IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

DEEP CONVOLUTED NEURAL NETWORKS (DCNN)- BASED FEATURE EXTRACTION AND CLASSIFICATION FOR THE DETECTION OF BANANA LEAF DISEASE

[1]K. Jeya Carolin Agnes, [2]S. Agnes Joshy
[1] PG Scholar, [2]Assistant Professor
[1,2]Department of Information Technology
[1,2]Francis Xavier Engineering College, Tirunelveli, India

***Abstract**:* Approximately 70% of India's population depends on agricultural output. In agricultural settings, insects that spread the illness to other plants have impacted plants and their leaves. The farm's output has dropped as a result of these illnesses' infestation. Consequently, it's critical to diagnose these illnesses as soon as possible. In the farm, the discovery of banana diseases has proven to be more confusing. The identification of banana plant diseases using image processing is becoming increasingly effective, and it is crucial for farmers to assess plant development without incurring financial costs for manual assistance. This study advocated the use of deep convolutional neural networks (DCNNs) for classification and feature extraction using GLCM for the detection of leaf disease. The collection of the dataset was done using prehistoric.

***Index Terms* – Banana plant disease, DCNN, GLCM, leaf disease, and the extraction and categorization of features.**

I. Introduction

The nation's economy depends on the output of its agricultural sector. Consequently, since plant infection is common, disease diagnosis becomes extremely important in agriculture. The research field of disease detection without personal assistance and categorization has become essential, requiring computer vision with automation or a system with machine vision by applying image processing techniques [2]. The most prevalent illnesses that affect banana leaves include banana streak virus, banana bract mosaic virus disease, infectious chlorosis, black spot, banana bunchy top, moko disease, sigatoka disease, and panama disease. bananas are susceptible to a number of diseases, including banana speckle and banana sigatoka. Mycosphaerella fijiensis is the fungus that causes black sigatoka. Their initial signs and symptoms early on, it would show up as tiny, light brown spots; as time goes on, the spots would get bigger and eventually turn black. If treatment for this disease is not received, the plant may die. If not, this could be addressed as soon as it is discovered, perhaps saving the plant [5].Artificial intelligence (AI), which is integrated with deep learning prototypes where human activities are given significant weight, has been used to identify plant diseases based on their appearance and unusual symptoms. AI-enabled mobile applications help farmers identify and prevent crop diseases by providing early warnings. Farmers still require manual labor even though they cannot afford these apps. According to the Global System for Mobile Association (GMSA), there will be 5 billion smartphone subscriptions worldwide by 2020, with about 1 billion of those subscriptions occurring in Africa. Artificial intelligence (AI), the Internet of Things (IoT), robotics, satellites, cloud computing, and machine learning are just a few of the technologies that have helped farmers and agriculture grow [11].

Deep learning is a cutting-edge method 7'of image processing that has been applied to object recognition and has increased the classification accuracy of several crop diseases. Transfer learning is one of the deep learning techniques that is commonly used. Pre-model design has been obtained as a novel application of this technique. Deep transfer learning has been developed for both the analysis of predictions made and the new architecture of image processing. With the use of several training image datasets, DTL has proven to be an incredibly effective method for determining the farm illness that mobile apps have spread. And the farm is now using this in a very discrete manner in real time. The effectiveness of AI in identifying crop diseases, like those that affect wheat, has been assessed in the current study. Additionally for datasets containing both disease-infected and non-infected plants. Using computer vision design as the foundation, crop disease detection has been accomplished through feature extraction, which produces useful results. Nevertheless, feature extraction is computationally very challenging and requires a specialist's understanding of healthy representation. There are some restricted and trained dataset images of diseased crops accessible. More than 50,000 pictures even though many photographs have been thought of as having a straightforward background with leaves removed and images that have been trained with CNN, these results are not obtained when using actual farm images. Massive photos of both healthy and diseased plants were obtained by simulating a robust and highly implemental identification system. The images were collected from different areas of infected plants as well as from budding plants under different environmental conditions. All of the photos have required labeling and pre-training by plant pathology professionals. The current methods for identifying diseases in crop designs have primarily focused on leaf symptoms up to now. Unsuitably, numerous signs have also been observed in other plant sections, with banana pest and disease-related symptoms serving as the best example [16]. The remaining portions of the paper are arranged as follows: we provide the relevant work in Section 2. The technical specifics of the architecture and the suggested strategy are explained in Section 3. Additionally, Section 4 reports the experimental evaluation and results. Section 5 wraps up the report and offers a look ahead to future research.

II.RELATED WORKS

Finding dangerous bacteria and viruses in a plant is crucial to guaranteeing that it is safe and environmentally friendly for farming [3]. Recently, much thought has gone into developing the current designs, which can achieve both quick and practical pathogen detection as well as target extraction using a crucial sample that greatly improves detection. Serological and molecular approaches have gained substantial traction in recent times, as a vast number of samples need to be analyzed. Antibiotics such as monoclonal and recombinant are utilized for the majority of plant pathogens, and they undoubtedly aid in the detection of specific serologies. Purifying the nucleic acid automatically from pathogens using robotics or columns has improved molecule-based detection [1]. PCR is a novel modification that involves simple otherwise multiplex integration in a single closed tube, co-operative PCR that also uses amplicons or numerical PCR to observe in real-time, and access to increased sensitivity for detecting one or more pathogens in a separate evaluation. Recent advancements in nucleic acid evolution have been achieved through the use of microarray assessment technology; however, in order to increase the sensitivity for detection, this requires the extraction of genes from DNA/RNA as well as their pre-amplification processes.

The downsides of molecular approaches include the necessity for manual support, a lengthy processing time, and a prolonged process, especially when it comes to sample preparation that yields feasible output. Since molecular approaches cannot be utilized for initial testing in processing many plant samples due to the time necessary for this process, they are a dependable tool for assuring plant diseases [14]. Based just on their leaves, Lee et al. (2015) [7] designed CNN to detect plant diseases without the need for human assistance. A relatively basic yet successful neural network was published by Grinblat et al. (2016) [4] for the efficient detection of three different types of legumes based on patterns in leaves that are morphologically characterized by their veins. In order to detect 26 variants of plant illness, Mohanty et al. (2016) [8] compared two commonly used and deployed system designs for CNNs by expanding the leaf database with photos that show 14 variants in plants. Based on an evaluation of their 99.35% automatic detection rate, the outputs are extremely efficient. However, their main picture disadvantage is that it only works with single images for simulation purposes—it is not suitable for the real-time implementation of agricultural farms. The same method for identifying plant illnesses based on leaf photos was presented by Sladojevic et al. (2016) [15] using a similar amount of data that is available through web services and includes a minimum number of plant variations and diseases. Their output efficiency, which depends on the data utilized for testing, is somewhere between 91% and 98%. Currently, a few conventional pattern detection techniques are related by [14].

These techniques rely on CNN designs and their ability to recognize plants by three different dataset images, which can include plants, fruits, or plant leaves. In the end, CNN is a highly conventional technique. Lastly, Seetharaman et al. (2021) [12] created a CRNN prototype that can identify nine different banana fruit diseases and pests with a better level of satisfaction. The image processing technique requires the greatest amount of simulation effort based on a few initials that could reduce the expense of the experiment. A method for connecting texture, color, and form features was developed by certain researchers. When examining images from similar databases, the Region-Based Convolution Neural Network (RCNN) in (2022) [13] achieves an accuracy rate of more than 99%. Even though the model has been tested against the information acquired from reliable websites, their accuracy has dropped by 31.4% in this instance. The dataset complete its application for digital real-time farming after being taken into consideration in specific circumstances and for the detection of an extreme stage of sickness. Additionally, the existing datasets do not take into account the presence of multiple diseases in a single plant; as a result, trained and tested prototypes have been used to detect diseases with greater visibility, even though they are not strictly necessary for crop production.

III.PROPOSED METHODOLOGY:

The purpose of this suggested method is to isolate and categorize banana diseases at the earliest stage in order to prevent the illness from spreading to neighboring plants. The prehistoric agricultural data was used to compile the banana leaf dataset. The leaf photos have been saved in the database, where 10 samples have been examined for the purpose of detecting diseases that have required further processing. Figure 1 shows the architectural diagram for the suggested system.

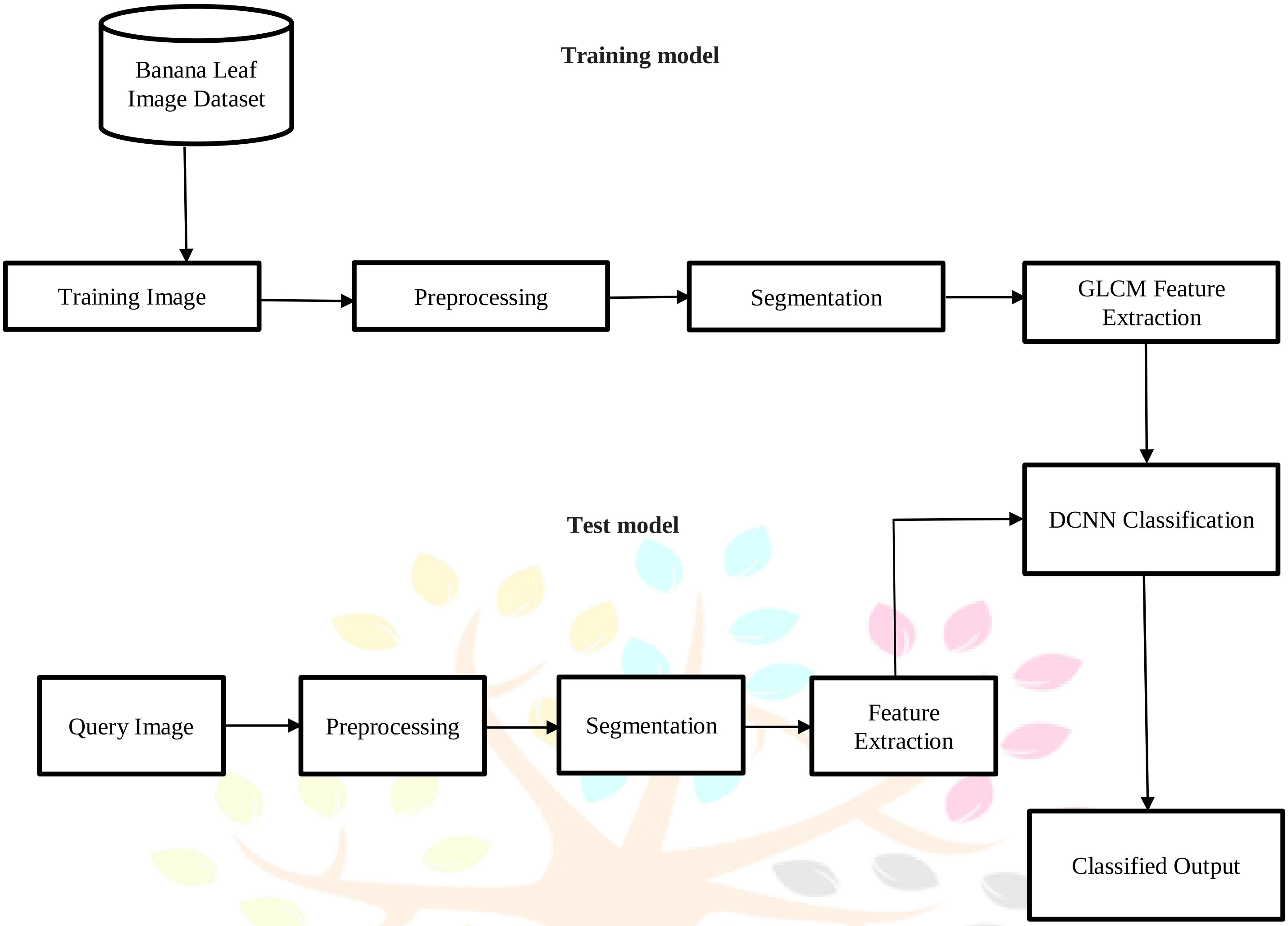

Figure-1 Proposed Architecture

Figure 1 above depicts the suggested architecture. The picture has first undergone pre-processing for data cleansing, noise reduction, and image scaling. After the picture has been segmented, its features are retrieved using GLCM, and DCNN is used to classify the extracted features. The portion of the banana leaf with the disease detected will be the categorized output

A. Feature Extraction using Gray level co-occurrence matrix (GLCM):

One of the most popular methods for analyzing textures is GLCM, which is used to assess an image's properties once it has been associated using order 2 statistics. The assessment of the relationship between the pixels in the spatial domain has been done by matrix design. Thus, the foundation of this technique is the texture data that was extracted from the result of a different correlation. Based on the pair of occurrence numbers for gray levels, i and j, each input of GLCM is (i, j), which is divided by the d distance of the input image. The comparison calculation between the different GLCM has been carried out by the 22 features, and different numbers of features have been employed in previous works. This suggested simulation technique distinguishes this simulates a spatial comparison between the image's pixels and compares the pixels that are available in the input image based on their direction and distance, which are denoted by Θ and d, respectively. All of the images have been extracted, quantized, and stored as 16 grey levels and 4 GLCMs (M) for \ = 0, 45, 90, and 135 degrees, where value = 1. Five features will be derived from each GLCM. Therefore, twenty characteristics have been extracted for each image. The classifiers process the feature normalization, which spans from 0 to 1, and each classifier receives a similar pair of features. Texture features using Co-occurrence matrix representation are the features used in this inquiry. GLCM is the genuine second-arrangement . Initially, the image is transformed into l-grey-level, and the number of intensity pairings found between the neighbor and current pixels for each scaling and orientational feature generates the GLCM. The average of the matrices for each scaling and orientational feature is used to create the feature vector. To compute standardized GLCM, use the formula:

$$G(i,j) = \frac{N(i,j)}{\sum_{m=0}^{l-1}\sum_{n=0}^{l-1} N(m,n)}$$

(8)

In this case, the grey values in the l-gray image are represented by i and j. **N(i,j)** indicates the relative recurrence frequency matrix of co-occurrence by:

$$\mathbf{N(i,j)} = \mathbf{num}(\{[(x_1,y_1),(x_2,y_2)]\}|x_2 - x_1 = d\cos\theta, y_2 - y_1 = d\sin\theta, I(x_1,y_1) = i, I(x_2,y_2) = j)$$

(9)

In this case, pixel positions are shown by (x_1, y_1) and (x_2, y_2) and the grey level of the pixel is shown by $\mathbf{I}(\cdot)$. Num indicates the number of pixel matches that meet the comparison's requirements.

B. Classification using Deep convolution neural network (DCNN):

Following the extraction of the feature vector from the input image, the image is represented as a fixed length vector that needs to be classified using a classifier. A typical CNN typically consists of the following layers: input, convolution, activation, pool, completely connected, and output. The CNN pool structure continuously performs decoding, deducing, convergent processing, and mapping the signal features of input information to the feature space of the hidden layer. The layers of CNN establish the relationships among various nodes, and the input information is forwarded through various layers sequentially. Classification is carried out by the fully connected layer, and outputs are provided in relation to the retrieved characteristics. One important analytical mathematical operation is deep convolution. When this operator applies two functions, f and g, it creates a third function, which is the overlapping region between the two functions that is either translated or flipped. The calculation for this third function is as follows:

$$\mathbf{z}(\boldsymbol{t})^{def} = \mathbf{f}(\boldsymbol{t}) * \mathbf{g}(\boldsymbol{t}) = \sum_{\tau=-\infty}^{+\infty} \boldsymbol{f}(\boldsymbol{\tau})\boldsymbol{g}(\boldsymbol{t}-\boldsymbol{\tau})$$

(10)

The above equation's integral form can be found using,

$$\mathbf{z}(\boldsymbol{t}) = \boldsymbol{f}(\boldsymbol{t}) * \boldsymbol{g}(\boldsymbol{t}) = \int_{-\infty}^{+\infty} \boldsymbol{f}(\boldsymbol{\tau})\boldsymbol{g}(\boldsymbol{t}-\boldsymbol{\tau})\boldsymbol{d\tau} = \int_{-\infty}^{+\infty} \boldsymbol{f}(\boldsymbol{t}-\boldsymbol{\tau})\boldsymbol{g}(\boldsymbol{\tau})\boldsymbol{d\tau}$$

(11)

A digital image is classified as a discrete function f (x, y) of a two-dimensional (2D) space. Taking into account g (x, y), a 2D convolutional function, yields the output picture z (x, y) as follows:
Figure 2 depicts the DCNN architecture.

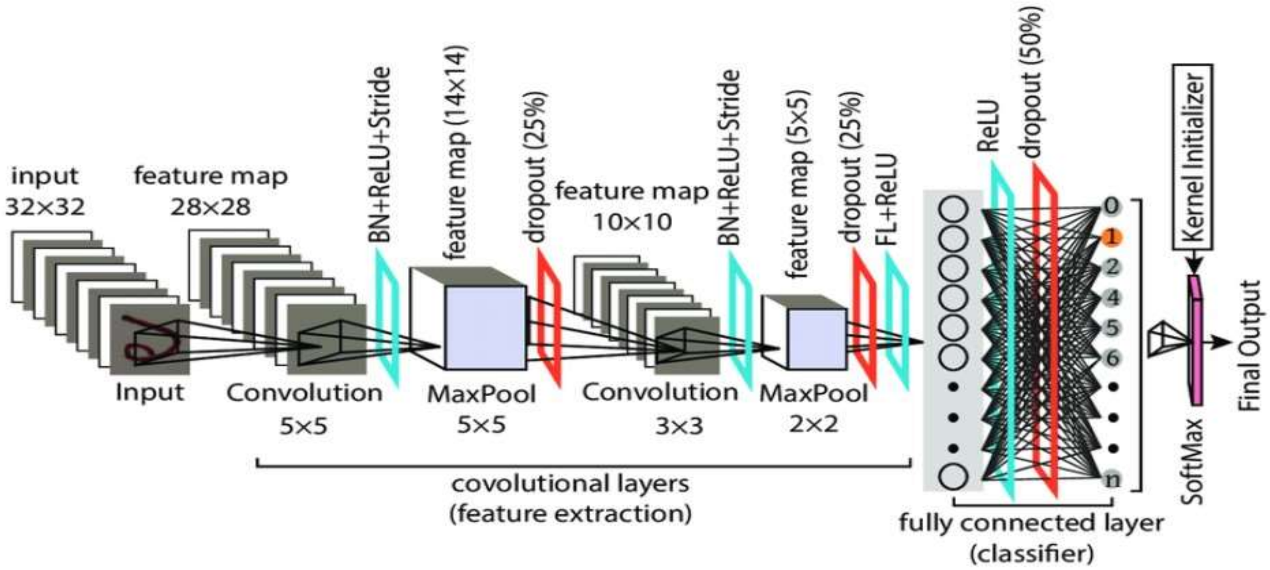

Figure 2-DCNN architecture

$$\mathbf{z}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) * \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y})$$

(12)

In this case, the picture features are extracted using the convolutional process. Similar to this, in applications utilizing deep learning techniques, a color image provided as input becomes a high-dimensional array with dimensions of 3 × image width × image length; consequently, the CNN convolutional kernel is referred to as accounting in the deep learning algorithm. Additionally, the computing parameter is a high-dimensional array. The corresponding convolutional operation for the supplied 2D image is thus provided by:

$$\mathbf{z}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) * \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) = \sum_{t}\sum_{h} \boldsymbol{f}(\boldsymbol{t},\boldsymbol{h})\boldsymbol{g}(\boldsymbol{x}-\boldsymbol{t},\boldsymbol{y}-\boldsymbol{h})$$

(13)

The following is the integral form:

$$\mathbf{z}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) * \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) = \int\int \boldsymbol{f}(\boldsymbol{t},\boldsymbol{h})\,\boldsymbol{g}(\boldsymbol{x}-\boldsymbol{t},\boldsymbol{y}-\boldsymbol{h})\boldsymbol{dtdh}$$

(14)

Regarding the specified m × n convolution kernel,

$$\mathbf{z}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) * \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) = \sum_{t=0}^{t=m}\sum_{h=0}^{h=n} \boldsymbol{f}(\boldsymbol{t},\boldsymbol{h})\boldsymbol{g}(\boldsymbol{x}-\boldsymbol{t},\boldsymbol{y}-\boldsymbol{h}) \quad \textbf{(15)}$$

In this case, f denotes the input G with m and n convolution kernel sizes. on most cases, convolution is implemented on a computer as a matrix product. Assume that the convolution kernel size is n × n and the picture size is M × M. In computing, the convolution kernel multiplies each image region by the image of size n × n, resulting in an image region of n × n that represents the length of the column vector. When zero-zero padding is carried out with step 1, all (M − n + 1) ∗(M − n + 1) results are feasible; the real image can be described as the matrix [n∗n∗(M − n + 1)] when the small parts of the image are represented as n x n column vectors. Taking into account K as the convolution kernel count, the output picture after convolution is k∗(M − n + 1) ∗(M − n + 1); that is, the number of convolution kernels × image width × image length. In the suggested model, every trained parameter has a random initialization value between -0.05 and 0.05. The two steps of the training phase are called forward and reverse propagation. The forward propagation's goal is to approximate the output of the input data classification process using the current parameters. The goal of back propagation is to minimize the discrepancy between the intended and actual categorization output by updating the parameters.

C. Forward Propagation:

The (L+1) layer DCCN model that is shown here has $n1$ input units in the input layer, $n5$ output units in the output layer, and a large number of hidden units in the C2, M3, and F4 layers. Where L=4. Taking xi as both the input and the output of the $(l-1)^{th}$layer, we can compute xi+1 as

$$\boldsymbol{x}_{i+1} = \boldsymbol{f}_i(\boldsymbol{u}_i) \quad \textbf{(16)}$$

Where

$$\boldsymbol{u}_i=\boldsymbol{W}_i^T\boldsymbol{x}_i + \boldsymbol{b}_i \quad \textbf{(17)}$$

and WTistands for the weight matrix on the input, b$i$for an additive bias vector, and (·) for the $i$thlayer's activation function. The hyperbolic tangent function, tanh(u), is selected as the activation function for the C1 and F3 layers in this instance. max(u), the M2 layer's maximum function, is implicated. Given that the DCNN classifier in question is of the multiclass variety, the output of the F3 layer is passed to the $n5$ softmax function, which distributes the data over the $n5$ class labels. One way to characterize softmax regression is as

$$\mathbf{y}=\frac{1}{\sum_{k=1}^{n} e^{w_{L,K}^T x_L + b_{L,k}}}\begin{bmatrix} e^{w_{L,K}^T x_L + b_{L,1}} \\ e^{w_{L,K}^T x_L + b_{L,2}} \\ e^{w_{L,K}^T x_L + b_{L,n}} \end{bmatrix} \quad \textbf{(18)}$$

The final probability of each class in the current iteration is represented by the output vector y = xL+1 of the output layer.

D. Back Propagation:

The training parameters are updated in this stage using a gradient descent approach with a minimal cost function and an estimated partial derivative for each parameter used in the training process. The loss function can be found as

$$\mathbf{J}(\boldsymbol{\theta}) = -\mathbf{1}/\boldsymbol{m}\sum_{i=1}^{m}\sum_{j=1}^{n_5} \mathbf{1}\{\boldsymbol{j} = \boldsymbol{Y}^{(i)}\}\,\mathbf{log}\,\boldsymbol{y}_j^{(i)} \quad \textbf{(19)}$$

where the vector's size is $n5$, and the variables m, Y, and y(i) j represent the quantity of training samples, estimated output, and j^{th}value of the original output for the i^{th}training sample. The probability for the label class in the estimated output (i) for sample i is 1, and the probability for the other classes is 0. If j is the estimated label of training sample i, then j = Y(i) = 1; if not, then j = 0. Before $\boldsymbol{\eta}$, a minus sign is appended for ease of calculating. The loss function associated with u$i$is obtained as

$$\boldsymbol{\delta}_i = \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{u}_i}\begin{cases} -(\boldsymbol{Y}-\boldsymbol{y}).\acute{\boldsymbol{f}}(\boldsymbol{u}_i), \boldsymbol{i} = \boldsymbol{L} \\ \left(\boldsymbol{W}_i^T\boldsymbol{\delta}_{i+1}\right).\acute{\boldsymbol{f}}(\boldsymbol{u}_i), \boldsymbol{i} < \boldsymbol{L} \end{cases} \quad \textbf{(20)}$$

Therefore, updating is carried out on each iteration by

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \boldsymbol{\alpha}.\nabla_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\theta}) \tag{21}$$

in order to modify these parameters, where α denotes the learning factor (αtaking 0.01) and

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\theta}) = \left\{ \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{\theta}_1}, \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{\theta}_2} \dots \dots, \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{\theta}_L} \right\} \tag{22}$$

The cost function becomes less as the number of training iterations rises, suggesting that the initial output was quite close to the estimated result. If their difference becomes smaller, the iteration ends.

IV. PERFORMANCE ANALYSIS

The simulation findings for banana leaf disease detection from the input dataset are covered in this part. Below, a number of outputs have been discussed: To begin with, a dataset folder containing several photos of banana leaves was created for the purpose of testing. Higher accuracy has been achieved in identifying the indication of leaf disease, which has also been appropriately trained and gathered into the dataset. Below is the parametric analysis. Figures 4,5, 6, 7, and 8 below show how the suggested technique and the current technique compare in terms of accuracy, precision, recall, F1-score, true positive rate, and false positive rate.

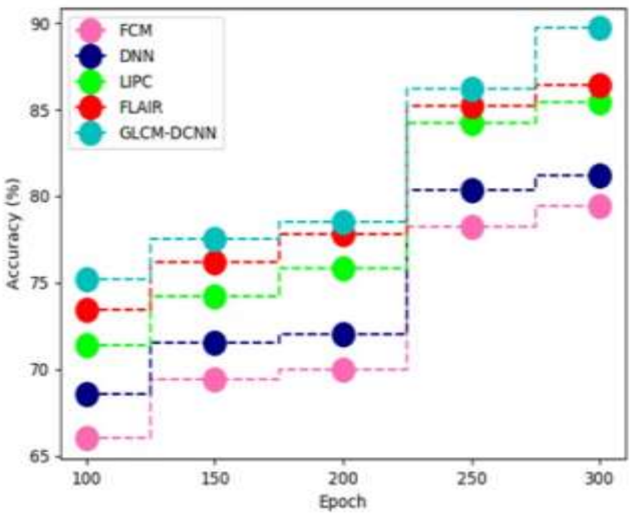

Figure-4 Comparison of accuracy

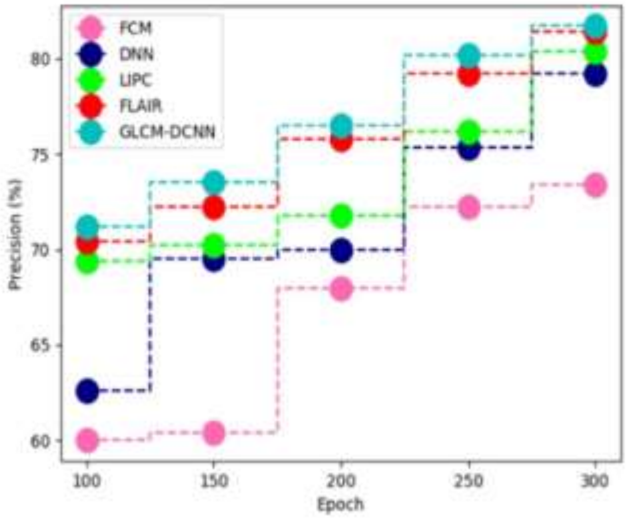

Figure-5 Comparison of Precision

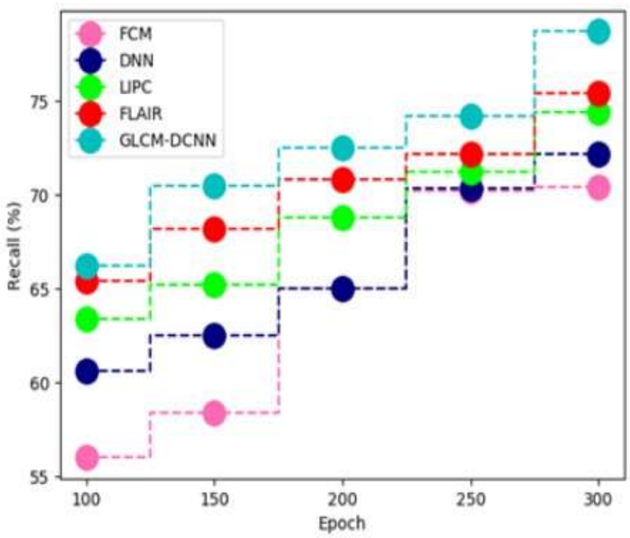

Figure-6 Comparison of Recall

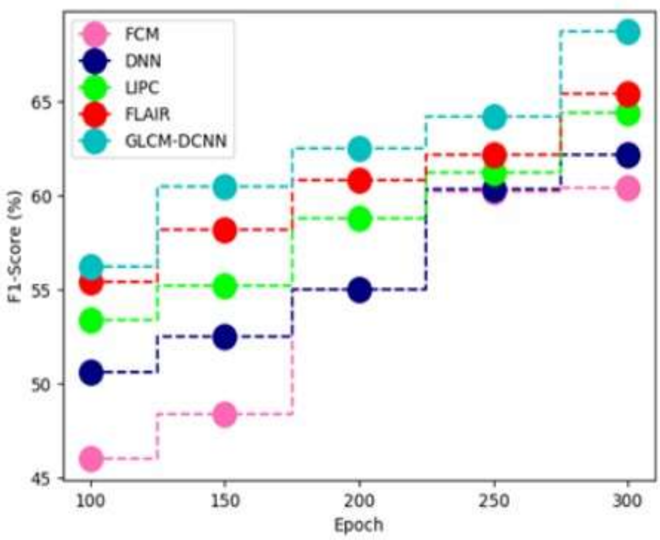

Figure-7 Comparison of F-1 Score

Figure 8 below compares the parametric measurements that were determined for the suggested method.

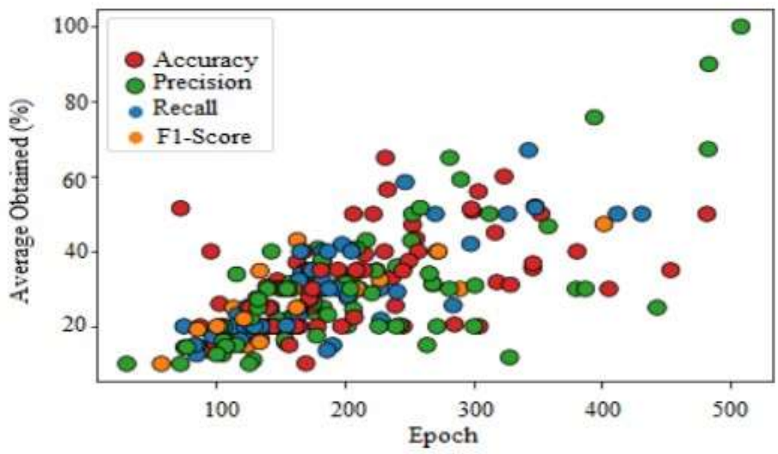

Figure-8 Overall Comparison of Proposed Technique

V. CONCLUSION

Using image processing techniques to recognize and classify photos of banana plants can improve the accuracy of disease diagnosis. In this case, a Python application was used to simulate the classification and feature extraction processes. When it comes to identifying and categorizing banana plant diseases, manual systems have been used in place of more accurate and time-consuming ones when compared to the suggested technique. Farmers have employed this suggested method to detect diseases earlier and more accurately, preventing them from spreading to neighboring plants. Next, the suggested system provides the best possible classification using DCNN and feature extraction using GLCM, which may result in a higher yield. As a result, two banana leaves have been found, and further research can be done to detect different diseases in the banana plant's fruit and stem.

VI. REFERENCE

[1] Amara, Jihen, Bassem Bouaziz, and Alsayed Algergawy. "A deep learning-based approach for banana leaf diseases classification." Datenbanksysteme für Business, Technologie und Web (BTW 2017)-Workshopband (2017).

[2] Basavaraj Tigadi, Bhavana Sharma. "Banana Plant Diseases Detection and Grading Using Image Processing." International Journal of Engineering Science and Computing, Volume 2, Issue 4, June 2016.

[3] Ferentinos, Konstantinos P. "Deep learning models for plant disease detection and diagnosis." Computers and Electronics in Agriculture 145 (2018): 311–318.

[4] Grinblat, G.L., Uzal, L.C., Larese, M.G., Granitto, P.M. "Deep learning for plant identification using vein morphological patterns." Computers and Electronics in Agriculture 127 (2016): 418–424.

[5] Gurleen Kaur Sandhu. "Plant Diseases Detection Techniques: A Review." International Conference on Automation, Computational and Technology Management (ICACTA), 2019.

[6] Karthik.G, Praburam.N. "Detection And Prevention Of Banana Leaf Diseases From Banana Plant Using Embedded Linux Board." Online International Conference on Green Engineering and Technologies (IC-GET), 2016.

[7] Lee, S.H., Chan, C.S., Wilkin, P., Remagnino, P. "Deep-plant: Plant identification with convolutional neural networks." 2015 IEEE Intl Conf. on Image Processing, pp. 452–456.

[8] Mohanty, S.P., Hughes, D.P., Salathé, M. "Using deep learning for image-based plant disease detection." Frontiers in Plant Science 7 (2016): 1419.

[9] Pawara, P., Okafor, E., Surinta, O., Schomaker, L., Wiering, M. "Comparing local descriptors and bags of visual words to deep convolutional neural networks for plant recognition." 6th Intl Conf. on Pattern Recognition Applications and Methods (ICPRAM 2017).

[10] Sagar Patil. "A Survey on Methods of Plant Diseases Detection." International Journal of Science and Research (IJSR), 2013.

[11] Sandip P. Bhamare, Samadhan C. Kulkarni. "Detection of Black Sigtoka on Banana Tree Using Image Processing Techniques." International Journal of Electronics and Communication Engineering (IJECE).

[12] Seetharaman, K., Mahendran, T. "Detection of Disease in Banana Fruit using Gabor Based Binary Patterns with Convolution Recurrent Neural Network." Turkish Online Journal of Qualitative Inquiry, Vol. 12(9), 2021, pp. 6958–6966.

[13] Seetharaman, K., Mahendran, T. "Leaf Disease Detection in Banana Plant using Gabor Extraction and Region-Based Convolution Neural Network (RCNN)." Journal of the Institution of Engineers (India): Series A, 2022. DOI: https://doi.org/10.1007/s40030-022-00628-2.

[14] Singh, Vijai, and Ak K. Misra. "Detection of plant leaf diseases using image segmentation and soft computing techniques." Information Processing in Agriculture 4.1 (2017): 41-49.

[15] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D. "Deep neural networks based recognition of plant diseases by leaf image classification." Computational Intelligence in Neuroscience (2016): Article ID 3289801.

[16] Surya Prabha, J. Satheesh Kumar. "Study on Banana Leaf Diseases Identification Using Image Processing Methods." International Journal of Research in Computer Science and Information Technology (IJRCSIT), Volume 2, Issue 2(A), March 2014.