al[2] presented a paper and in it, they classify embryo images using a convolutional neural network. The author collects images of day 5 embryos having the morphology of 113h post insemination from 543 patients at Massachusetts General Hospital (MGH) fertility center in Boston. The CNN model was tested with different network architecture such as Inception-V3, ResNet-50, and the embryo quality was classified. The author found that the CNN built using Xception architecture performed well when compared to all other architectures used in this study. Zhe Tang et al[3] proposed a lightweight CNN model to classify the grape leaves infected by black rot, leaf bright, and black measles. The author used ShuffleNet V1 and V2 to construct the CNN and improve the architecture using Squeeze and Excitation block. Among the various architecture used to construct the CNN, the shuffleNet architecture gives better accuracy of 99.14%. The author also proved that the computational cost of the algorithm is greatly reduced while placing squeeze and excitation blocks at each level of the proposed architecture. M.P.Vaishnnave et al [4] presented a paper and in it, they used Deep Convolutional Neural Network (DCNN) for predicting groundnut disease prediction. The author collects groundnut leaves infected by eight varieties of diseases and formed a dataset. They built Deep Convolutional Neural Network using two pooling techniques namely average pooling and max pooling. The author proved that the Deep Convolutional Neural Network built using Max Pooling techniques gives high accuracy when compared with the max-pooling technique.

Hongbiao Ni [5] presented a paper and in it, the author used LeNet-5 architecture CNN for face recognition. The author used the LReLU function to reduce the computation time. The author also proved that using the A-Softmax Loss function in LeNet-5 architecture improves the accuracy of the CNN while used in a Bigdata environment. Ali M. Hasan et al[6] presented a paper and in it, they used a CNN to classify the MRI scanned images that have a brain tumor. The author used to extract the features in the MRI images by converting them to greyscale and assign the pixel values in a Co-Occurance matrix. The images are then fed into the CNN and the images which are infected with brain tumors are identified. The author also constructs the CNN with a different structure such as ALexNet, GoogLeNet, etc, and the results were tabulated. Yao Chuinjing et al[7] presented a paper and in it, they used CNN to classify the panchromatic images produced by the GF-1 satellite of china to classify various land images in which different crops were cultivated. The satellite images containing river, rice, wood, etc were collected, labeled and a dataset was formed. The dataset with labeled images were then fed to CNN and the types of crop cultivated images were classified.

3. Background Knowledge

3.1 Discrete Fourier Transform

The Fourier Transform is an important image processing method used to decompose an image into its sine and cosine components. When the input image is in a spatial domain the Fourier transformation function converts the image into a frequency domain. In the Fourier space, the center region represents the low frequency and the outer region represents the high frequency. Normally the pixel of an image is represented as frequency ranging from 0-20,000 Hz with 1Hz spacing from each other. The RGB image is converted to a grayscale image using the following equation.

X=((0.3×R)+(0.59×G)+(0.11×B)) Equation (1)

The equation for representing the image in Continuous Frequencies is given by the equation

x (k)=$\int_0^P x(t)e^{-j\omega_k t}dt$ Equation (2)

Where k is the frequency values ranging from (-∞,……+∞). The frequency value of the image is multiplied by a sinusoidal function denoted as (ω) and the value is calculated as

x(ω)=$e^{j2\pi nk/N}$ Equation (3)

The equation for representing the image in Discrete Fourier Transform (DFT) is given by

X (k)=$\sum_{n=0}^{N-1} x(n)e^{-j\omega_k n}$ Equation (4)

Where k is the frequency values ranging from (0,1…..N-1). The pixels of the images represented in the spatial domain is converted to the time domain by using Euler's formula given by

$e^{-\varphi i} = \cos(\varphi) - \sin(\varphi)i$ Equation (5)

The image after undergone Discrete Fourier Transform is represented in the time domain (t) using the equation

x(t)=$\sum_{n=0}^{N-1} x[n].(\cos(\varphi) - \sin(\varphi)\, i)$ Equation (6)

3.2 Discrete Cosine Transformation (DCT)

DCT is one of the transformation practices which convert the pixels of the image into equal frequency coefficients. The dismissal present in the adjoining pixels of an image is removed by DCT and the pixels can be encoded individually which helps to shrink the amplitude of the image while embodying in the spatial domain. The 2D transformation of an image is done using the following equation.

C(u,v)=α(u)α(v)$\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)cos\left[\frac{\pi(2x+1)}{2N}\right] cos\left[\frac{\pi(2y+1)v}{2N}\right]$ Equation (7)

The equation used to renovated the transformed image is given by

$f(x,y)$ =)$\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u,v)cos\left[\frac{\pi(2x+1)u}{2N}\right] cos\left[\frac{\pi(2y+1)v}{2N}\right]$ Equation (8)

3.3 Discrete Wavelet Transformation (DWT)

In this technique, the input image is transformed into a signal which contains an orthogonal set of wavelets that are conjoint. By applying to filter and subsampling the original input image is decomposed and the image data is compressed using a special coding principle. While applying DWT to a 2D image, the transformation is done row-wise using a convolution filter method that handles the borders of an image properly. The column-wise transformation of an image is done using a lifting scheme which saves the memory space of an image. The row and column-wise transformation of a 2D image using DWT results in four subbands (LL-coarse approximation, HL-Horizontal, HH-Diagonal, LH-Vertical). The equation for representing the signals into orthogonal is given by

$\emptyset(x) = \sum_{k=-\infty}^{\infty} a_k \emptyset(S_x - k)$ Equation (9),

Here S is a scaling function that is obtained by

$\varphi(x) = \sum_{k=-\infty}^{\infty} (-1)^k a_{N-1-k} \varphi(2x - k)$ Equation (10)

3.4 Image Enhancement using Thresholding Techniques

This is one of the important techniques used to separate the image from its background and its foreground. The RGB image is converted to a binary value and a threshold value is automatically selected by the algorithm. If the binary value of the image is less than the threshold value then it is assigned as zero, otherwise, if the binary value of the image is greater than the threshold then it is assigned as 255. The equation for assigning the threshold value is given by the following equation.

$\int_{maxval}^{0} \begin{matrix} if\ src(x,y) > thresh \\ Otherwise \end{matrix}$ Equation (11)

3.5 Image Enhancement using Homomorphic Filters

This is one of the popular image enhancement techniques used for improving the quality of a greyscale image. This method removes the multiplicative noise present in an image and helps to improve the quality of an image by combining the illumination with reflection which is done using the following equation.

$m(x,y) = i(x,y) * r(x,y)$ Equation (12)

3.6 Image Enhancement using Contrast Stretching

It is one of the popular image enhancement techniques used to improve the contrast of an image by changing the pixel values to either zero or 255. The difference between the maximum and minimum pixel value is calculated for an image and that value is set as a threshold. The formula to stretch the pixel value to its maximum and minimum intensity is calculated using the following equation.

$X_{new} = \frac{Xinput - Xmin}{Xmax - Xmin} \times 255$ Equation (13)

3.7 Convolutional Neural Network

Convolutional neural networks (CNN) is an advancement of Neural Network which is specially designed for image classification by YannLeCun in 1988. In this paper the CNN is built using ResNet-50 architecture, here an image is converted to an array of pixels having three values namely width, height, and RGB channel values. For example, if an input image having a size 300×300 pixel image is read as a 3D array (300×300×3) where the first value 300 represents the Width, the second value 300 represents the height, the third value 3 represents the RGB values and it is called as a tensor. A value from 0 to 255 is assigned to these 3D array values that describe the intensity of the pixel at each point. The input image is then passed through the CNN consist of series of layers and finally classifies the image to a particular target class.

3.7.1 Convolutional Layer

The CNN layer is a feed-forward artificial neural network that works as same as the animal visual cortex and identifies the connectivity pattern between the neurons [8,9]. Unlike Artificial Neural Network, the neuron in the CNN is connected only to a small portion of the neural network. A small portion of the 3D array of the image is selected and it is called a tensor or filter. The filter starts a smaller portion of the image at the top left corner and begins to multiply the original pixel value by the filter value and the multiplied value is summed up to produce a single value called the convolution value. The process repeats for all the 3D values of the input image with different filter value and produces different convolved feature map. Suppose m is the selected kernel width and height, h is the convolutional output for an input image x, then the equation for convolutional operation is given by

$h_{i,j} = \sum_{k=1}^{m} \sum_{l=1}^{m} w_k, lx_i + k - 1, j + l - 1$ Equation (14)

3.7.2 ReLU Layer

In Convolutional Neural Network the output of the hidden layers was detected with the help of activation function. The Rectified Linear Activation Unit [8] uses a linear activation function that converts the output values of the hidden layers from 0 to 1. The equation for the ReLU activation function is given by

$f(x)=\{^{0\ for\ x<0}_{x\ for\ x\geq 0}$ Equation (15)

3.7.3 Pooling Layer

The pooling layer is one of the building blocks of the CNN which is used to reduce the parameters of the feature maps by downsampling them [9]. By reducing the parameters of the feature map the computation of the CNN is reduced to a large extent. Max pooling is used in this proposed architecture and the feature map having dimension $n_h \times n_w \times n_c$ is applied with the pooling function and will produce the output given by

$\frac{(n_h-f+1)}{s} \times \frac{(n_w-f+1)}{s \times n_c}$ Equation (16)

Here n_h is the altitude of the feature map and n_w is the girth of the feature map and the channels used in the pooling function are represented by n_c. The filter size is represented as f and the stride length is represented as s.

3.7.4 Flattening Layer

The next step involved in building a CNN for image classification is called Flattening. Normally flattening is done to convert the reduced feature map to a one-dimensional array which is then feed-forward to the next layer of the CNN. Suppose the input image x weights w and bias as b then the equation for converting the feature map to a one-dimensional array is given by

$Z=W^T.X + b$ Equation (17)

3.7.5 Fully Connected Layer

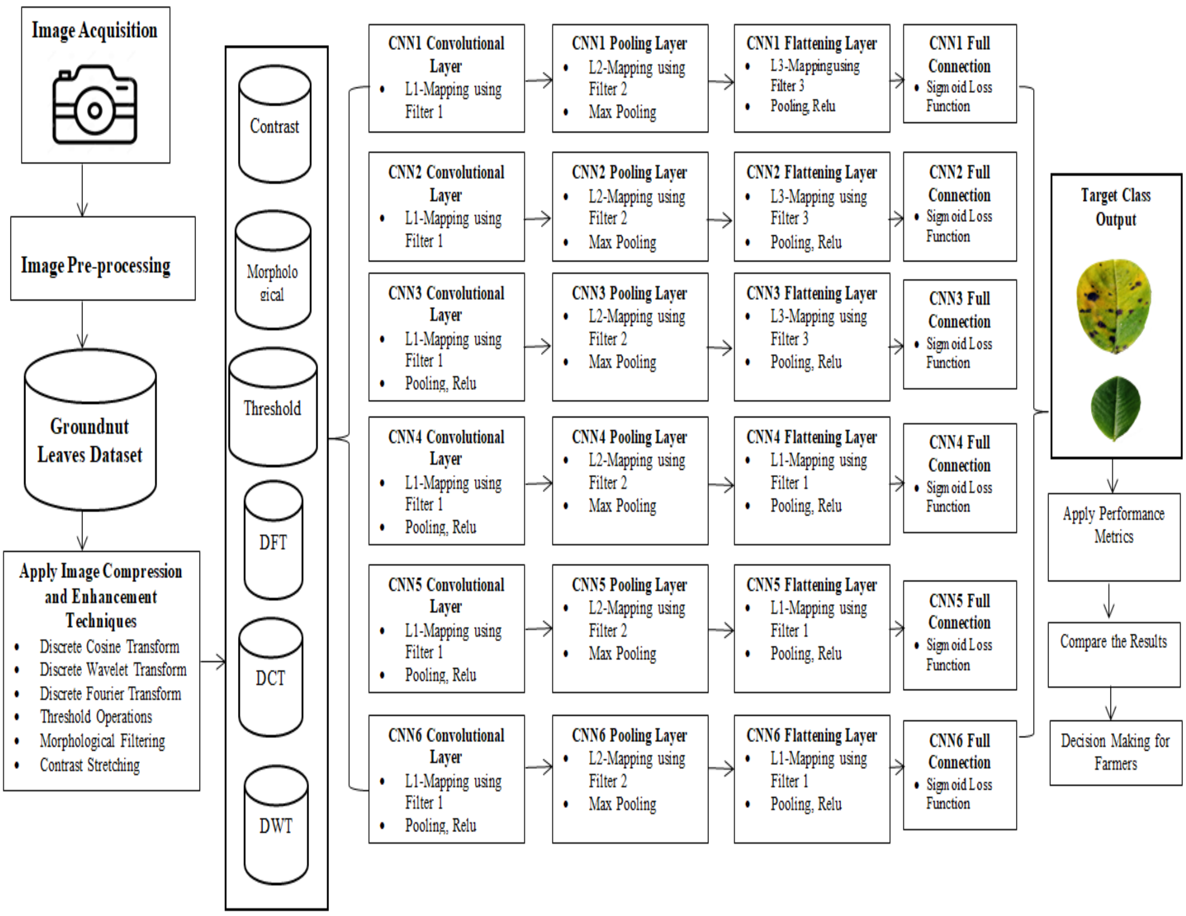
The fully connected layer accepts the input from the flattening layer and it applies an activation function to predict the target class. The weights of the neurons, the accuracy of the model, and the error rate were calculated. The information is stored and then feed-forward to the neural network. The process repeats until the maximum accuracy of the model is reached. The information extracted from the model is used to predict the target class of the input image. Suppose the input image is in n×n dimension and the filter applied to the image is f×f. The padding and stride applied at the convolutional layer is p and s then the equation for predicting the output is given by

$O=\frac{n-f+2p}{s} + 1$ Equation (18)

4. Proposed Convolutional Neural Network Algorithm for Groundnut Disease Prediction (CNN-GDP)

The Convolutional Neural Network used for image classification has some limitations such as the computational time for building the neural network is very large when the image used in the dataset has high resolution. For effectively classifying the images, the CNN requires a large number of images to be present in the dataset to train the model. Providing few images to the model will reduce the performance of the classifier. If the images provided to the CNN have the same background or it has some degree of tilt, then the CNN will fail to identify the images.

Figure 1: Proposed Convolutional Neural Network Algorithm for Groundnut Disease Prediction (CNN-GDP)



The proposed CNN-GDP shown in Figure 1 above overcomes these difficulties by applying proper pre-processing of the input image. The data used for this study is collected from the agricultural land situated at Aapatharanapuram village located at Vadalurtaluk in Cuddalore District. The groundnut crop is planted in the Tamil season Margazipattam (December 2020 to January, 2021) and the plant is monitored daily. The images used in this study were taken in February 2021. The groundnut leaves were collected and placed on a board covered with white cloth and photographed by OPPO A12 mobile having model number CPH2083 at 11.AM. The resolution of the camera used to take the picture is 13-megapixel (f/2.2, 1.12um-micron) + 2-megapixel (f/2.4)). The leaf portion of the image is selected, zoomed, and saved in PNG format using Photoshop CS 3 version. The images present in the dataset are then applied with image compression algorithms such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT). The output image of each compression algorithm is stored as a new database. To analyze the performance of the CNN in predicting the leaf spot disease, image enhancement algorithms such as Thresholding Operations, Morphological Filters, and Contrast Stretching were used and the output images of the enhanced images were stored as each database. The Convolutional Neural Network is build using the ResNet-50 architecture for each database. The image is classified as Healthy and Infected leaves based on the features extracted by the Convolutional Neural Network and leaf with Leaf Spot infected diseases is identified using the proposed convolutional neural network architecture. The sample images present in the database are shown below.

Figure 2: Samples of Healthy Leaf Images

Figure 3: Samples of Leaf Spot Disease Infected Images

5. Results

The proposed Convolutional Neural Network Algorithm for Groundnut Disease Prediction (CNN-GDP) is implemented in Python 3.8. Anaconda Navigator is used as the IDE and coding was done using Spyder 5.0.0. The input image is read using the methods present in the cv2 library and the RGB image is converted to a Fourier transformed image by np.fft.fftshift() method present in numpy library. The Fourier transformed image is converted to RGB using np.fft.ifft2() method.

Figure 4: Original input image

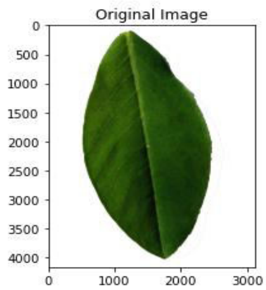


Figure 5: Grey Scale image

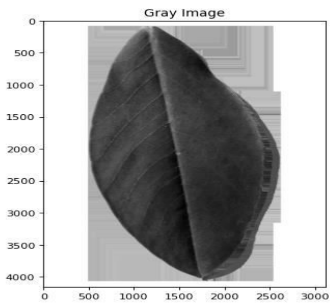
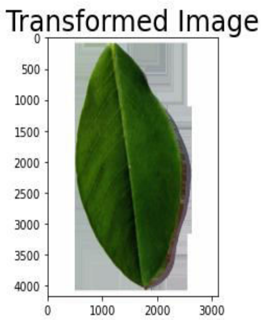


Figure 6: Fourier Transformed Image

Figure 7: Fourier Transformed Image with RGB



To convert the input image into discrete cosine transform the dct and idct methods were used which is present in the scipy.fftpack library. The discrete wavelet transform is done using the pywt.wavedec2() and pywt.waverec2() method present in the pywt library. Level 2 wavelet transformation is used to transform the input image with periodization mode. The horizontal, vertical, Approximation and Diagonal coefficients were calculated from the pixel values of the image and finally, the image is reconstructed to wavelet transform form. The image enhancement algorithms such as Thresholding with morphological operations, Homomorphic filters, and Contrast Stretching were used in this paper. To enhance the image using Thresholding method is implemented using the built-in function present in the cv2 library is used. First, the input image is read and it is blurred using the cv2.medianBlur() method, and Thresholding is done using cv2.threshold() method by using parameters such as cv2.THRESH_BINARY, cv2.THRESH_OTSU. The Homomorphic filter is implemented by taking the pixel values of the image and creates a black circle on white background for filtering the image using cv2.circle() method. The filtered image is anti-aliased using GaussianBlur() method present in cv2 library. The complex and imaginary components of the image is extracted and the image is reconstructed after implementing the Homomorphic filter using cv2.normalize() method. The contrast stretching algorithm is implemented using PIL library. The bands of the image such as RedBand, GreenBand, BlueBand were split using the imageObject.split() method and normalized using the user-defined function and finally reconstructed the image. The output of the input image obtained after implementing the various image processing techniques is shown below in the following figures.

Figure 8: Input Image

Figure 9: DCT image

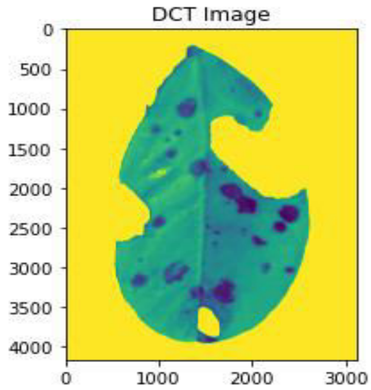


Figure 10: DWT image

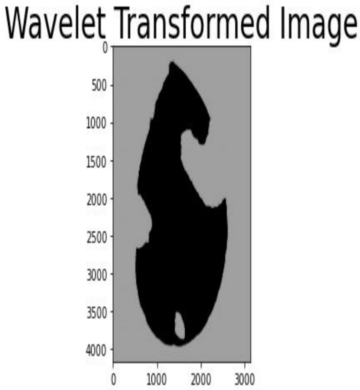


Figure 11: Thresholding Operated Image

Figure 12: Homomorphic Filtered Image

Figure 13: Contrast Stretched Image

The Convolutional Neural Network is built using the ResNet-50 architecture with Tensorflow 2.3.0. Pre-processing of the image is done using Keras library. The first step deals with building the convolution layer of the CNN. For this the number of filters used is 32, the size of the kernel is 3 and ReLU activation function is used to transform the output of the hidden layer to a linear state. The size of the feature map is (64×64×3). Flattening is done using the Maxpool method and the dense units used in the fully connected layer is 128. The sigmoid function is used in the output layer of the proposed Convolutional Neural Network. The various process of the input image belonged to the target class Infected which is feed into the proposed CNN is shown below.

Figure 14: Fourier Transformed Image with RGB Value of Disease Leaf

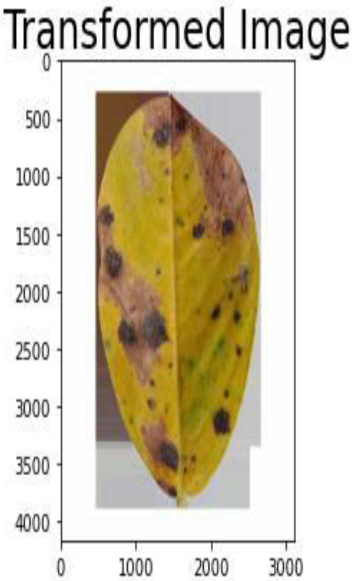


Figure 15: Image of the Disease Infected Leaf in Convolutional Layer of the built CNN-GDP

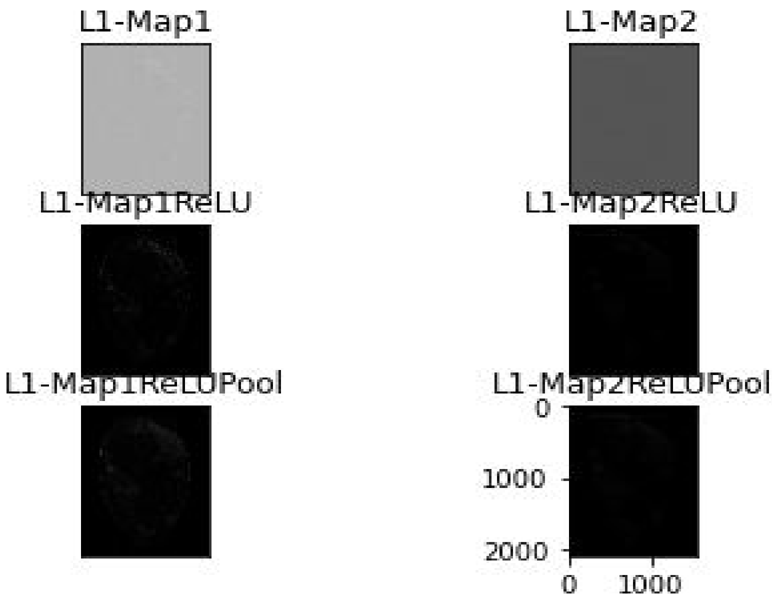


Figure 16: Image of the Disease Infected Leaf in Flattening Layer of the built CNN-GDP

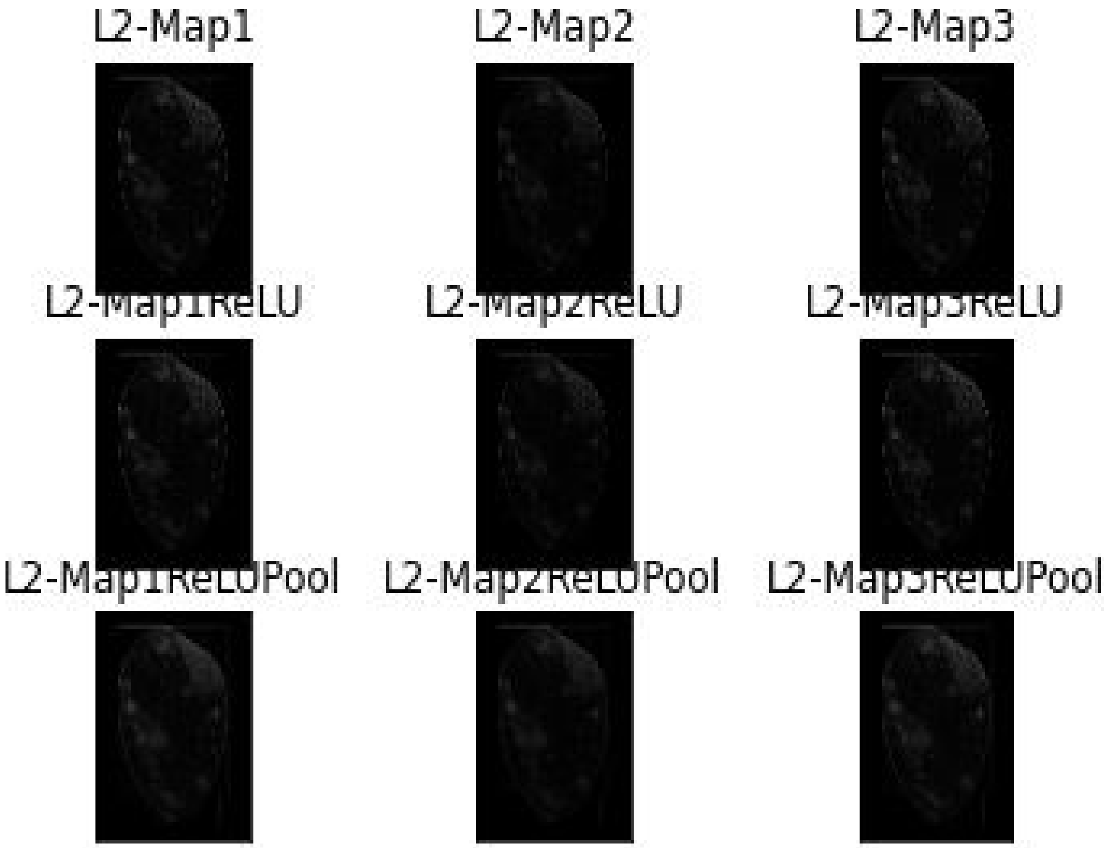
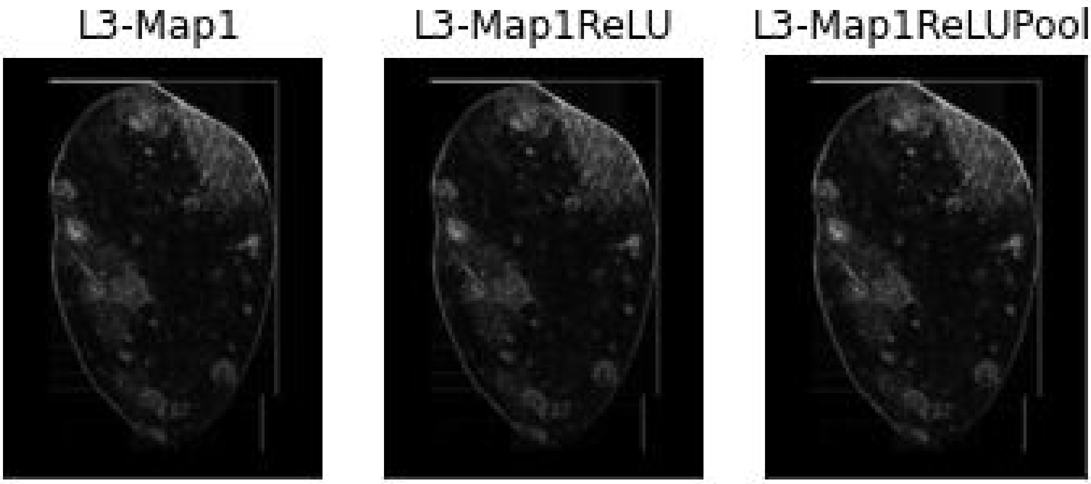


Figure 17: Image of the Disease Infected Leaf in Fully Connected Layer of the built CNN-GDP



5.1 Evaluation of Results

To measure the performance of CNN, the neural network is trained using six different types of datasets. The first dataset is named Raw Dataset which contains images directly captured using the OPPO A12 mobile having modeled number CPH2083 **DISCUSSION.** Thedataset is named as PNG dataset which contains images that are pre-processed using Photoshop CS3 version. The remaining four datasets contain names of the techniques which are used for image compression and enhancement. The first CNN is built using a dataset containing 200 images which contain two target classes Healthy and Bacteria. The dataset is divided into a training set containing 140 images and a testing set containing 60 images and the performance of the CNN is measured using the performance metrics and the results are given below.

Table 1: Performance of the CNN-GDP for 200 Images

	Raw Dataset	PNG Dataset	DFT Dataset	DCT Dataset	DWT Dataset	Contrast Dataset	Homomorphic Dataset	Threshold Dataset
Time	11m:45s	25m:51s	19 secs	19 secs	20 secs	14m:18s	8m:25s	8m:40s
Accuracy	0.835	0.771	0.785	0.785	0.75	0.835	0.771	0.75
Error Rate	0.165	0.229	0.215	0.215	0.25	0.165	0.229	0.25
Recall	0.835	0.771	0.785	0.785	0.749	0.835	0.771	0.75
Specificity	0.835	0.771	0.785	0.785	0.749	0.835	0.771	0.75
Precision	0.836	0.771	0.785	0.784	0.749	0.835	0.771	0.75
Negative Predicted Value	0.836	0.771	0.785	0.784	0.749	0.835	0.771	0.75
False Positive Rate	0.763	0.828	0.813	0.814	0.849	0.764	0.828	0.84
False Negative Rate	0.763	0.828	0.813	0.814	0.849	0.764	0.828	0.84
False Discovery Rate	0.763	0.828	0.814	0.814	0.849	0.763	0.828	0.84

From Table 1 it is clear that the dataset which is constructed using the images of Discrete Fourier Transform and Discrete Cosine Transform performs best and produces the best result with low computation time in 19 seconds. While considering the image enhancement algorithm the dataset constructed using the Homomorphic Filter algorithm performs best with low computation time in 8minutes:25seconds. For the second time, the CNN is built using 300 images. Here the dataset is divided into a training set containing 220 images and a testing set containing 80 images and the performance of the CNN is given below in Table 2.

Table 2: Performance of the CNN-GDP for 300 Images

	Raw Dataset	PNG Dataset	DFT Dataset	DCT Dataset	DWT Dataset	Contrast Dataset	Homomorphic Dataset	Threshold Dataset
Time	17m:32s	37m:57s	26secs	29secs	27secs	21m:20s	12m:50s	9m:52s
Accuracy	0.831	0.809	0.745	0.804	0.804	0.8	0.80	0.76
Error Rate	0.169	0.191	0.255	0.196	0.196	0.2	0.20	0.24
Recall	0.831	0.809	0.745	0.804	0.804	0.8	0.805	0.763
Specificity	0.831	0.809	0.745	0.804	0.804	0.8	0.805	0.763
Precision	0.831	0.809	0.745	0.804	0.804	0.8	0.805	0.763
Negative Predicted Value	0.831	0.809	0.745	0.804	0.804	0.8	0.805	0.763
False Positive Rate	0.767	0.790	0.854	0.795	0.795	0.8	0.790	0.836
False Negative Rate	0.767	0.790	0.854	0.795	0.795	0.8	0.790	0.836
False Discovery Rate	0.767	0.790	0.854	0.795	0.795	0.8	0.790	0.836

From Table 2, it is clear that CNN built using Discrete Wavelet Transformed images classifies the 300 images efficiently than the remaining algorithm and produces the best result. For the third time, the CNN is built using a dataset containing 400 images in which the training set contains 300 images and the testing set contains 100 images, and the performance of the CNN is given below.

Table 3: Performance of the CNN-GDP for Classifying 400 Images

	Raw Dataset	PNG Dataset	DFT Dataset	DCT Dataset	DWT Dataset	Contrast Dataset	Homomorphic Dataset	Threshold Dataset
Time	23m:17s	45m:30s	36secs	39secs	40secs	28m:27s	17m:06s	11m:59s
Accuracy	0.764	0.78	0.766	0.82	0.853	0.796	0.803	0.803
Error Rate	0.236	0.22	0.234	0.18	0.148	0.204	0.197	0.197
Recall	0.764	0.78	0.766	0.819	0.853	0.796	0.803	0.803
Specificity	0.764	0.78	0.766	0.819	0.853	0.796	0.803	0.803
Precision	0.764	0.78	0.766	0.819	0.854	0.796	0.803	0.803
Negative Predicted Value	0.764	0.78	0.766	0.819	0.854	0.803	0.803	0.803
False Positive Rate	0.834	0.78	0.833	0.779	0.746	0.803	0.796	0.796
False Negative Rate	0.834	0.78	0.833	0.779	0.746	0.803	0.796	0.796
False Discovery Rate	0.834	0.78	0.833	0.779	0.745	0.803	0.796	0.796

From Table 3, it is clear that DFT and DCT perform better and produce the best result when compared with the rest of the algorithms. For the fourth time, the CNN is built using 500 images in which the training set contains 370 images and the testing set contains 120 images. The performance of the CNN for classifying the 500 images is given below in Table 4.

Table 4: Performance of the CNN-GDP for Classifying 500 images

	Raw Dataset	PNG Dataset	DFT Dataset	DCT Dataset	DWT Dataset	Contrast Dataset	Homomorphic Dataset	Threshold Dataset
Time	29m:09s	56m:52s	44secs	48secs	46secs	35m:57s	21m:22s	14m:55s
Accuracy	0.794	0.81	0.786	0.802	0.805	0.8	0.818	0.786
Error Rate	0.206	0.19	0.214	0.198	0.195	0.8	0.182	0.214
Recall	0.794	0.81	0.786	0.802	0.805	0.8	0.818	0.786
Specificity	0.794	0.81	0.786	0.802	0.805	0.8	0.818	0.786
Precision	0.794	0.81	0.786	0.802	0.805	0.8	0.818	0.786
Negative Predicted Value	0.794	0.81	0.786	0.802	0.805	0.8	0.818	0.786
False Positive Rate	0.805	0.789	0.813	0.797	0.794	0.8	0.781	0.813
False	0.805	0.789	0.813	0.797	0.794	0.8	0.781	0.813

Negative Rate								
False Discovery Rate	0.805	0.789	0.813	0.797	0.794	0.8	0.781	0.813

From Table 4, it is clear that the DCT algorithms perform better and produce the best result when compared to the rest of the algorithms. Finally, the CNN is trained using 600 images in which the training set contains 400 images and the testing set contains 200 images.

Table 5: Performance of the CNN-GDP for 600 Images

	Raw Dataset	PNG Dataset	DFT Dataset	DCT Dataset	DWT Dataset	Contrast Dataset	Homomorphic Dataset	Threshold Dataset
Time	35m:15s	1h:08m:17s	51secs	57secs	54secs	42m:59s	25m:27s	11m:59
Accuracy	0.8	0.77	0.844	0.817	0.807	0.795	0.8	0.803
Error Rate	0.2	0.23	0.156	0.183	0.193	0.205	0.2	0.197
Recall	0.8	0.77	0.844	0.817	0.807	0.795	0.8	0.803
Specificity	0.8	0.77	0.844	0.817	0.807	0.795	0.8	0.803
Precision	0.8	0.77	0.844	0.817	0.807	0.795	0.8	0.803
Negative Predicted Value	0.8	0.77	0.844	0.817	0.807	0.795	0.8	0.803
False Positive Rate	0.8	0.83	0.754	0.782	0.792	0.805	0.8	0.796
False Negative Rate	0.8	0.83	0.754	0.782	0.792	0.805	0.8	0.796
False Discovery Rate	0.8	0.83	0.754	0.782	0.792	0.805	0.8	0.796

From Table 5, it is clear that the dataset created with Discrete Fourier Transformed images gives the best result with less computation time in 51 seconds when compared with the rest of the image processing techniques. The validation accuracy and validation loss for building the neural network for the Fourier Transform technique are given below.

Figure 18: Validation Accuracy of CNN-GDP for building the network with DFT images having 600 images

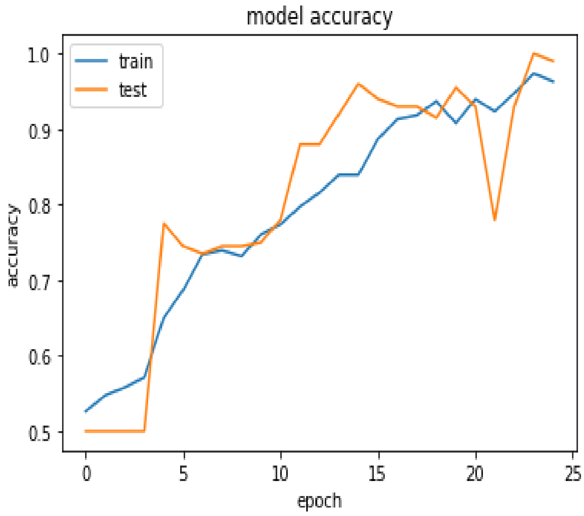
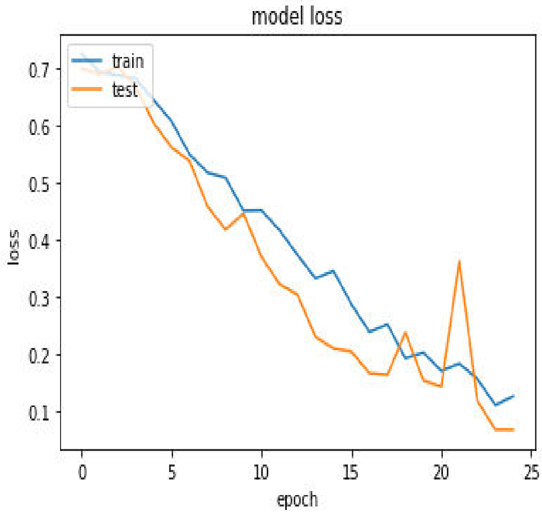


Figure 19: Validation Loss of CNN-GDP for building the network with DFT images having 600 images



From Figure 20 it is clear that the accuracy of the model increases with increases in epoch and for the 25th epoch the model reaches its maximum accuracy. From Figure 21, it is clear that the loss of the model decreases with an increase in epoch, and for the 25th epoch, the built model has no loss and produces the best result.

6. Discussion

While comparing the overall results of the CNN for all datasets, it is clear that CNN can learn automatically and detect the patterns present in the image irrespective of the image format. The result produced by all types of the dataset is the same and only the computation time is varied. The Discrete Fourier Transform and Discrete Cosine Transform applied to the input image drastically reduced the computation time of the CNN and produce an accurate result better than the Raw data. It is suggested that while using CNN for real-time applications which involve a large number of images, transforming the input image with Discrete Fourier or Discrete Cosine Transformation will produce the best result with low computation time. While building the CNN, it is recommended to use a high epoch value to reduce the validation loss of the built model. The leaf spot disease epidemics cause severe loss in the yield of groundnut crops and it can be prevented by using pesticides that contain carbendazim in the ratio of 0.1% or mancozeb in the ratio of 0.2%. The leaf spot disease severity can be controlled by spraying the pesticide which contains the combination of Trichodermaviride and Verticilliumlecanii with 5 percent each. While dealing with organic farming, the combination of neem extract, Mehandi, neem oil, and neem kernel extract in the ratio (5:2:1:3) can effectively control the leaf spot disease.

7. Conclusion

The proposed Convolutional Neural Network Algorithm for Groundnut Disease Prediction (CNN-GDP) overcomes the difficulties present in the classical Convolutional Neural Network. Using more images with common background improves the performance of the CNN. Applying image compression techniques to the input image decreases the computation time taken to build the CNN and also increases the performance of the CNN. It is also clear that CNN can extract the features present in the image automatically, irrespective of the image formats. This paper also suggests various disease control measures that should be followed to control the epidemic caused by leaf spot disease and thereby helps to improve the yield of groundnut crop.

References

Smith, D. H (1985).Disease-forecasting method for groundnut leaf spot diseases, Agrometeorology of Groundnut, 239-242.

Thirumalaraju, P., Kanakasabapathy, M. K., Bormann, C. L., Gupta, R., Pooniwala, R., Kandula, H., ...&Shafiee, H (2021). Evaluation of deep convolutional neural networks in classifying human embryo images based on their morphological quality, Heliyon, 7(2), 1-12.

Tang, Z., Yang, J., Li, Z., & Qi, F (2020).Grape disease image classification based on lightweight convolution neural networks and channelwise attention, Computers and Electronics in Agriculture, *178*,1-9.

Vaishnnave, M. P., Devi, K. S., &Ganeshkumar, P (2020).Automatic method for classification of groundnut diseases using deep convolutional neural network, *Soft Computing*, 24(21), 16347-16360.

Catak, F. O., Sivaslioglu, S., &Sahinbas, K (2021). A Generative Model based Adversarial Security of Deep Learning and Linear Classifier Models, Informatica, 45, 33-64.

Ni, Hongbiao (2020). Face Recognition Based on Deep Learning Under the Background of Big Data, Informatica, 44(4), 491-496.

Kuo, C. C. J (2016). Understanding convolutional neural networks with a mathematical model, Journal of Visual Communication and Image Representation, *41*,406-413.

He, K., Zhang, X., Ren, S., & Sun, J (2015), Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, Proceedings of the IEEE international conference on computer vision,1026-1034.

Scherer, D., Müller, A., &Behnke, S (2010).Evaluation of pooling operations in convolutional architectures for object recognition, International conference on artificial neural networks,Springer, Berlin, Heidelberg, 92-101.

Srirangam, S. (2021). Artificial Intelligence Based Hyper-parameter Adjustment on Deep Neural Networks: An Application of Detection and Classification of COVID-19 Diseases. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(14), 816-825.

Micheala, A. A., & Shankar, R. (2021).Automatic Age and Gender Estimation using Deep Learning and Extreme Learning Machine. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(14), 63-73.

Acknowledgements

This article has been written with the financial support of RUSA-Phase 2.0 grant sanctioned vide Letter No.F.24-51/2014-U, Policy (TNMulti-Gen), Dep.ofEdn.Govt. of India, Dt.09.10.2018.

Author's Profile

S.Muthukumaran is a Research Scholar in the Department of Computer Science, Alagappa University, Karaikudi. His main Research Interest includes Machine Learning, Big Data Analytics, and Internet of Things. He has published 13 International Journals.

P.Geetha is working as an Associate Professor in the Department of Computer Science at Dr.UmayalRamanathan College for women, Karaikudi. Her Research Interest includes Data Mining, Network Security and Internet of Things. She has published 20 International Journals.

E. Ramaraj is working as the Professor and Head of the Department of Computer Science, Alagappa University, Karaikudi. He has very sound knowledge in many research fields especially in Data Mining, Network Security, Remote Sensing, and Big Data Analytics. He has published more than 100 international journals.