International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

GNN Based Cauliflower Plant Disease Prediction Using Deep Learning Techniques

Meenalochini M [1], P Amudha*[2]

Submitted: 19/01/2024 **Revised**: 28/02/2024 **Accepted**: 05/03/2024

Abstract: A vital component of agriculture, vegetables are essential for maintaining people's overall health. The information systems can assist vegetable producers in producing high yields that support sustainable farming practices and help ensure global food security. The common vegetable cauliflower (Brassica oleracea var. botrytis) is susceptible to several illnesses that can reduce output and performance. Furthermore, deep learning-based disease identification systems that can assist farmers in identifying cauliflower infections and enabling them to respond promptly have yet to be created for the crop. This study suggests an automated machine vision-based expertise method for identifying cauliflower illnesses. A Graphical Neural Network based Plant Disease Prediction (GNN-PDP) model is designed in this research using the Internet of Things (IoT) and deep learning algorithms. A cell phone or other IoT portable device's taken picture is analysed and then classed to detect disease to help cauliflower growers. Depending on feature extraction, the algorithm categorizes four illnesses in cauliflowers, including bacterial softness, white rusting, black rotting, and downy mildew. This study makes use of 750 photos in total. Before extracting two attributes, including statistical and co-occurrence variables, the Cat Swarm Optimization (CSO) technique was performed on the collected pictures to segment the disease-affected areas. Six classification algorithms— CNN, DNN, RF, DT, LDA, and PCA —were utilized for illness classification comparison. Their performance was assessed using different efficiency measures. With an efficiency close to 89%, it was discovered that the GNN classification performed better than all other classifications for identifying cauliflower illness.

Keywords: *Graph Neural Network, Plant Disease Prediction, Deep Learning, Internet of Things*

1. Introduction

For rural residents, particularly those in developing nations, the agricultural sector is crucial in supplying them with food, income, and employment. This industry contributes 6.5% to global economic output [1]. A 2020 poll found that 68% of individuals earn their livelihood through agriculture. Additionally, agriculture is the foundation of several marketplaces for agricultural commodities that customers need, particularly in rural regions. To provide steady food security for humans, it is crucial to establish a worthwhile and practical agrarian economy. Good agricultural production system management is required [2]. Most farmers in rural regions need more expertise to identify and quickly address issues with their budding crops [3]. Due to damage brought on by numerous illnesses, they frequently fail to produce the appropriate yield of their planted crops. By promptly and precisely identifying conditions in the farmer's grounds, the output of any plant, such as vegetables, might be considerably boosted. The ability to control any infections and assure improved outcomes depends on the early, precise diagnosis of the disease [4].

Cauliflower is a crop that farmers mostly grow on farmland. It belongs to the family Brassicaceae of vegetable crops [5]. Both fiber and B vitamins are abundant. It has phytonutrients that lower the chance of developing cancer. It includes choline, an important ingredient for supporting sleep, muscular movement, training, and cognition, and it is a cruciferous vegetable that provides fiber to reduce the risk of cardiovascular issues. Around the world, cauliflower is grown in various nations, including China, Ireland, France, Brazil, and Pakistan [6]. About 2.5 lac acres and 7.8 million tons of cauliflower are produced annually in India. Bangladesh produces 73K tons of cauliflower annually, to the extent of roughly 9,400 ha. Several diseases, including bacterial soft rotting, black legs, black rot, downy fungus, powdered mildew, ring mark, white rusty, etc., affect cauliflower during growth. These diseases have a substantial impact on cauliflower development and production. To maximize the productivity and profitability of cauliflower production, farmers must apply the proper control method and identify a particular disease early on [7]. Creating an automated process for recognizing cauliflower illness will substantially assist farmers in early disease detection and guarantee a higher production of cauliflower. Consequently, an expert system must be created to classify cauliflower disorders. This work used a machine vision technique to investigate the detection of cauliflower illness [8]. The architecture is an IoT-based, deep learning-based platform that can take pictures with cell phones or other portable devices and use them as inputs. Then it identifies four cauliflower disorders and gives consumers immediate feedback [9]. A GNN technique was used on diseased photos to separate the damaged regions from recognizing the cauliflower illness [10]. A collection of characteristics were retrieved using image analysis methods based on the outcomes. Then, six well-known classification methods were utilized for evaluation and training to identify the disorders. The outcome criteria were used to compare the effectiveness of these systems. The study's primary significance is as follows:

- The primary goal is to identify cauliflower illnesses automatically using IoT devices.

[1] *Department of Computer Science and Design, Kongu Engineering College Perundurai, Erode, Taminadu, India*
[2] *Department of Computer Science and Engineering, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, Tamilnadu, India*
* *Corresponding Author Email: amudha_cse@avinuty.ac.in*

- They are methodically arranging the top solid features for the classifications' training and evaluation to categorize the illnesses affecting cauliflower using a deep learning model.
- The simulation outcomes show how the suggested technique significantly affects the picture data of cauliflower illnesses using Graphical Neural Networks (GNN) and IoT.

1.1 Background to the Plant Disease Detection and Prediction Models

Illness categorization and disease identification are the two components of the problem of machine vision-based disease or defect identification. At the same time, another study has focused on illness categorization or disease identification. Apple, pineapple, tomatoes, potatoes, and other fruits and veggies have all been the subject of efforts for the automated identification of various illnesses.

The advent of several crop-related illnesses impacts the efficiency of the agriculture industry. Plant disease diagnostics is important in addressing this problem, educating farmers about how to stop the spread of diseases in their crops, and putting appropriate management in place. Academics have already utilized numerous methods for this aim, but other methods involving eyesight still need to be investigated. Support vector machines, k-means grouping, radial basis values, genetic algorithms, image analysis methods like filtration and classification, and deep organized learning methods like convolutional neural networks are examples of frequently used approaches. It has developed a hybrid method for agricultural leaf disease identification using auto-encoders and convolutional neural systems [11]. This study presents a unique way of identifying plant disease using leaf pictures and convolutional encoder systems.

Machine learning and image analysis are now Information Technology (IT) sector keywords, and their implementations in the agricultural sector can improve decision-making in various areas. One of the most labor-intensive crops is rice, particularly in South Asia. This study suggests a Convolutional Neural Network (CNN) based predictive system for disease categorization and forecasting in rice crops. If rice crop illnesses are not treated early on, they can be extremely lethal and hurt the harvests. The suggested approach would enhance decision-making utilizing CNN in the occurrence of various illnesses in the rice crop for early disease detection and massive loss in production of the total yield avoidance [12].

The development and production of the rice plant are greatly influenced by nitrogen (N). The CNN-based method for predicting rice nitrogen deficit is proposed in this research [13]. By substituting the final output level of CNN with a superior classifier like the Support Vector Model (SVM), the pre-trained CNN design is updated to increase classification performance. To select the best categorization model, quantitative evaluation is done after considering the outcomes of 100 separate simulations. ResNet-50+SVM outperformed the other five CNN-based categorization systems with an efficiency of 81%, according to the scientific analysis.

Through IoT and deep learning integration, this study seeks to construct an "Automated and Intelligence Data Collection and Classification" system [14]. The platform continuously gathers parametric information and pictures from the field in Mysore, India, which grows millets. It immediately transmits the collected data to the Microprocessor and the cloud service. The "Custom-Net" architecture developed for this study is implemented on the web server. It works with the Raspberry Pi to accurately forecast pearl millet's blasting and rust illnesses.

Additionally, the system displays the characteristics that the "Custom-Net" retrieved. Despite being on par with cutting-edge models in categorization, "Custom-Net" successfully cut learning time by 62%. The algorithm is now more suited to automate illness identification. This demonstrates that the suggested methodology successfully gives farmers a cheap and practical instrument to increase crop output and the quality of products.

The research uses data-mining algorithms on complex spatial arrays of remote sensing information while considering topography, bioclimatic, and soil variables. Relying on ensemble training methods and spatially recurrent neural systems, it offers a novel crop disease forecasting system for agriculture emergency management. It is a time- and space-based extension of the Recurring Neural Networks (RNN) [15]. Practical tests are carried out on a dataset based on incidents of wheat yellow rust epidemics in Longnan city. According to empirical findings, the suggested approach performs better at predicting crop symptom severity than every baseline technique.

Academics are always working to create an effective computerized disease categorization system to identify agricultural diseases. This article describes making an automated process that will instruct farmers by analyzing photos of unhealthy, infected rice plants [16]. The automatic identification and categorization of rice illnesses is the primary objective of the system's design. The two primary divisions of the effort are identifying rice illnesses and categorizing crop diseases. In the disease identification work, the disease-related characteristics were first obtained from the rice pictures' diseased region using various extracting features approaches. Then key and pertinent aspects are chosen from the retrieved elements using the suggested methodology.

This paper presents three distinct CNN designs that integrate crop characteristics and other contextual non-image meta-data onto an image-based CNN [17]. By doing this, the difficulty of the disease categorization problems is made simpler while still allowing for simultaneous training from the complete multi-crop database. The crop-conditional plant's disease categorization network achieves a balanced efficiency by upgrading the prior approaches and eradicating the miss-categorizations of the earlier methods by concatenating the contextual data at the embedded vector layer.

Scaling up automatic crop disease identification technologies is challenging due to bias in stratified sampling. The unpredictability as a misclassification indicator and cutting-edge Probabilistic deep learning algorithms are used in this research to construct a stochastic programming strategy for crop disease diagnosis [18]. The findings demonstrate that Bayesian inference yields categorization accuracy equivalent to the common optimization techniques for optimizing deep learning networks. The suggested approach quantifies the ambiguity of the estimates for out-of-sample events while approximating the posterior probability for the crop disease identification issue.

The major goal of this study is to put into practice a unique deep-learning plant disease categorization system [19]. The benchmark database is utilized throughout the data-collecting stage and has undergone median filtration and contrasting enhancement procedures. After preprocessing the picture, the Optimized Fuzzy C-Means (FCM) Grouping identifies abnormality from the leaves. The classification of the many types of crop diseases was done most effectively using the revised Long Short-Term Memory (LSTM) with enhancement. The proposed methodology achieves the highest categorization performance when measured against more conventional methods.

The fuzzy inference method is the primary decision-making engine behind an innovative strategy for detecting agricultural diseases that operate on Android smartphone devices [20]. The device can speak with Pakistani producers in their native Urdu and help them identify illnesses in their crops. It might be equally useful to government agricultural professionals in detecting and treating plant disease. It uses its inferential engine to provide an outcome in the shape of a detected condition from input plant indicators and an ambiguous input. The suggested approach can identify the major illnesses of Pakistan's two major crops, cotton.

Razieh Pourdarbani et al. (2023) illustrated the cauliflower has a variety of plant pathology illnesses; nevertheless, the traditional approach for detecting and isolating sick cabbages is laborious and time-consuming; hence, cauliflower can be recognized in the smallest amount of time utilizing imaging methods and Artificial Neural Networks (ANN) [21]. Cauliflower Sis an extremely nutritious vegetable that is naturally high in fiber and B vitamins. The prediction includes fiber that helps in weight reduction and digestion, choline that is necessary for learning and memory, and a variety of other minerals. Cauliflower development is hampered through variables such as a lack of nutrients, climatic conditions, and illnesses.

K P Asha Rani et al. (2023) Deep Transfer Learning (DTL) can be used to tackle these issues through recycling existing knowledge representations. To people take into account photos of sunflower and cauliflower leaves, bulbs, and flowers taken in their natural settings, and the Agri-ImageNet collection. The limitation of the plant village dataset, in that photos are recorded under standardized, laboratory conditions, is addressed. With the aim to determine which deep transfer learning model is most effective for the plant disease dataset, that research sets intended to investigate and assess all of the available methods. Agriculture, being the backbone of the national economy, has far-reaching effects on daily life. Monitoring plant health and identifying plant diseases are crucial for environmentally responsible farming.

Mónica Pineda et al. (2023) detailed the use of Computer Vision (CV) for automatic illness identification is gaining popularity. Cauliflower accounts for a significant proportion of both the land dedicated to winter crops and the quantity harvested in Bangladesh. Cauliflower's production, quantity, and quality can all suffer with the right kind of care, nevertheless, keeping an eye for the plant diseases manually is an impossible task that takes a lot of time and effort. Through helping farmers in Bangladesh take the right action resulting from automated illness identification that suggested method can boost cauliflower output and improve crop yields. Obesity, diabetes, and cardiovascular disease are all reduced as a result, among other health advantages.

Ch Lavanya Ratna et al. (2023) explicate the uses of several models based on frame variances, adaptive contextual removal, adaptive frame modifications, and the Gaussian mixture model [27]. Sulastri Sabudin, Muhammad Eric Zulkarnaen (2022) conducted the study to inspect the temperature distribution in a container-type plant factory by computational fluid dynamics (CFD) [28]. A three-dimensional computational fluid dynamics (CFD) model was industrialized to simulate the container-type plant factory with air habituation and a deplete fan providing cooling air and freshening across the plant factory. Amgad Muneer. (2023) implies the backpropagation process when the recurrent network weights tend to diminish and hinder the network and also aims to provide an experimental investigation of LSTM networks with an importance on inadequacy in long-term dependencies convergence because of VGP [29]. Ahmad Shahir Jamaludin, Wan Sharuzi Wan Harun, Kumaran Kadirgama, & Devarajan Ramasamy. (2023) a water quality monitoring system [30] utilizing temperature, pH, and ESP32, with the parameter values shown on the Blynk platform was established. The system successfully measures real-time water conditions and transmits the data to the Blynk application.

The literature survey shows the available models for plant disease detection and classification. The existing models fail to produce higher accuracy in variety because of the need for prediction models. The proposed model with deep learning and GNN combined with IoT has better results.

2. Methodology

2.1 Construction of Graph Neural Network

The system inputs the labelled enclosed substrings into the graphing neural networks for forecasting -disease connections after separating them and marking the nodes in each enclosed sub-graph. It employs the disease correlations in bipartite graph networking, although most currently suggested graph neural systems are suited to heterogeneous networks. It enhances the deep graphical neuronal networks to achieve higher efficiency in heterogeneous systems.

2.1.1. Convolution levels in a graph

Learning the node descriptions is the function of the graph convolution level. The vector description of each node in a diagram is retrieved using multi layered convolution as an input to the graphing convolution level. Local substructural characteristics of the graph are contained in the vectors. Gathering feature data from each node's neighbors are known as graph augmentation.

The data vector of a graph, assuming its equivalence vector $A \in R^{m \times m}$, is given by$X \in R^{m \times d}$, where m denotes the number of nodes and d is the size of the attributes. Equation (1) shows the representation of graph convolutions.

$$Z = \propto (f(A) \times X \times W) \qquad (1)$$

Where$W \in R^{d \times l}$ denotes the variables that need to be trained to transform the d-dimensional information into l-dimensional data and $f(A)$stands for the adjacency vector A's propagating value. Additionally,$(f(A) \times X \times W)$specifies that the information is converted after the function matrix of each node is collected in the method of propagating functions. The activating variable is also $\propto (i)$. GNN is solely linked to illnesses in the bipartite unguided system. One illness node can only touch another condition after a two-step leap; the same is true for GNN circuits. As a result, it employs a propagating value for second-order topographical data that allows direct aggregation of data across homogeneity nodes, i.e., by establishing a propagating value $f(A) = A^2$.

The topology properties of the graph learned by the graph convolutional system vary somewhat depending on the propagating function chosen. The graphs' convolutional layers learn improved graph topological properties by splicing the neighbour data of nodes gathered by various propagating algorithms. It depicts the graphs convolution layers of the GNN-PDP model, and Equation (2) is used to describe it.

$$C^{p+1} = \propto (A \times C^p, A^2 \times C^p, D^{-1} \times A, D^{-1}AD^{-1}C^p) \qquad (2)$$

The p-th layer graphing convolution's outcome is represented by $C^0 \; and \; C^p$, its output dimensions is d_p, and the spliced of row matrices, which joins node matrices produced using various propagating rates, is represented by [.]. The adjacent vector is denoted A, and the convolutional outcome is marked C. After splicing, it is possible to consider all the topological properties that these various propagating functions have collected. The variable that has to be learned is W^{4d_p}, which converts the characteristics of the spliced nodes from the $4d_p$ dimensions to the

d_{p+1}dimensions. Since four propagating variables are engaged in the graph convolution procedure, dimensions $4d_p$ are employed here.

2.1.2 Pooling graph layers

A graph convolutional layer is employed to learn a dormant vector depiction for each node. In this case, the graph pooling level of the GNN-PDP chooses the k most crucial nodes from among the graph's vertices to describe the chart. Using the output of the graph convolution layers, the significance of the node is assessed. The result of the preceding layer is mapped to one dimension in the last level of graph convolution. In other words, measurements$4d_{p-1}$ is mapped to sizes 1 by the variable W^p of the final layer of graph convolutions. This method yields a number for each component, representing the node's significance inside the graphs.

According to Z's final size, it arranges Z in descending rank. If two nodes appear to be equivalent in the last Z-dimension, it evaluates their subsequent dimensions until it can tell them apart. The graph max pooling outputs the top k elements in the rankings, assisting the next layers of a traditional neural network to produce a vector with a given specification. The zero matrices are inserted after the sorting nodes when the amount of elements, n, is below the threshold value, k.

2.1.3 Fully linked layers and convolutional layers

A tensor Z is produced after the graph pooling level. The subsequent layers progressively improve the graph representations' characteristics using the conventional one-dimensional convolutional neural networks in conjunction with the top pooling layer. Then it does the final predictions using a completely connected layer.

Tensor Z must first be transformed into a one-dimensional matrix for input for one-dimensional convolutional neural networks. The initial one-dimensional fully convolutional network's filter dimensions and step distance are both set to $(d_p + d_{p+1} + \cdots + d_{p+n})$ to implement the filtering for each node attribute.i.e., each node feature is combined separately first. A one-dimensional convolutional neuronal layer is then utilized for training the graph further to capture the local characteristics in the sequential parts following a top pooling layer. It is linked to the completely connected level at the end. The projected values of all forecasted samples under the proper labeling are added together using the error function. The predicted value, in this case, maps the forecast range from (0, 1) to (0, +∞) and is a negative integer in the exponential form of a standardized exponential expression (softmax). If every predictive sample is properly forecasted, the error function is nearer to 0. The likelihood of linkages between disease node pairs and GNN is ultimately output by the completely connected layers.

3. Proposed model

3.1 Proposed model for plant disease prediction

The suggested system's step-by-step operating process has been detailed to clarify the technique. This method acquires the picture, downscales it to a predetermined size, uses histogram normalization to intensify the contrasts, and changes the image's color space from Red, Green, and Blue (RGB) to L*x*y, where length, width, and height are denoted L, x *and* y. Following preprocessing, this method uses the k-means classification method to divide the pictures, extracts their statistics and co-occurrence characteristics, and then trains them with six well-known classifications to produce the final outputs. The next subsections describe the specific explanations of the steps above.

3.1.1 Acquisition of image

The preprocessing step in this method begins with many photos of cauliflowers. Seven hundred and sixty color pictures of cauliflower were amassed (disease-affected or disease-free). One hundred and six of them are from the Website, while the other six hundred and sixty photographs were gathered locally.

3.1.2 Preprocessing

The term "image preprocessing" describes operations on pictures at their most basic. If entropy is a measure of data, then these methods decrease the data in the images. On the other hand, preprocessing seeks to improve the image data by reducing undesirable distortions or increasing crucial visual qualities for further processing and assessment. The input data is preprocessed in this study using a histogram equalizer method and contrast improvement.

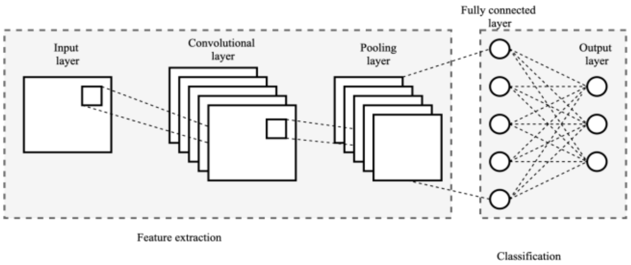

Fig. 1. Architecture of the GNN

The architecture of the GNN is shown in Fig. 1. It has input layer, convolutional layer, pooling layer, fully connected layer and output layer. All the layers consists of graph and follows graphing theory. The feature extraction and classification model enhances the overall accuracy.

3.1.3 Resizing image

The picture is converted into a predetermined size image using exponential smoothing. Cauliflower photos are reduced in size to 300x300 pixels since extraction of features differs depending on image dimension. Assume that "I" is the brightness and that p_x, p_y *and* $p_{xy}$are the variants. These numbers are used for identifying a unit square's corners, which are (1, 1), (1, 0), (0, 1), and (0, 0). The approximation surface's brightness can be expressed in Equation (3).

$$f(i,j) = \sum_{x=0}^{N} \sum_{y=0}^{N} m_{xy} \times i^x \times j^y \qquad (3)$$

The scaling factor is denoted m_{xy}, and the two-dimensional pixels are expressed i^x *and* j^y.

3.1.4 Exacerbation of contrast using histogram equalization

The histogram normalization is used to increase the picture contrast. Assume that S and T correspondingly represent the row number (size) and column (length) in pixels. $S_l$is the color saturation of pixel number n_l, and M is the picture's numerically calculated brightness level. Then, using the continuity formula, a mapping procedure between BrightnessS_l and each pixel brightness T_l is carried out. The pixel brightness is expressed in Equation (4).

$$T_l = F(S_l) = \frac{M-1}{ST}(n_1 + n_2 + \cdots + n_L) \qquad (4)$$

The total number of pixels is M, and the color intensity and brightness are S and T. The current normalization is denoted n_l.

3.1.5 Conversion of color space from RGB to L*x*y

After the contrast becomes more intense, RGB transformation is applied. The primary goal of this transformation is to create the L*x*ycolor scheme. Since the transformed picture doesn't change, the image was transformed into the L*x*ycolor system. Since L*x*y contains all possible colors, there is no risk of image quality declines. Because of this, segmenting the pictures in k-means clustering works better. The RGB color system is first transformed, then that color scheme is changed to L*x*y.

Equation (5)translates from the RGB color system toXYZ color area.

$$\begin{bmatrix} i \\ j \\ k \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (5)$$

The matrix element is denoted m_{ij}. It can consider the tri-stimulus numbers for changing the L*x*ycolor area. S_b, T_b, and $R_b$are the standard whites for those quantities in this context. The red, green, and blue colours are denoted R, G, and B. The picture is denoted in Equation (6).

$$f(x) = \begin{cases} x^{-3} & if\ x > 0.02 \\ 2\pi + 0.13 & else \end{cases} \qquad (6)$$

The input feature of the image is denoted x, and based on the present location, the pixel value is denoted.

3.1.6 Segmentation of the image

The segmentation of the cauliflower photos is done using the k-means clustering technique. This grouping procedure, where k is the number of groups, is most often used. The efficiency of extracting features is excellent when the segmented quality is satisfactory. As “k” in this research is set to 3, the system will recognize three clusters in the picture while sectioning it.

3.1.7 Extraction of co-occurrence and statistical features

Two characteristics are retrieved following the separation of the disease-affected area: statistical and co-occurrence attributes, which are detailed in more depth in the next sub-section C. The phase of extracting features, which is important, extracts the most pertinent information, including color characteristics and pixel characteristics. In the illness detection phase, the deep learning classification is trained using all these collected characteristics collectively.

Pixels are identified using Local binary patterns (LBP), a straightforward and efficient texture generator, by normalizing each pixel's neighbourhood and converting the result to a binary number. The LBP has a better capacity for discrimination and is simpler to apply. In addition to the decimal integers, the LBP operation also identifies the picture pixels.

Each picture pixel is computed with its surrounding pixels during the labeling step by removing the average pixel value. The resulting negative numbers are also represented as 0, as well. The positive and 0 deals have been described as 1. All binary codes—known as LBP codes—are combined counter clockwise from the top-left to produce a binary code. The global picture, composed of several local representations, is created using the textural identifier.

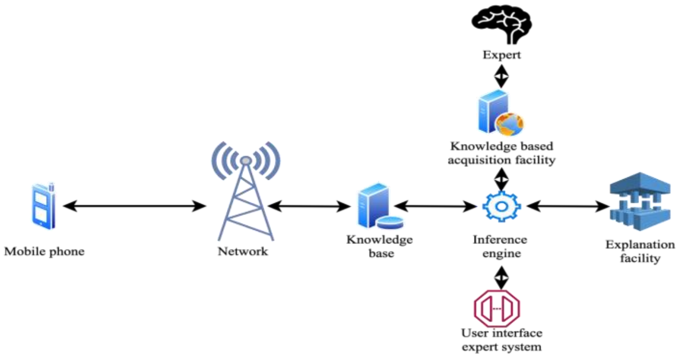

Fig. 2. The GNN-based mobile application for plant disease prediction model

The GNN-based mobile application for the plant disease prediction model is depicted in Fig. 2. The plants are monitored using the camera and the data is sent to the users based on the mobile application created by the designer. The features collected by the IoT devices are sent to the user and stored in the cloud. The interface engine is used to interface IoT and users. The knowledge base gives the necessary data for Cauliflower disease.

3.1.8 Training with six classifiers

The segmentation-generated feature matrices were fed into a classification for training and validation. Six commercial types were chosen: CNN [21], Deep Neural Network (DNN) [22], Random Forest (RF) [23], Decision Tree (DT) [24], Linear Discriminant Analysis (LDA) [25], and Principal Component Analysis (PCA) [26]. These algorithms examined the test dataset to provide a wide range of performance indicators in search of the most appropriate classification algorithm.

3.1.9 Output generations

The effectiveness of the classifications was evaluated using the testing database. The categorization types could not be determined or calculated using an appropriate matrix created from an unbalanced database since it raised several occurrences for widely different categories. Other indicators were applied to evaluate performances more precisely. The confusion matrices record the number of true positivity (TPs), true negativity (TNs), false positivity (FPs), and false negativity for binary issues or issues with two classes (FNs). When there were several categories or more than two classifications, the confusion matrix's size was nxn. There were n vertical lines, n sections, and nxn items in multi-class. The multi-class confusion matrices (is expressed in Equation (7)).

$$C_M = v_{xy} \qquad (7)$$

The two-dimensional vector is expressedv_{xy}, and the confusion matrix is denoted C_M.

3.2 Architecture of Graphical Neural Network with CNN for classification

There are two primary components to the GNN architecture:

1) A convolutional tool known as feature extraction isolates and groups the distinct aspects of pictures for study.

2) The outcome of the fully interconnected layer is subjected to convolution, which predicts the category of the picture depending on the information previously retrieved.

- **Convolution Layers**

The convolutional level, the max-pooling, and the fully-connected layer are the three main levels that make up the GNN. A GNN is created when these levels are layered. The dropping layer and the transfer functions are two additional layers on top of these three. The convolution layers' first level concentrates on extracting characteristics from the input pictures. Each input picture is mathematically combined with a certain size of convolution filtration at this level, performing the statistical convolution operation. A dot product between both the filtration and portions of the input picture matching the area of the filtration is calculated by moving the filter across the source images. The result is shown by feature mapping. The region of interest can be a source of data for other levels.

These are the arguments that the Conv2D function accepts:

1) Filtration - The variety of feature detection will apply various filtering techniques to the original picture to create the extracted features. Several filters include the Blur Filtration system and the Edge Identification Filters.

2) Kernel Length - This specifies the length of the (n x n) convolutional median filter.

3) Activation - A neuron's ability to be activated. Each layer except the output level has an activation value that is a Rectifier Linear Unit (ReLU) value. Using ReLU, they also have incorporated nonlinearity into the system. To find any linear correlations in the feature mapping, this is crucial.

4) Input Nodes: This layer is composed of several streams and input pictures (3 for color)

- **Pooling Layer**

The following layer in the fully convolutional system is the pooling level. The pooling element's primary goal is to reduce the spatial dimensions of data traveling across the network. Convolutional neural system support pooling in two separate methods. Median and maximum pooling. The Pooling Layer, the most popular of the two, scans the maximum value for each picture segment. The mean of an image's components within a predetermined size zone is determined via the pooling layer. Among the Convolution Operation and the Completely Connected Layer, the Max Pooling acts as a link.

- **Fully Connected Layer**

Biases and weighting are included in Fully Connected Levels, which link neurons between levels. The outputs of the previous convolution layers are flattened at this level, and each node in the following frame is connected to every other node in the following layer. Whether it's a convolutional level, ReLU level, or pooling module, this layer essentially accepts its inputs from the layer before it. The classifying procedure starts at this point.

- **Dropout layer**

The training database gets overfit when all characteristics are coupled to the fully connected layer. It is said to be over fitted if a system can function well on training databases but exhibits poor efficiency when used with fresh data points. A dropping layer is employed to address this issue, which reduces the scale of the model of neuronal networks by removing a small number of neurons during learning. A randomized 20% of the connections are deleted from the neuronal network upon passing dropouts of 0.2.

- **Activation layer**

An important part of the neural network procedure is the prominent ones. It chooses which data from the modeling should be fired ahead and which data shouldn't at the platform's end. As a result, it gives the network more nonlinear behavior. It has been noted that rather a few prominent ones are often employed. The most often used are the activation algorithms Sigmoid, tanH, Softmax, and Backpropagation. Each activating mechanism has a unique use. ReLU and Max-pooling functions are typically used for multiclass categorization.

1) ReLU: The most popular input layer in wired networks is the rectified linear units algorithm. The ReLU function has an advantage over the other kernel function because it does not simultaneously activate all cells, which is a benefit. Negative input is translated to 0, and the neurons are not triggered. The neurons are triggered and the affirmative value of x is returned if the input is affirmative. As a result, only a few neurons are active at once, creating a minimal and highly effective system. The ReLU functional also contributed to deep learning by resolving the disappearing gradient issue. The ReLU function is denoted in Equation (8).

$$ReLU = max(0, x) \quad (8)$$

The input feature is expressed x. GNN enhances the feature selection process using deep learning and IoT modules. These values are combined and sent to the user using mobile application.

2) Softmax: The classification's output level, where it attempts to obtain the possibilities to determine the category of each input, is where the max pooling is best utilized. It can classify data points and decide to which group they belong more easily. Without using pre-trained algorithms, convolutional neural networks will be utilized to categorize pictures. Some well-liked pre-trained algorithms can distinguish between hundreds of categories without learning each one individually. Because of their complicated designs, these models can manage tens of thousands of subclasses. A novice needs help to picture the scenario. Keras facilitate the creation of customized CNNs. This design was created using Customized CNN.

3.3 Disease and feature description

Recently, diseases have been the main obstacle to producing large cauliflower. Farmers can only make quick judgments if they have the necessary information and rules. The strategy is to examine cauliflower illnesses, help patients understand disease signs, and recognize the right cues. Several illnesses cause the decreased production of cauliflower. The research used visual signs to identify four important cauliflower diseases—bacterial soft, white rusty, black rotting, and downy mildew. A short explanation of various disorders is provided in the subsequent sub-sections for greater knowledge.

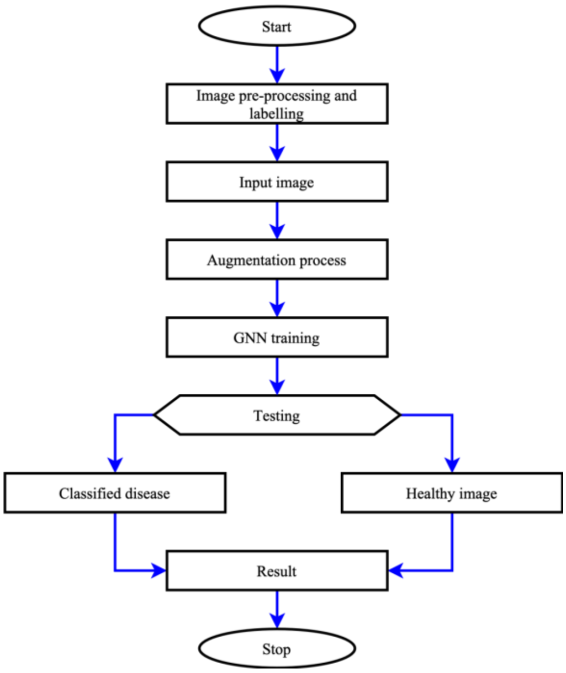

Fig. 3. The GNN-based cauliflower plant disease prediction model

The GNN-based cauliflower plant disease prediction model is shown in Fig. 3. The input from the cauliflower plant is collected and then the augmentation process is done. The GNN model is used to train the system. The input images are collected from IoT devices. The testing samples are analyzed and the results are displayed whether normal or diseased. If the disease is present, the classified results are plotted.

Bacterial soft rot: Carotovora, a bacteria that results in various diseases, is the source of bacterial soft rotting. The water-soaked patches grow to create a big revolving mass and cause damage to the leaves and blossom crowns, the wounded area of the leaflets breaks and leaks a slimy substance that turns darker brown or black when exposed to air. Warm, humid circumstances are necessary for the development of microbiological soft rotting.

White rusting: The oomycete AlbugoAlbicans is the culprit behind white rusting. It is an illness that results in white blisters on the stems and leaflets that can harbor different pathogens. White rust only appears in dry circumstances.

Black rotting: Xanthomonas campestris bacteria are the primary cause of black rot. The center of the leaves develops V-shaped lesions. The establishment of black rotting is reliant on warmer, humid weather.

Downy mildew: The oomycete Fusarium is the primary cause of powdery mildew. It first emerges on older leaflets as white, yellow, or reddish patches on the top edges and downy grey mildew on the matching undercarriage. It penetrates the plant through injuries and natural holes. Ultimately, the hue of these spots darkens, and the leaf withers. Powdery mildew needs damp and chilly environments to grow.

3.3.1 Extraction of features

This segment yields two kinds of extracted characteristics: GNN and statistical features. To identify textile defects, statistical aspects of high quality were used. Among the many statistical variables available, some were used to identify cauliflower sickness. The following is a description of the characteristics used for this study.

Mean (μ): The mean formula is expressed as follows: P is the number of pixels in defective areas, N is the number of pixels in imperfect-free areas, and G is the number of pixels with luminance and shades of gray in the faulty area. The mean is denoted in Equation (9)

$$\mu = \frac{1}{N}(G_1 + G_2 + \cdots + G_N) \qquad (9)$$

Standard deviation (std): The standard deviation formula is expressed as follows P is the total number of pixels in the defective areas, F_i is the total number of grey-level pixels in the badplaces, and C is the color brightness in the average grayscale. The standard deviation is expressed in Equation (10).

$$std = \frac{1}{\sqrt{T}}\sqrt{(F_1 - C)^2 + (F_2 - C)^2 + \cdots + (F_N - C)^2} \qquad (10)$$

Variance (var): The formula for variance is expressed as follows if P is the number of pixels in defective areas, F_i is the number of pixels of luminance with a gray level in faulty areas, and C is the color brightness in mean gray level. The variance is expressed in Equation (11).

$$var = \frac{1}{N}(F_1 - C)^2 + (F_2 - C)^2 + \cdots + (F_N - C)^2 \qquad (11)$$

Kurtosis (κ): The formula for kurtosis is expressed as follows if N is the number of pixels in defective areas, F is the number of pixels with intensity with grayscale in the badplace, and C is the color brightness in average gray level. The kurtosis is denoted in Equation (12).

$$K = \frac{\frac{1}{N}((F_1-C)^4+(F_2-C)^4+\cdots+(F_N-C)^4)}{\sqrt[2/3]{\frac{1}{N}((F_1-C)^2+(F_2-C)^2+\cdots+(F_N-C)^2)}} \qquad (12)$$

Skewness (γ): The skewness formula is expressed as follows if the mode, average, and standard error in defective areas for grey level luminance are, correspondingly, var, F, and T. the skewness is expressed in Equation (13).

$$\gamma = \frac{var-F}{T} \qquad (13)$$

3.3.2. GNN training model

Cat Swarm Optimisation (CSO) was created from inspiration gained from cat activity. Faster convergence allows the CSO model to resolve challenging optimization problems. These two cat behaviors—"seeking mode" and "monitoring mode"—are the most typical. Black widow insects' "unique breeding activity" also inspired bio-inspired optimization. The bio-inspired optimization method successfully solves complex objective functions with highly convergent results.

Additionally, the searching agents, in this instance, find global answers within the searching area. Based on the research, mixed optimization techniques demand a higher degree of resolution than conventional methods. This work combined bio-inspired optimization with the optimization concept; as a result, the hybrid technique suggested is the Cat Swarm Updating Black Widow prototype concept. Below is a diagram of the proposed model's phases:

Step 1: Start the M populace (pop) of the searching agent in the D-dimensional environment in step 1. The search agent's location is indicated by the letter X and the search agent's speed.

Step 2: The cats are dispersed randomly throughout the dimensional area, and the quantity is chosen randomly from among the ranges of possible velocities.

Step 3: The number of cats is chosen, and the tracking mode is established according to the "Mix Proportion (MR)." The other cats have been put into a seeking state.

Step 4: For the current cat, "Cat_i," J counts of clones are created in the seeking phase. Here, J is the Swarm Mixture Proportion

(SMP). Set J = (SMP-1) and designate the current cat as the greatest if the MR value = true. Then swap out the old ones for the new ones.

Equation (14) computes each political way's objective functions (Fit).

$$L_i = \frac{F_x - F_y}{F_{max} - F_{min}} \quad (14)$$

The formula calculates the selection probability (F_x and F_y) for each candidate point whenever all are not comparable. The maximum and minimum features selected by the CSO using the deep learning model are $F_{max}\ and\ F_{min}$. The selection likelihood is adjusted to 1 for each alternative energy when the fit is equal for all candidate points. Since reduction is the goal here, $F_y = F_{max}$. The location of the cats Cat_i is changed, and a randomized location is chosen to move out from the potential sites.

Step 5: The cat travels at its speed in every dimension while in the tracing phase. The tracking model's stages are illustrated as follows: (a) Utilizing the newly suggested expression found in Equations, the searching agent's speed is adjusted for each level. Therefore, the inertia weight is denoted by μ, τrepresents the randomly dispersed speed in the range [0, 1]. Additionally, the governing variables are and. The control variable utilized to steer the cats throughout their exploration phase is shown theoretically as (t) and supplied as per Calculations.

The pixel and iteration values are expressed in Equations (15a) and (15b).

$$\mu(p) = \mu_{max} - \frac{\mu_{max} + \mu_{min}}{p_{max}} \quad (15a)$$

$$\tau(p) = \tau_{min} + (\tau_{max} - \tau_{min}) \sin\left(\frac{\pi p}{p_{max}}\right) \quad (15b)$$

Check to see if the speed is within the maximum speed range. If the new rate exceeds the top speed band, set it to the highest speed range. $\mu_{min}$and μ_{max} denote the minimum and maximum values, respectively. The present repetition is pointed to as p, while the maximum repetition is given as p_{max}. Additionally, $\tau_{min}\ and\ \tau_{max}$ stand for the value of the initial and final iterations, correspondingly.

Instead of utilizing the conventional CSO updating method, update the location of Cat_i using themutation updating concept. The Mute-pop value is chosen randomly from the populace using the mutations updating mechanism (pop). The modification rate is used to calculate the Mute-pop.

Step 6: Apply the Equation to determine the searching agent's fitness. The cat with the finest fitness value is believed to be the best X best option.

Step 7: The cats are relocated by their flags; if it is discovered that Cat_i is in the searching mode, the searching mode procedure is applied; otherwise, the tracking mode procedure is used.

Step 8: Re-pick the number of cats and put them in tracking and searching modes depending on the MR.

Step 9 is to finish.

The proposed GNN-PDP is designed to detect and predict the disease present in cauliflower plants. The condition is identified and classified based on the training model. The results are sent to the consumer using mobile applications. The IoT modules help to collect data from the plants whenever needed and reduce the workforce. The simulation analysis of the proposed model is discussed in the next section.

4. Simulation analysis and performance comparisons

The research and the controlled test were run in the Ubuntu 18.04 system using the following hardware: an Intel Core i7 9820X CPU, 128GB of memory, and anNvidiaGeForce. Cuda9.0 and the deep learning platform Pytorch were used for learning. Given the number of network variables, the learning set and testing accuracy batch sizes were determined to be 16 and 8, correspondingly, throughout the experimental design and monitoring procedure. All GNN models have a fixed iteration limit of 50.Modeling training's training speed is set at 0.0001, and the optimization uses GNN. Additionally, the final output unit categories are increased to 6, and all algorithms in the picture branch follow their respective network structures. All algorithms do not employ pre-training simulations to maintain the objectivity of comparative results.

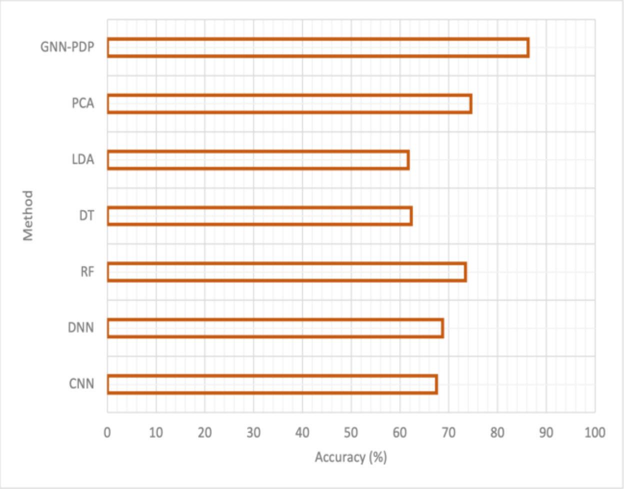

Fig. 4. The accuracy evaluation and comparisons

The accuracy evaluation and comparisons are shown in Fig. 4. The proposed GNN-PDP model is designed, and the simulation accuracy in detecting the disease is computed. The results are related to the existing values such as PCA, LDA, DT, RF, DNN, and CNN. The cauliflower disease is identified using the IoT devices placed on the farmland, and the condition is predicted using GNN. The prediction results are sent to the farm owner using the notification in mobile applications. The results show higher accuracy than the previous models.

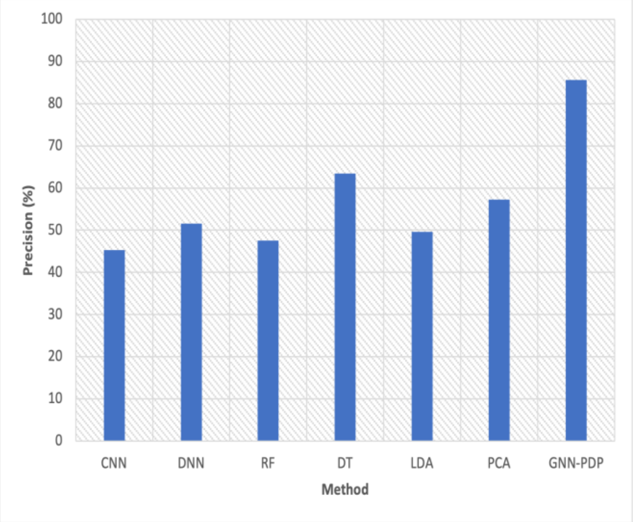

Fig. 5. Precision evaluation and comparison

The precision evaluation and comparison results of the proposed GNN-PDP are made, and the results are compared with the previously available models. The combined precision results for both existing and proposed GNN-PDP are depicted in Fig. 5. The precision shows the number of correct results predicted by the proposed model. The GNN-PDP outperforms the previous model by using the GNN and CSO deep learning model for optimization results. The IoT devices further enhance the data collection and transfer process.

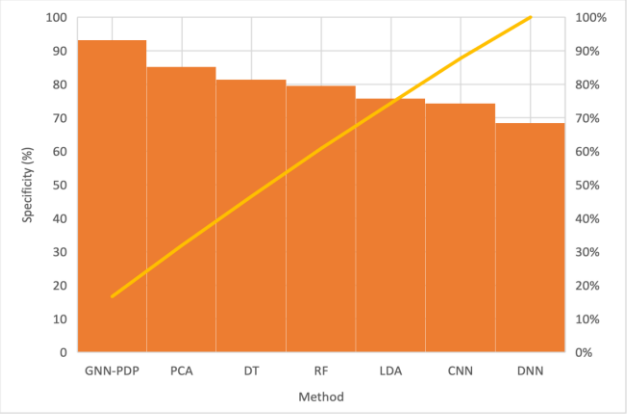

Fig. 6(a). Specificity evaluation and comparison results

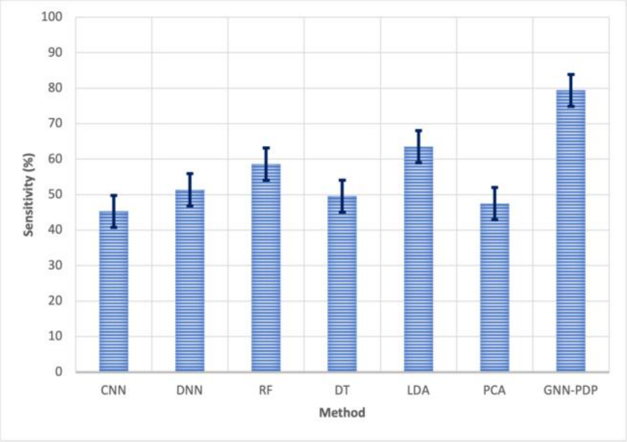

Fig. 6(b). Sensitivity evaluation and comparison results

The specificity and sensitivity evaluation and results in comparison of the proposed GNN-PDP model and the comparisons of the results with the existing models are shown in Fig. 6(a) and Fig. 6(b), respectively. The comparison results show the simulation results of cauliflower plant disease prediction, and the results show the highest performance of the proposed GNN-PDP model. The GNN-PDP enhances the simulation performance using IoT and GNN optimization models. The Cat Swarm Optimization algorithm further improves the results.

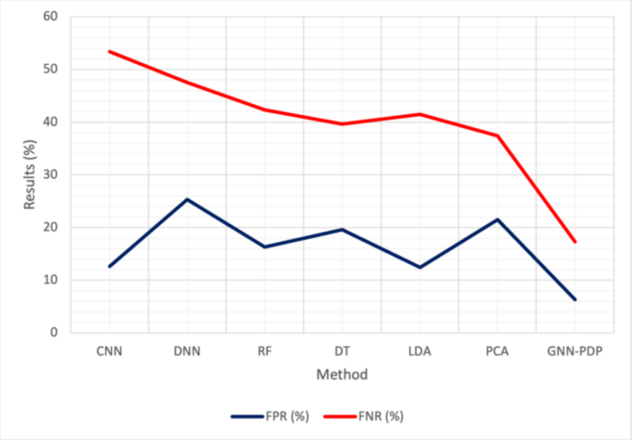

Fig. 7(a). FPR and FNR analysis of the proposed GNN-PDP

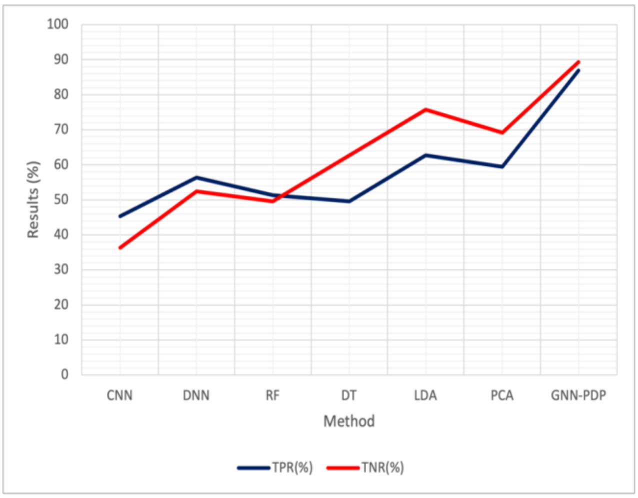

Fig. 7(b). TPR and TNR analysis of the proposed GNN-PDP

The False Positive Rate (FPR) and False Negative Rate (FNR) results are plotted in Fig. 7(a). The True Positive Rate (TPR) and True Negative Rate (TNR) results are depicted in Fig. 7(b). The combined results show higher simulation results in predicting and detecting the results of the cauliflower plant disease prediction from the collected IoT devices. The GNN model process the collected images from the IoT devices, and the classification results are sent to the users using mobile applications. The CSO algorithm enhances the classification of cauliflower plant disease.

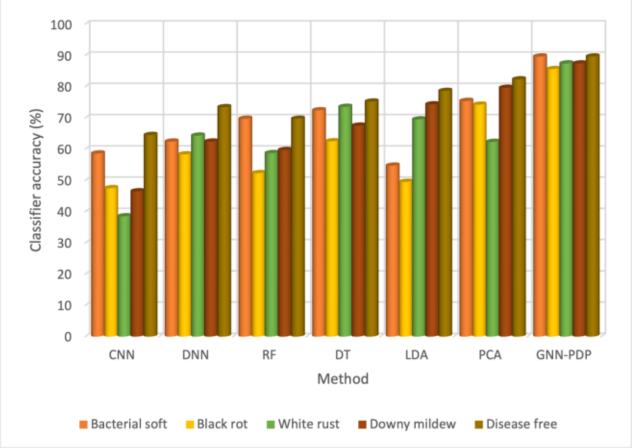

Fig. 8. Classification accuracy analysis of the proposed GNN-PDP

The classification accuracy analysis of the proposed GNN-PDP is computed, and the results are shown in Fig. 8. The cauliflower plant disease is predicted and compared with the existing models, such as CNN, DNN, RF, DT, LDA, and PCA. The simulation results show the highest classification accuracy of the GNN-PDP than the previously available models. The proposed GNN-PDP enhances the classification results using a graphical neural network and IoT devices. The CSO algorithm further improves the classification results with the deep learning model.

The proposed GNN-PDP model is used to detect and predict cauliflower plant disease. The IoT devices gather the cauliflower disease, and the results are processed with GNN. The CSO enhances classification accuracy.

Conclusion

A Graphical Neural Network based Plant Disease Prediction (GNN-PDP) model is designed in this research using the Internet of Things (IoT) and deep learning algorithms. The study created an intelligent system utilizing deep learning to identify illnesses in the popular and crucial vegetable, cauliflower. This empirical investigation made use of 750 photographs in total. The disease-affected areas of the recorded photos were segmented using the k-means classification technique. To identify cauliflower illnesses, two different feature sets were used. It used image processing methods to extract the image's characteristics. Following extracting features, the parts were put into six different classifications, with the GNN classification outperforming the others with an average accuracy of 89%.Applying deep learning algorithms to an expanded picture collection will improve the efficiency of identifying the cauliflower illness, but further study is required. The long-term research objective is to create a comprehensive intelligent system for automatically identifying diseases on various crops.

Acknowledgements

The author would like to appreciate the effort of the editors and reviewers. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Author contributions

All authors are equally contributed.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Kour, V. P., & Arora, S., “Recent developments of the internet of things in agriculture: a survey”. Ieee Access, 8, 129924-129957, 2020.

[2] Barbosa, M. W., “Uncovering research streams on agri-food supply chain management: A bibliometric study”, Global Food Security, 28, 100517, 2021.

[3] Idoje, G., Dagiuklas, T., & Iqbal, M., “Survey for smart farming technologies: Challenges and issues”, Computers & Electrical Engineering, 92, 107104, 2021.

[4] Dietz, T., Estrella Chong, A., Grabs, J., & Kilian, B., “How effective is multiple certification in improving the economic conditions of smallholder farmers? Evidence from an impact evaluation in Colombia’s Coffee Belt”, The Journal of Development Studies, 56(6), 1141-1160, 2020.

[5] Adeline Sneha, J., & Rekha, C., “Energy-efficient data transmission to detect pest in cauliflower farm”, In Soft Computing Techniques and Applications (pp. 659-670). Springer, Singapore, 2021.

[6] EL-Bauome, H. A., Abdeldaym, E. A., Abd El-Hady, M. A., Darwish, D. B. E., Alsubeie, M. S., El-Mogy, M. M. &Doklega, S. M., “Exogenous Proline Methionine, and Melatonin Stimulate Growth, Quality, and Drought Tolerance in Cauliflower Plants”. Agriculture, 12(9), 1301, 2022.

[7] Mrnka, L., Frantík, T., Schmidt, C. S., BaldassarreŠvecová, E., &Vosátka, M., “Intercropping of Tagetes patula with cauliflower and carrot increases yield of cauliflower and tentatively reduces vegetable pests”, International Journal of Pest Management, 1-11, 2020.

[8] Maria, S. K., Taki, S. S., Mia, M., Biswas, A. A., Majumder, A., & Hasan, F., “Cauliflower disease recognition using machine learning and transfer learning”, In Smart Systems: Innovations in computing (pp. 359-375). Springer, Singapore, 2022.

[9] Findura, P., Hara, P., Szparaga, A., Kocira, S., Czerwińska, E., Bartoš, P.,&Treder, K., “Evaluation of the effects of allelopathic aqueous plant extracts, as potential preparations for seed dressing, on the modulation of cauliflower seed germination”, Agriculture, 10(4), 122, 2020.

[10] Huang, Q., Yamada, M., Tian, Y., Singh, D., & Chang, Y., “Graphlime: Local interpretable model explanations for graph neural networks”, IEEE Transactions on Knowledge and Data Engineering, 2022.

[11] Khamparia, A., Saini, G., Gupta, D., Khanna, A., Tiwari, S., & de Albuquerque, V. H. C, “Seasonal crops disease prediction and classification using deep convolutional encoder network”, Circuits, Systems, and Signal Processing, 39(2), 818-836, 2020.

[12] Sharma, R., Das, S., Gourisaria, M. K., Rautaray, S. S., & Pandey, M., “A model for prediction of paddy crop disease using CNN”, In Progress in Computing, Analytics, and Networking , pp. 533-543, Springer, Singapore, 2020.

[13] Sethy, P. K., Barpanda, N. K., Rath, A. K., & Behera, S. K. Nitrogen deficiency prediction of rice crop based on convolutional neural network. Journal of Ambient Intelligence and Humanized Computing, 11(11), 5703-5711, 2020.

[14] Kundu, N., Rani, G., Dhaka, V. S., Gupta, K., Nayak, S. C., Verma, S., &Woźniak, M., “IoT and interpretable machine learning-based framework for disease prediction in pearl millet. Sensors”, 21(16), 5386. 2021.

[15] Xu, W., Wang, Q., & Chen, R., “Spatio-temporal prediction of crop disease severity for agricultural emergency management based on recurrent neural networks”, GeoInformatica, 22(2), 363-381, 2018

[16] Das, S., & Sengupta, S., “Feature Extraction and Disease Prediction from Paddy Crops Using Data Mining Techniques”, In Computational Intelligence in Pattern Recognition, pp. 155-163. Springer, Singapore, 2020.

[17] Picon, A., Seitz, M., Alvarez-Gila, A., Mohnke, P., Ortiz-Barredo, A., &Echazarra, J., “Crop conditional Convolutional Neural Networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions”, computers and Electronics in Agriculture, 167, 105093, 2019.

[18] Hernández, S., & López, J. L., “Uncertainty quantification for plant disease detection using Bayesian deep learning”, Applied Soft Computing, 96, 106597, 2020.

[19] Farooqui, N. A., Mishra, A. K., & Mehra, R., “Concatenated in-depth features with modified LSTM for enhanced crop disease classification”, International Journal of Intelligent Robotics and Applications, 1-25, 2022.

[20] Toseef, M., & Khan, M. J., “An intelligent mobile application for diagnosing crop diseases in Pakistan using a fuzzy inference system”, Computers and Electronics in Agriculture, 153, 1-11, 2018.

[21] Lu, J., Tan, L., & Jiang, H., “Review on convolutional neural network (CNN) applied to plant leaf disease classification”, Agriculture, 11(8), 707, 2021.

[22] Nigam, A., Tiwari, A. K., & Pandey, A. “Paddy leaf diseases recognition and classification using PCA and BFO-DNN algorithm by image processing”, Materials Today: Proceedings, 33, 4856-4862, 2020.

[23] Pankaja, K., & Suma, V., “Plant leaf recognition and classification based on the whale optimization algorithm (WOA) and random forest (RF)”, Journal of The Institution of Engineers (India): Series B, 101(5), 597-607. 2020.

[24] Basavaiah, J., & Arlene Anthony, A., “Tomato leaf disease classification using multiple feature extraction techniques”, Wireless Personal Communications, 115(1), 633-651, 2020.

[25] Sabzi, S., Pourdarbani, R., &Arribas, J. I., “A computer vision system for the automatic classification of five varieties of tree leaf images”, Computers, 9(1), 6, 2020.

[26] Gadekallu, T. R., Rajput, D. S., Reddy, M., Lakshmanna, K., Bhattacharya, S., Singh, S. &Alazab, M., “A novel PCA–whale optimization-based deep neural network model for classifying tomato plant diseases using GPU”, Journal of Real-Time Image Processing, 18(4), 1383-1396, 2021.

[27] Ch Lavanya Ratna, & Y Srinivas., “A Hybrid of NHPP and Generalized Gaussian Mixture Model: A Combinatorial Approach for Background Elimination”, Journal of Advanced Research in Applied Sciences and Engineering Technology, 34(1), 1–14, 2023.

[28] Sulastri Sabudin, Muhammad Eric Zulkarnaen, Akmal Nizam Mohammed, & Mohd Faizal Bin Mohideen Batcha., “Numerical Investigation of Temperature Distribution in a Container-type Plant Factory ”, Journal of Advanced Research in Applied Sciences and Engineering Technology, 28(2), 90–101, 2022.

[29] Safwan Mahmood Al-Selwi, Mohd Fadzil Hassan, Said Jadid Abdulkadir, & Amgad Muneer., “LSTM Inefficiency in Long-Term Dependencies Regression Problems”, Journal of Advanced Research in Applied Sciences and Engineering Technology, 30(3), 16–31, 2023.

[30] Syazwan Izharuddin Mohamad Sabri, Sazali, N., Ahmad Shahir Jamaludin, Wan Sharuzi Wan Harun, Kumaran Kadirgama, & Devarajan Ramasamy., “Investigation on Water Quality for Farmed Aquatic Species by IoT Monitoring System”, Journal of Advanced Research in Applied Sciences and Engineering Technology, 31(3), 317–327, 2023.