Design and Develop Novel Framework for Plant Disease Detection Using Convolution Neural Network, RandomForest Classifier and Support Vector Machine

Ashish Gupta, Research Scholar,Dept of CSE, RNTU,
India.Sanjeev Kr Gupta, Dean Engineering, RNTU, Bhopal,
India Pritaj Yadav, Asso.Prof., Dept of CSE, RNTU,
Bhopal,IndiaDeepak Gupta, Asso Prof., Dept of CSE, ITM,
Gwalior, India

Email:guptaashishnitm@gmail.com,sanjeevgupta73@yahoo.com,
yadavpritaj@gmail.com,deepak.gupta@itmgoi.in

ABSTRACT

Plant diseases are undesirable conditions that significantly reduce crop growth and quantity. Observing plants for disease with the naked eye is a common practice among expert biologists and farmers, although it can be inaccurate and take a lot of time. In this research, we build and construct an intelligent classification method for leaf diseases using computer vision and artificial intelligence approaches. In this study, two approaches are used, and the results of their simulations are analyzed to evaluate performance. Convolutional neural networks are utilized to extract the deep properties of the plants from the photos of the plants from the rice, potato, and tomato Plant Village data set. These characteristics are categorized using a Bayesian optimum SVM classifier, and the outcomes are evaluated in the form of precision, sensitivity, accuracy, and f-score. The aforementioned approaches will allow farmers all around the world to act quickly to save their crops against suffering irreversible damage, avoiding both a global economic disaster and their own personal financial crisis. The second part of the approach extracts the texture and color information using the HoG, GLCM, and color moments histogram after processing the pictures from the data set. Color, texture, and deep features are the characteristics mixed in this case creating hybrid features Optimization of the binary particle swarm is used to include those hybrid features, and then the simulation outcomes are categorized utilizing a random forest classifier. In hybrid feature selection, binary particle swarm optimization is essential; this algorithm's object is to produce the desired output with the fewest features possible. Utilizing the aforementioned evaluation criteria, a comparison of the two methodologies is performed.
Keyword: Deep learning, GLSM, SVM, HoG

1. INTRODUCTION

Agriculture productivity can drastically fall as a result of diseases, pests, and other unfavorable elements in crops. The influence of these hazardous elements on crops directly affects the deterioration in crop productivity along with quality. The term "pesticides" was created to describe approaches for preventing, controlling, and lessening the effects of biological organisms and diseases [1]. Typically, visual examination focused on the morphology, appearance, and other properties of the leaves are used to identify plant diseases and insects. It is advised that only a highly qualified biologist do and interpret this visual examination because a misdiagnosis could result in an irreversible loss of productivity. It should be observed that research on pest and disease control is typically expensive. The names "machine learning" (ML) and "deep learning" (DL), which, in terms of convenience, allow computers to "learn" a huge number of patterns and then take action, were coined in response to the numerous uses of AI in daily life, have also been created. Without also being specifically created to do so, ML and DL enable software applications to increase the overall accuracy rate. Intelligent algorithms have emerged as a result of the connection between computer vision and DL technology that examine and categorise patterns or objects with better accuracy than the regular person. Computer vision concentrates on training computers to consider and act with the least amount of human involvement, whereas DL focuses on computers learning to think by using nervous system-inspired architecture [2,3].

2. Literature Survey

For identifying plant diseases. Due to the aforementioned, computing for the auto detection of plant diseases has grown. This technique has led to the creation of a method for identifying diseases of foliage and then using a smartphone that uses two network architectures, AlexNet [4] and Google Net [5]. After training, the proposed approach had a 99.35 percent accuracy rate for identifying leaf infections. The R-CNN diagnostic algorithm, which is quicker than CNN, was utilized for the analyzing algorithm [6] to find early-grown maize and tell it apart from weed species in three distinct climates, with an overall 97.71 result of percent. In Combining the quickest sensor, deep learning, and a ResNet50 neural network, this study achieved an accuracy of 95.78 percent and overlay 89.85 percent from another investigation by Yu and colleagues [7] to recognize fruits that eventually be applied in a robot that picks strawberries. The quicker R-CNN in combination with the VGG-16 architecture resulted in improved results when Fuentes and others [8] examined the three various sensor family types concerning various architectural approaches. Article "Plant diseases can significantly harm crops by drastically lowering their yield [9] because they impede the expansion of crops and result in subpar goods [10]. Biofuel crops are less productive and produce less fiber as agriculture tries to feed the world's expanding population. Using the human eye is an established way of seeing and identifying plant leaf infections. However,as the symptoms are assessed based on their own experiences, manual identification can lead to misdiagnosis [10]. Furthermore, the spread of infection and plants must be consistently monitored. Given how much time it takes to perform this continuous monitoring, it is a challenging task [11]. As a result of the aforementioned, computational models for automatic detection have been developed, such as those described in. With the use of mobile phones, they developed a method for the detection of disease in plant leaves. They used two CNN models, Alex Net [12] and Google Net [13], using a set of 54,306 photos of healthy and sick leaves that were made accessible to the public

CNN model was developed by Sladojevic and associates to categorize 13 different common disorders. Additionally, the results were accurate to 91 percent and 98 percent, yielding an overall average of 96.3 percent. Based on images of tomato leaves, Brahimi and colleagues [14] used a CNN model to categorize tomato leaf infections. They have employed visualization techniques to comprehend the indications of the deep model and, as a result, identify the areas affected by the leaf illness. The accuracy of the findings was 99.18 percent. In another study, Lu and others [15] suggested utilizing CNN to identify the 10 using most prevalent rice illnesses real-world photos of both healthy and ill rice stems and leaves that were taken in the testing field. In the end, their model had a 95.48 percent overall accuracy. Kawasaki as well as others [16] suggested using CNN to distinguish between healthy and unhealthy cucumbers using photos of the leaves. The algorithm showed a 94.9 percent average accuracy when dividing up a cucumber into two classifications that often have illnesses and one healthy class. However, the prior study only employed a small number of designs. The three pretrained CNNs that are suggested to be assessed in this study [17] are Alex net, Google Net, and InceptionV3. Squeeze Net [18], ResNet50 [19], and ResNet101 [20] are also proposed to be examined. This study offers a thorough resource for researchers who want to create and use a system that is adequately thorough in the categorization of various ailments that affect agriculturally utilized plant leaves by incorporating five different types of crop species. In this work, we most often compare used imaging techniques for classifying and analyzing objects. Our research attempts to reproduce the algorithm that offers the most precise predictions for the diagnosis of plant leaf disease. The outcomes are anticipated to be used to decide which algorithm is best for developing a smart system.

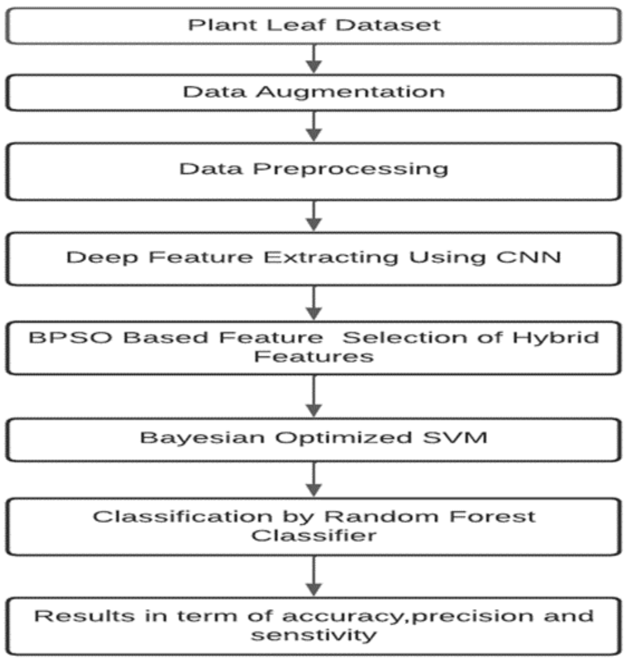


Fig.1: A block diagram for a hybrid technique for detecting plant leaf disease.

3. Proposed Methodology

This work describes the detection of leaf diseases problem in rice, tomato, as well as potato leaves using deep features, texture, and color characteristics, followed by feature selection using BPSO and comparing two classifications: Bayesian optimal SVM and random forest. Fig. 1 depicts the planned work's general flow chart.

4. Image Acquisition

with CNN To identify leaf illness in tomato, rice, and potato leaves, this study compares the two classes, random forest, and Bayesian optimal SVM —by first extracting deep features, texture, and color characteristics, then choosing features based on BPSO. Figure 1 displays a conceptual flowchart of the proposed work. A large amount of data is necessary to practice clever systems for classification and visualization. ML and DL systems typically outperform traditional learning methods when taught on massive volumes of data. We used the Plant Village database in this article [21]. 41,124 images of 12 distinct plants make up this "data set," which is broken down into 38 groups of healthy leaves and photographs of plants with various diseases. Potatoes (2,245 images), tomatoes (11,445 images), and rice (3224) were the three types of plants that made up the 16,914 images used in this study. The volume of the data must be split into plans to practice and analysis sets to train and assess deep learning methods. The data set produced by this investigation is distinctive since it includes images of various shapes and sizes and offers additional sensor power. Images of plant leaves were taken from the Village plant data set and then processed, features extracted, features selected, and features categorized. Examples of numerous leaf diseases on potato, tomato, and rice plants are shown in fig. 2 , respectively.

4.1 Data Augmentation: This implies that the role model had to be able to understand the key characteristics of a set of data during training. These things are required for this

(i)Data Space: Data space for learning contains the most examples that are representative of the context in which the model will be used. It thus encompasses the whole range of possibilities.

(ii) Features: The training data's feature space, which includes the highest limit of representations of each data feature, also encompasses the full range of possibilities. We must therefore gather the widest range of training images that are relevant to the objective of our model is satisfy the context of use and the first criteria. The most popular data augmentation techniques are affine transformations (flip, rotation), which must be applied to the training images we have available to fulfill the second condition. Nonaffine alterations include variations in perspective, brightness, and contrast wraps, random resizing cropping of random portions of images, jittering, and cutouts, among others, squares of random blacks.

4.2 Convolutional neural networks for the extraction of deep features (CNN). Following the initials CNNs, a distinct class of Neural networks is recommended for processing data using grid structures or neurall networks. The typical sorts of information utilized here of network are images (a network of x and y pixels), time series (one-dimensional data with an additional dimension, such as a time measurement), 3D data, such as a scanner (two measurements associated with a one- dimensional image, typically related to the evolution of video in time), and scanner data. [22]. CNN has been applied to a wide range of functions with significant success. Recently, Deep CNN have been employed in place of humans eyesight for image recognition. [23]. Fig.3 depicts the conventional deep convolution neural network structure.

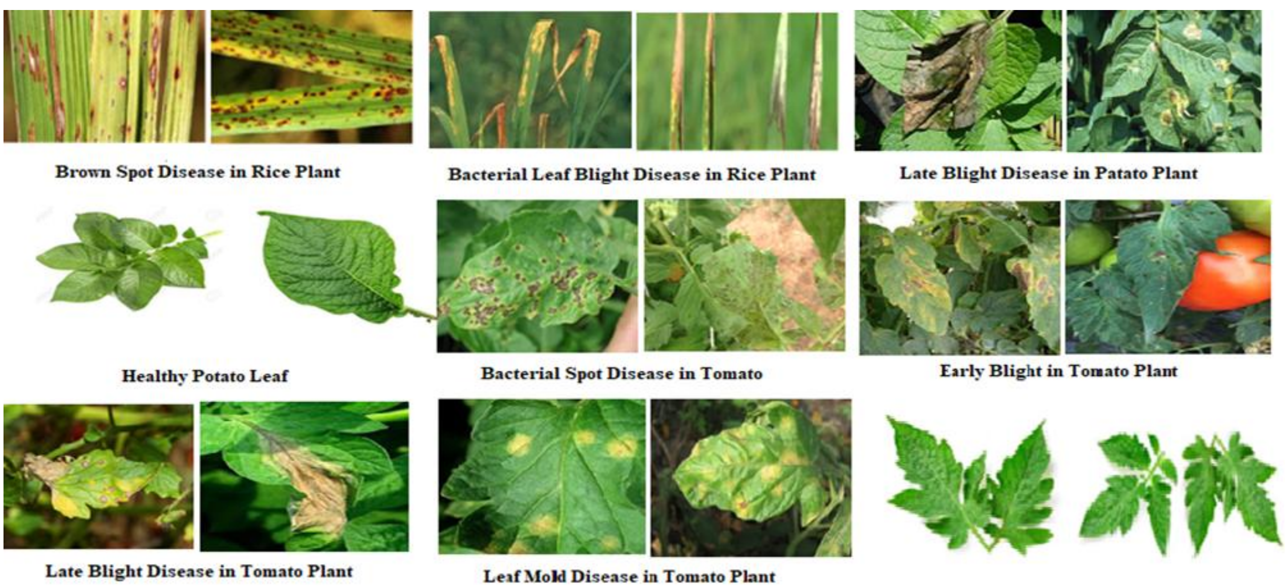


Fig 2: Various Diseases in in Rice Potato and tomato Plants Leaves [22]

4.2.1 The convolutional operation, "A method using two functions and real numbers as arguments is called convolution. The below mathematical equation describes the convolution operation:

$$u(t) = \int y(b)z(t-b)db. \quad (1)$$

Typically, the convolution procedure is indicated by:

$$u(t) = (y \times z)\,(t). \quad (2)$$

The first term (here (y)) is typically referred to as the input to a stable operation, While the second parameter (in this example, z) is referred to as the CNN kernel. The feature map is the outcome or product of this rotation. The integral functions of logical functions must be transformed into a group of continuous "discrete" functions as indicated below in order to discrete data when working with a computer

$$u(t) = (y\times z)\ (t) = \sum_{a=-\infty}^{\infty} x(b)z(t-b) \qquad (3)$$

In deep learning, the input is often a multidimensional vector, while the kernel updated by the learning process is typically a multidimensional parameter vector. For instance, when the image P is the input, it is typical to use a two-dimensional kernel represented by:

$$U(p,q) = (P\times R)\ (p,q) = \sum_a \sum_b P(p-a, q-b)R(a,b). \qquad (4)$$

Discrete convolutions can be thought of as matrix multiplication in real life.
Fig. 4 shows how the two-dimensional input P functions as the kernel of R, a two - dimensional array, and moves along input P while executing addition and multiplication on an element-by-element basis. Get the (P×R) result of the convolution.

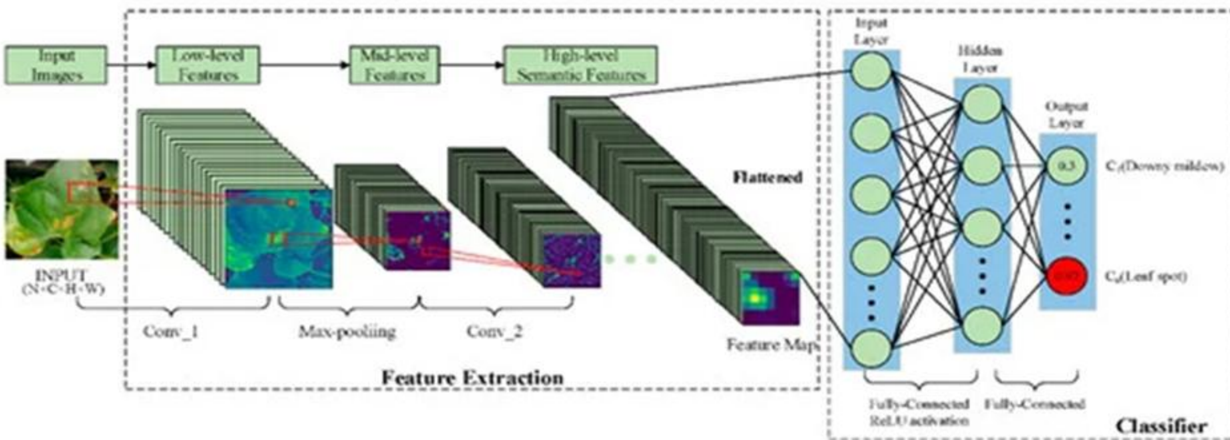
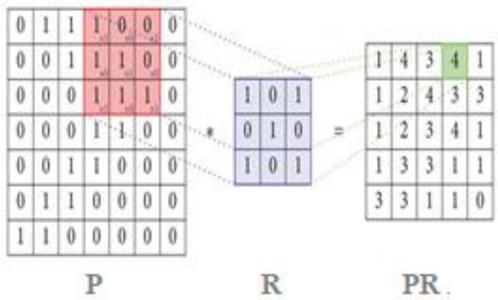


Fig.3:Basically, deep neural network architecture[29]

4.2.2 Pooling: A CNN typically has three layers. The operation of the input data is folded at the network's initial level. All decontaminated functions are transferred to activation functions in the second stage; the Reformed linear unit is the most used activation function. The perceptual stage is another name for this phase [24].
The final stage involves performing the join function, which substitutes a quantitative measure produced by the preceding neural network layer area for the prior output or network output. Figure makes this much clearer to see.

Max Pooling: This procedure chooses the highest possible value for each parameter, reducing the attributeor value by a factor of 4.

Average Pooling**:** This approach reduces data by four times by choosing to use the region's arithmetic mean. tributes or value by a factor of 4.



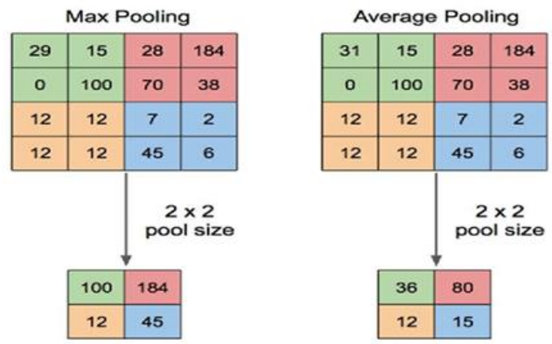


Fig.4:2-Dimensional Convolution function of CNN**Fig.5**: Pooling Function

4.2.3 Model, Architectures. Despite having a wide range of options for them, this model was picked to accomplish the particular task at hand. CNN is a pioneer in several cutting-edge computer vision tasks [25]. Particularly for those already well-established in the area and doing well on computer vision tasks, the trade-offs between the number of layers and trainable properties, such as the processing costs necessary for training, must be evaluated. Given the circumstances, CNN was chosen. Given the circumstances, Consequently, these are the chosen architectures
:

(1) AlexNet: To achieve the categorization, this network has three fully connected layers at the end and five convolutional layers up front. It aims to employ a CNN architecture with effective performance reported in related research. Additionally, it has max pooling and dropout intermediate layers [26].

(2) LeNet: This network, which was first created to recognize handwritten numerals, is composed of two convolutional layers, followed by layers of maximum pooling to extract attributes. After the final convolutional layer, two fully connected layers are added to categorize the output. [27].

(3) MobileNet: This CNN is made for use on mobile andembedded devices and is constructed using deep separable convolution operations, which reduce the workload of operations to be taken out in the first layers. [28].

(4) Shuffle Net: the channel shuffle, which randomly shuffles the output channels of the convolutions in the group, and the so-called group convolutions, which are multiple convolutions with each convolution containing a portion of the input channels, serve as the approach's cornerstones. Supporters of this design claim that it has a low computing cost and produces reliable results [29]

(5) Eff Net: It is comparable to the Mobile Net and Shuffle Net networks in terms of how in-depth separable convolution processes are used. For several well-known databases, it outperforms state-of- the-art performance while introducing a novel convolutional block that lowers the computational cost. [30].
Taking into account the aforementioned architectures, Table 1 displays the total, trainable, and nontrainable parameters. Noting that the total number of parameters is fewer than that of the VGG16 and VGG19 architectures, it is possible that their combination, as determined by a committee, may be proven to be less expensive than well-known architectures in the research, given the conditions taken into account. The under-consideration architectures were each trained using the previously described methods and assessed independently based on how well they performed on the test set[31].

4.3 Classification of Deep Features by Bayesian Optimized Support Vector Machine. In supervised learning, a set of examples are presented to train the classifier (input and desired output). In supervised approaches, the model is trained to produce the required output using a training set. Since the training set includes both correct and bad results, the model can get better over time.

Table 1: Consider the CNN parameter

Architecture	Trainableparameters	Non-Trainableparameters	Totalparameters
LeNet	2,769,641	0	3,983,446
ShuffleNet	1,196,336	22,146	1,876,565
AlexNet	2,447,921	0	2,941,571
Eff Net	404,587	1,734	316,834
MobileNet	1,444,295	12,649	1,719,558
Total	8,262,780	36,529	10,837,976

They use an ideal splitting hyperplane that is positioned precisely in the middle between the margins of the two classes to partition the feature space into regions while taking the SVM into account. The SVM analysis can make use of nonlinear functions such as quadratic, polynomial, radial basis function, Gaussian, and two-layer perceptions.

With this method, the separation margin between samples from two groups is maximized. It can be described by the equation below:

$$Max\ W(\lambda) = \sum_{p=1}^{N} \lambda p - \frac{1}{2} \sum_{p.q=1}^{N} y_p y_q \lambda_p \lambda_q (x_p x_q),$$

$$\text{Subject to } \begin{cases} 0 \leq \lambda p \leq C, \\ \sum_{p=1}^{N} \lambda_p y_p = 0\ , p = 1,2\_\_, \mathcal{N} \end{cases} \quad (5)$$

where y is the intended output, x is the input samples, N is the sample size, λp are the Lagrange multipliers, and C is the error limit.

Keep in mind the Bayes theorem. When the conditional probability P(B|A) for two events A and B is known, the probability B) is defined as follows:

$$P(A|B) = P(B|A)P(A) / P(B)\ , \qquad (6)$$

P(A) stands for the prior probability, P(B|A) for the likelihood of event B dependent on the occurrence of event A, and P(A|B) for the posterior probability.

The SVM method yields lesser accurate findings since it uses sampling methods for continuous parameters. In the suggested work, a technique that lets you change the SVM parameters is examined. Sampling is very helpful in studies. One of the most crucial factors in figuring out how reliable your study results are is this. The outcome will be affected if there is an issue with your sample. There are many different kinds of sampling methods, including multistage and sample random sampling.

The optimization of continuous variables and mixed discrete and continuous variables by tackling issues with various data sources is one of the applications of Bayesian optimization. The selection of the optimal value for each SVM parameter is the major objective of this application of Bayesian optimization. The settings for the kernel and the acquisition functions are two of the most important practical considerations that we must make. The common covariance function is the squared exponential kernel. [32].

$$K_{M52}(x, x^{'}) = \Theta o(1 + \sqrt{5r^2(x, x')} + \frac{5}{3} r^2\ (x, x'))exp\{-\sqrt{5r^2(x, x')}\ )(7)$$

The observation noiseν and the covariance amplitudeΘo are two of the few parameters that need to be handled for the aforementioned kernel function. This may be done by calculating the integrated acquisition function and marginalizing over hyperparameters.

4.4Pre-processing for the Second Phase, The first step of the approach was described in the preceding subcategories, and the second step begins here. Initial downsizing reduces the supplied image to 300×450 pixels. Given that this image is in RGB format, its deep properties and texture will be extracted by converting it to greyscale. Also, the color characteristics are converted from RGB to LAB.

4.5Extraction of Colour and Texture Features. To gauge an image's brightness and sharpness, moments of color are computed. The standard and standard deviation are the color moments that were employed in this investigation to exclude color pictures. The standard deviation may be calculated by taking the square root of the variance, and the mean can be interpreted as the average of the colors in the image. The color distribution is used in the histogram approach. We have a lot of data to store. Only the main color characteristics, such as the mean and standard deviation, are calculated rather than the entire distribution [33]

$$\text{Mean}=E_i = \sum_{q=1}^{N} \frac{1}{N} M_{pq} \qquad (8)$$

$$\text{Standarddeviation}=\sigma_i = \sqrt{\left(\frac{1}{N}\sum_{q=1}^{N}(M_{pq} - E_q)^2\right)} \qquad (9)$$

4.6 The GLCM shows the connection between two neigh boring pixels in gray-level imagery. Based on pixel spacing and angle, it is selected. The gray-level cooccurrence matrix (GLCM), also known as the gray-level spatial dependency matrix, is a static method of texture analysis that considers the spatial relationships between pixels. These factors provide details regarding the calibre of the Image.

The spatial connection of the picture for the vector d is displayed by the GLCM, which is an N-dimensional square matrix that specifies the number of pixel pairings with p and q values [34,36]. A grayscale image will be expressed using the function I. (r, c). Let the spatial relation vector d (dr, dc) be defined. The following is how the confirmation matrix $C_d$is written [34]:

$$C_d\ (p, q) = |\{(r, c): I(r, c) = p and I(r + dr, c + dc) = q\}| \qquad (10)$$

In addition to their separation, the pixels in this pair are oriented at $\theta = 0°$, 45°, 90°, and 135 degrees. Cooccurrence matrices from a 4 × 4 image are shown in Figure 5 in three different ways [35].

The expressions for the symmetrical gray-level cogeneration matrix $S_d$and the normalized gray- level cogeneration matrix N_d are given below [36]:

$$N_d\ (p, q) = \frac{C_d\,(p,q)}{\sum_p \sum_q C_d\ (p,q)}) \qquad (11)$$

$$S_d\ (p, q) = C_d(p, q) + C_{-d}(p, q)$$

The normalized gray-level cooccurrence matrix may be used to determine an image's properties, such as its energy, contrast, homogeneity, and correlation. The following equations [35,36] are used to determine the required properties:

Energy = $\sum_p \sum_q {N_d}^2(p, q)$

Contrast =$\sum_p \sum_q\ (p - q)^2 N_d(p, q)$,

Homogeneity = $\sum_p \sum_q\ \frac{N_d(p,q)}{1+|p-q|}$,

$$\text{Correlation} =\frac{\sum_p \sum_q\ (p-\upsilon_p)(q-\upsilon_q) N_d\,(p,q)}{\sigma_p \sigma_q} \qquad (12)$$

Here, υ_p and υ_q are the row and column totals of the gray-level cooccurrence matrix, respectively, and $\sigma_p$and σ_q are the standard deviations of the totals. Calculated values for the image's energy, contrast, homogeneity, and correlation are present. The attribute vector for the image is built using these data

5. HoG (Histograms of Oriented Gradient)

HoG counts the presence of orientation gradients in certain localized regions of an image identification window or region of interest while processing images for object detection in computer vision (ROI). This approach is already used in libraries like OpenCV HOG Descriptor. Figure 13 shows what the HoG method looks like when it is implemented.

5.1. Feature Selection Based on BPSO

PSO is a learning method. It is a sophisticated program that handles a specific cattle herd. An approach to predictive optimization known as BPSO is frequently employed to address issues in continuous domains. Although it applies PSO's speed and inertia concepts continually to binary domains, the type of PSO in question performs poorly.

For the purpose of developing particle clustering algorithms, fish and avian behavior has been examined. Binary optimization was used for the first time [37]. If the problem can be simplified, removed, or decreased, binary optimization may be useful for solving it. A diverse area is used to test a variety of optimization-related difficulties, such as conversion and rendering problems.

The fitness function f maximizes for the search region S = $\{0,1\}^D$, or (max f(x)). This is how the pth particle in the D dimension is defined.

$$X_p = (x_{p1}, x_{p2}, x_{p3} \ldots x_{pd}\}T,\ x_{pd} \in \{0,1\},\ d = 1,2,3 \ldots D,$$

$$V_p = (v_{p1}, v_{p2}, v_{p3} \ldots v_{pd}\}T,\ v_{pd} \in \{-V_{max}, V_{max}\},\ d = 1,2,3 \ldots D, \qquad (13)$$

where, V_{max} is the greatest velocity vector. the current best position can be summarized as [38]:

$$P_i = (P_{i1}, P_{i2}, P_{i3} \ldots P_{id}\}T,\ P_{id} \in \{0,1\},\ d = 1,2,3 \ldots D, \qquad (14)$$

The following definitions can also be used to define the algebraic notation mentioned above. Formula for velocity,

$$V_{pd} = V_{pd} + C_1\ \mathrm{rand}d_1\ (p_{pd} + X_{pd}) + C_2\mathrm{rand}d_2\ (p_{gd} - X_{pd}) \qquad (15)$$

the position equation

$$X_{id} = \begin{cases} 1 & if\ U(0,1) < sigm(v) \\ 0 & otherwise \end{cases}\ d = 1,2 \ldots D;\ i = 1,2 \ldots . N. \qquad (16)$$

Transfer function:

$$sigm\ (v_{id}) : \frac{1}{1 + exp\ (-\lambda v_{id})} (17)$$

where g is the "index of the highest-performing particle," N is the "width of the fortification,"," p_{gd} is the "best portion," and "Social and cognitive component constants" are C_1, C_2; U (0, 1) random numbers, rand1, rand2. The sigmoid transform function is sigm(v_{pd}).

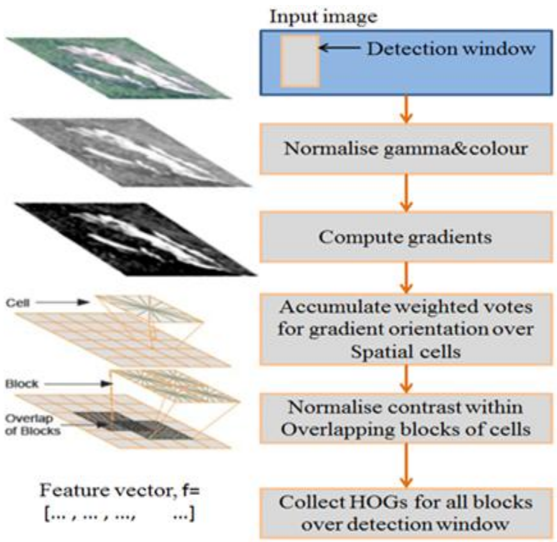


Fig. 6: Procedure for HoG approach [36]

5.2. Random Forest Classifier Classification

A random forest is a sort of classifier made up of a collection of classification trees with the name $\{h(x,T,\theta_k),k=1,2,...,k\}$where θ_k denotes uniformly spaced random vectors. The statistical unit is ultimately assigned to the class which received the majority of the votes, whereas the vast bulk of the random forest trees volunteered themselves [37]. In order to allocate the class's statistical subunit based on the vector of values x, each tree in the collection (forest) expresses just one vote.

1. The statistical properties of the random forest-based classification:

2. It is fairly robust when it comes to outliers and experimental noise.

3. It is speedier compared to many other numerical classification methods.

4. It can be effectively applied on parallel computers thanks to its simplicity and ease of parallelization

one of the key characteristics of random forests is that the generalization error converges "most certainly" for a number of diverging trees in the forest, and thus the potential of overfitting the entire classification technique is minimized as the number of trees increases.

Table 2:Analysis of parameter

TP (true positive)	"Reflect the amount of plant leaf diseases successfully classified was indicated"
TN (true negative)	"Reflect the amount of plant leaf diseases incorrectly classified"
FP (false positive)	"Reflect the amount of wrongly identified plant leaf diseases"
FN (false negative)	"Reflect the amount of plant leaf diseases that were incorrectly classified"

6. Results of the Simulation

6.1 Evaluation criterion. The assessment criteria used to assess the model's effectiveness are shown in Table 2.

Accuracy = $\frac{TN+TP}{TP+TN+FP+FN}$

Precision = $\frac{TP}{TP+FP}$

Sensitivity = $\frac{TP}{TP+FN}$

$$\text{F-Score} = \frac{2TP}{2TP+FN+FP} \tag{18}$$

The function of various classification techniques is examined in this section. These methods are based on a Bayesian-optimized SVM and a CNN. The identification of plant diseases requires both of these methodologies.Both approaches are essential for detecting plan leaf disease.

Case 1: Here TP= 56, TN= 221, FP=3, and FN= 19

Accuracy = $\frac{221+56}{56+221+3+19}$=$\frac{277}{299}$=92.64%

Precision = $\frac{TP}{TP+FP}$= $\frac{56}{56+3}$ = $\frac{56}{59}$ = 94.91%

$$\text{Sensitivity} = \frac{TP}{TP+FN} = \frac{56}{56+19} = \frac{56}{75} = 74.66\%$$

$$\text{F=Score}=\frac{2\times56}{2\times56+19+3}=\frac{112}{134}=83.58\% \tag{19}$$

Case 2: Here TP= 72, TN= 221, FP=3, and FN= 8

Accuracy = $\frac{221+72}{72+221+3+8}$ =$\frac{293}{304}$ = 96.38%

Precision = $\frac{TP}{TP+FP}$ = $\frac{72}{72+3}$ = $\frac{72}{75}$ = 96.00

$$\text{Sensitivity} = \frac{TP}{TP+FN} = \frac{72}{72+8} = \frac{72}{80} = 90.00\%$$

$$\text{F=Score} = \frac{2\times72}{2\times72+8+3} = \frac{144}{155} = 92.90\% \tag{20}$$

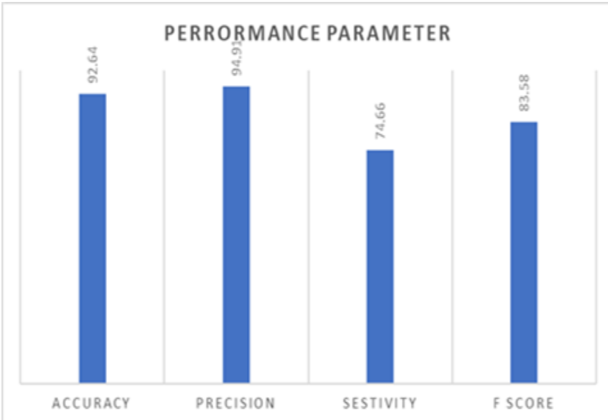
Case 3: Here TP= 55, TN= 223, FP=2, and FN= 18

$$\text{Accuracy} = \frac{223+75}{55+223+2+18} =\frac{278}{298} = 93.28\%$$

Precision = $\frac{TP}{TP+FP}$ = $\frac{55}{55+2}$ = $\frac{55}{57}$ = 96.49

$$\text{Sensitivity} = \frac{TP}{TP+FN} = \frac{55}{55+18} = \frac{55}{73} = 75.34\%$$

$$\text{F=Score} = \frac{2\times55}{2\times55+18+2} = \frac{110}{130} = 84.61\% \tag{21}$$



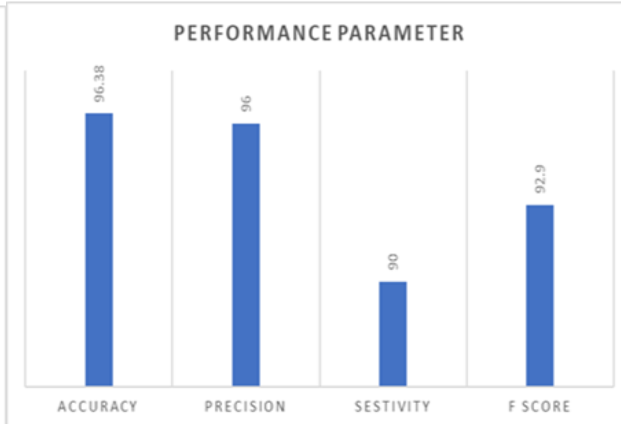


Fig.7: Efficiency graph for potato plan**Fig.8**: Efficiency graph for tomato plant

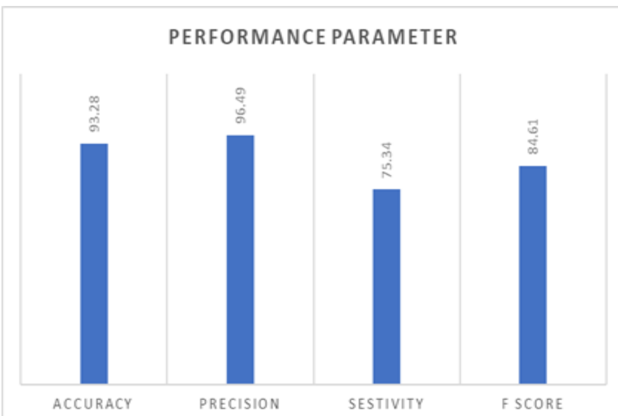
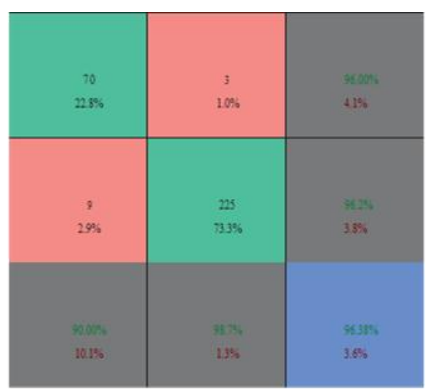


Fig 9: Efficiency graph for rice plant**Fig.10**: Efficiency comparison graph for all plants



Target ClassTarget Class

Fig 11: Confusion matrix for potato plant**Fig 12**: Confusion matrix for tomato plant

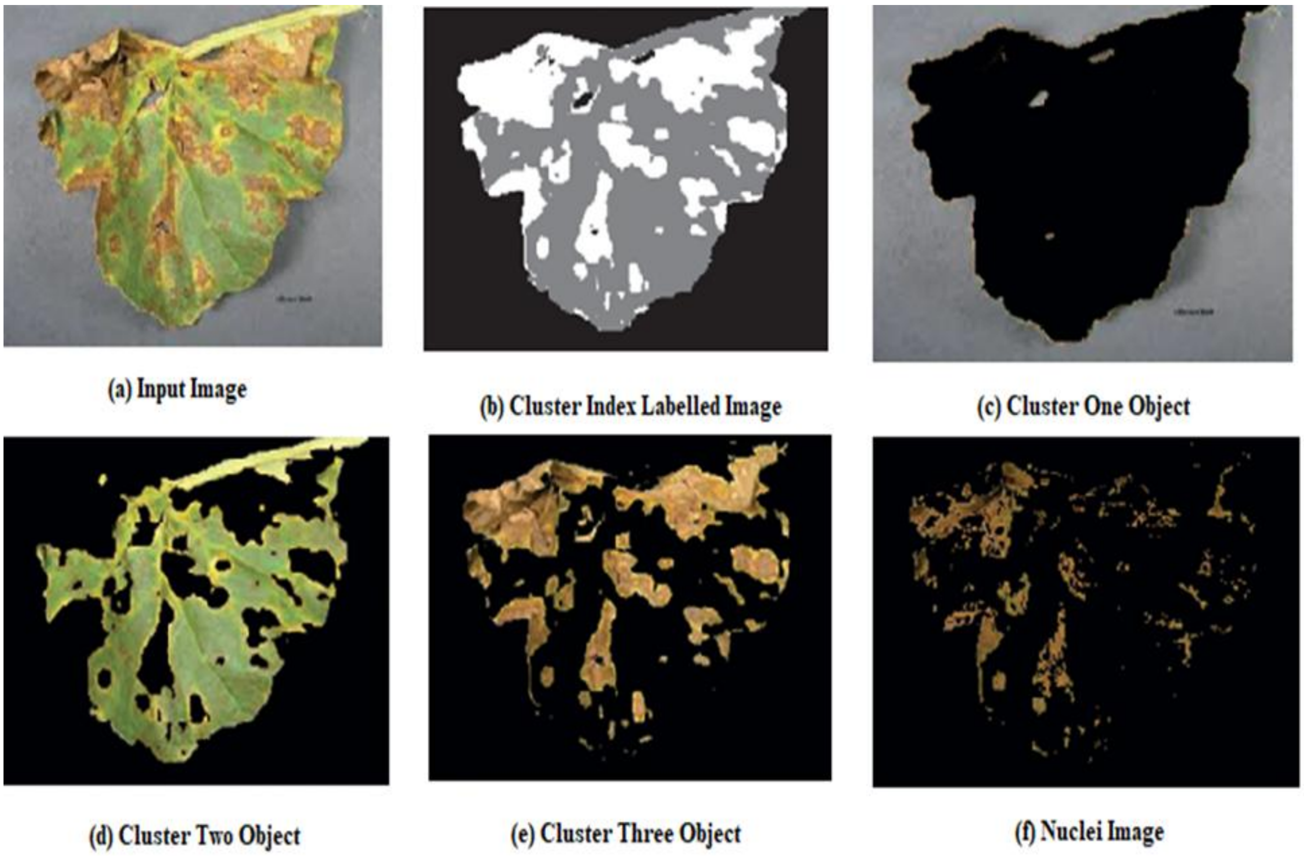


Fig 13: Various Segmented Images

. Target Class

Fig 14: Confusion matrix for rice plant

Fig.15: Confusion matrix for plant disease detection using Random Forest Bayesian optimized SVM classifier

7.Discussion

Deep learning advancements give the ability to expand studies and applications focused on the detection of plant diseases utilising digital images. Quick but also efficient approaches are required so that the relevant measures can be performed as soon as possible. The network model selected for the design of a classification model varies on whether the objective is to maximize or minimize. On the one aspect, the SqueezeNet approach is rapid if the situation necessitates frequent reconfiguration but is also too unstable to model. Alternative solutions for a significant level of accuracy include AlexNet as well as GoogLeNet. The work of Brahimi and others [19] was focused on classifying diseases for certain CNNs. In the little comparison measured by accuracy as well as utilizing similar learning transfer methods, InceptionV3 had the lower accuracy rate gained. Furthermore, AlexNet did not get comparable outcomes because AlexNet produced the best results while scoring significantly lower than the author's state. Particularly, model training takes far too long on a high-performance GPU. Lastly, the Plant leaf data set is unbalanced because some classes have larger images than others. If it is not trained correctly, this could be a problem and result in overfitting.

Deep learning has yielded good outcomes in a multitude of fields of research because of its exceptional ability to construct features completely autonomously. and without the assistance of humans. Several works in plant disease protection have suggested applying DL to identify and classify diseases of leaves, That is the reason we presented CNN with others techniques. The objective of developing a framework for those who can create and develop plant leaf disease classification mechanization of plant diseases, on the other side, ResNet50 is much more computationally costly in terms of processing time. Additionally, despite having the most depth, ResNet50, ResNet101, as well as InceptionV3 were not as efficient as more depth layers. Ultimately, it was suggested that different CNN be analysed in terms of time as well as effectiveness. AlexNet obtained a better outcome, whereas ShuffleNet achieved it in a smaller time, with an outcome of less than 1.53 percent. Similarly, in our research, here, we deploy AlexNet's activation layers to identify and pinpoint disease-related regions.

8. Conclusion

In the field of image identification, an architecture for deep learning called convolutional neural networks has been gaining fairly substantial popularity. LeNet, ShuffleNet, AlexNet, EffNet, &MobileNet are five convolutional neural network designs that were created and evaluated for the issue at hand. The latter's accuracy of 96.1% was the best of the three. All models were combined in committees and voted on using one of three voting methods: majority, hybrid feature-based random forest, or Bayesian optimum SVM. It was discovered that the best committee's entire set of trainable parameters was less compared to the canonical designs VGG16 & VGG19, which are often utilised in computer vision applications, in addition to their strong performance for the specified work. The recommended committees for the tasks typically performed well throughout the testing phase, indicating that they have a strong likelihood of success when used in settings with similar circumstances in the real world. Detailed comparative analysis with the results suggested here is challenging due to the multilabel classification of comparable publications in the research for the same database. But besides this challenge, it is stressed that viewing the issue as a task of binary classification can make the model proposed here easier to use and adapt for other situations while easing the burden on human experts to annotate examples. This is especially useful when using other interesting pictures of plants and diseases of various kinds that may affect them. This study's goals are to establish CNN's technical competence in spotting plant diseases and to open the door to agricultural AI solutions. This can be especially helpful in the agricultural industry, where it is necessary to do numerous field surveys of this kind. A variety of feature extractors for color and texture were used to evaluate the performance of the CNN, the Bayesian optimal SVM, as well as the Random Forest classifier with hybrid features. This study applies convolutional neural networks, which had the highest accuracy of 96.38%, to detect leaf blight in apple, maize, potato, tomato, & rice plants. [8]

Data Availability

All requested data will be provided.

Conflict of Interest

There are no conflicts of interest.

Acknowledgments

This work was not funded by any agency

References

[1] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, (2018) "Tomato crop disease classification using pre-trained deep learning algorithm," Procedia Computer Science, vol. 133, pp. 1040–1047.

[2] N. Kaur, "Plant leaf disease detection using ensemble classification and feature extraction, (2021)" Turkish Journal of Computer and Mathematics Education (TURCOMAT), vol. 12, no. 11, pp. 2339–2352.

[3] J. G. ArnalBarbedo, "Plant disease identification from individual lesions and spots using deep learning, (2019)" Biosystems Engineering, vol. 180, pp. 96–107.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, (2018) "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, vol. 25..

[5] C. Szegedy, W. Liu, Y. Jia, et al., (2015) "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9, Boston, MA,.

[6] L. Quan, H. Feng, Y. Lv et al., (2019) "Maize seedling detection under different growth stages and complex field environments based on an improved Faster R-CNN," Biosystems Engineering, vol. 184, pp. 1– 23.

[7] Y. Yu, K. Zhang, L. Yang, and D. Zhang, (2019) "Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN," Computers and Electronics in Agri., vol.163, Article ID 104846, 2019.9

[8] A. Fuentes, S. Yoon, S. Kim, and D. Park, (2022) "A robust deep learning-based detector for real-time tomato plant diseases and pests recognition," Sensors, vol. 17, no. 9, 2022 pages,.10

[9] I. M. Hanssen and M. Lapidot, (2012) "Major tomato viruses in the Mediterranean basin," Advances in Virus Research, Academic Press, vol. 84, pp. 31–66, 2012.11

[10] G. Thakre, A. R. More, and K. S. Gajakosh, (2017) "A study on real-time plant disease diagnosis system," International Journal of Advance Research, Ideas and Innovations in Technology, vol. 3, pp. 1118– 1124.

[11] M. Brahimi, K. Boukhalfa, and A. Moussaoui, (2017) "Deep learning for tomato diseases: classification and symptoms visualization," Applied Artificial Intelligence, vol. 31, no. 4, pp. 299–315.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, (2012) "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, vol. 25.

[13] C. Szegedy, W. Liu, Y. Jia, et al., (2015) "Going deeper with convolutions" in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9, Boston, M..

[14] M. Brahimi, K. Boukhalfa, and A. Moussaoui, (2017) "Deep learning for tomato diseases: classification and symptoms visualization," Applied Artificial Intelligence, vol. 31, no. 4, pp. 299–315, .

[15] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," Neurocomputing, vol. 267, pp. 378–384, 2017.

[16] S. Kagiwada, and H. Iyatomi, (2015) "Basic study of automated diagnosis of viral plant diseases using convolutional neural networks," Advances in Visual Computing, Springer, in Proceedings of the International Symposium on Visual Computing, pp. 638–64.

[17] I. Freeman, L. Roese-Koerner, and A. Kummert, (2018) an efficient structure for convolutional neural networks," in Proceedings of the 25th IEEE International Conference on Image Processing (ICIP), pp. 6–10, IEEE, Athens, Greece.

[18] F. N. Iandola, S. Han, M. W. Mickiewicz, K. Ashraf, W. J. Dally, and K. Kreutzer, (2016) "Squeeze Net: Alex Net-level accuracy with 50x fewer parameters and< 0.5 MB model size," in Proceedings of the International Conference on Learning Representations, Toulon, France..

[19] G K. He, X. Zhang, S. Ren, and J. Sun, (2016) "Deep residual learning for image recognition," in Proceedings of the IEEE conference Journal of Food Quality 15 on computer vision and pattern recognition, pp. 770–778, Las Vegas, NV, USA, June, 2016.

[20] D. Hughes and M. Salathe, (2015) "An open access repository of ´ images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing," 2015, https://arxiv.org/abs/1511.08060

[21] D. R. Hammou and M. Boubaker, (2022) "Tomato plant disease detection and classification using convolutional neural network architectures technologies," in Networking, Intelligent Systems and Security, pp. 33–44, Springer, Singapore.

[22] R. Khan, N. Tyagi, and N. Chauhan, (2021) "Safety of food and food warehouse using VIBHISHAN," in Journal of Food Quality, A. Durazzo, Ed., vol. 2021, Hindawi Ltd, Journal of Food Quality, pp. 1–12.

[23] Y.Toda and F. Okura, (2019) How Convolutional Neural Networks Diagnose Plant Disease, Plant Phenome's, Berlin, Article ID 9237136.

[24] P. Taraba, (2013) "Linear regression on a set of selected templates from a pool of randomly generated templates," Machine Learning with Applications, vol. 6, Article ID 100126, 2013

[25] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, (2018) "A guide to convolutional neural networks for computer vision," Synthesis Lectures on Computer Vision, vol. 8, no. 1, pp. 1–207.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, (2012) "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, vol. 25,

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, (1998) "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998

[28] R. Khan, S. Kumar, N. Dhingra, and N. Bhati, (2021) "'e use of different image recognition techniques in food safety: a study," in Journal of Food Quality, I. Tomasevic, Ed., vol. 2021, Hindawi Ltd, Journal of Food Quality, pp. 1–10.

[29] X. Zhang, X. Zhou, M. Lin, and J. Sun, (2018) "Shuffle net: an extremely efficient convolutional neural network for mobile devices," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6848–6856, Nashville, USA, .

[30] I. Freeman, L. Roese-Koerner, and A. Kummert, (2018) "October. Eff net: an efficient structure for convolutional neural networks," in Proceedings of the 25th IEEE International Conference on Image Processing (ICIP), pp. 6–10, IEEE, Athens, Greece.

[31] R. Sujatha, J. M. Chatterjee, N. Jhanjhi, and S. N. Brohi, (2021) "Performance of deep learning vs machine learning in plant leaf disease detection," Microprocessors and Microsystems, vol. 80, p. 103615, .

[32] J. Snoek, H. Larochelle, and R. P. Adams, (2012) "Practical Bayesian optimization of machine learning algorithms," Advances in Neural Information Processing Systems, vol. 25, pp. 2951–2959,

[33] N. Keen, (2005) Color Moments, pp. 3–6, School Of Informatics, University Of Edinburgh, Edinburgh, .

[34] K. Khairnar and R. Dagade, (2014) "Disease detection and diagnosis on the plant using image processing A review," International Journal of Computer Application, vol. 108, no. 13, pp. 36–38, .

[35] H. Rizk, (2017) "Automated early plant disease detection and grading system: development implementation," Available online at: https://fount.aucegypt.edu/cgi/viewcontent.cgi? article=1173&context=etds, 2017

[36] J. Kennedy and R. C. Eberhart, (1997). "A discrete binary version of the particle swarm optimization, "Computational cybernetics and simulation, vol. 5, pp. 4104–4108, 1997.

[37] L. Breiman, "Random forests, (June 2001)." Machine Learning, vol. 45, no. 1, pp. 5–32,.

[38] M.A. Khanesar,M Teshnebiah, and M.A. Shoorehdeli, "(June 2007). A novel binary particle swarm optimization "in proceeding of the 2007Mediterranean conference on control a automation, pp1-6,IEEE,Athene,Grece