Classification of Potato Leaf Diseases using Deep Convolutional Neural Networks

Aditya Kumar Singh, Nishant Sharma, Tonmoy Baruah, Soumya Sen
Project Guide: *Dr. R. Murugan*

Department of Electronics and Communication Engineering
National Institute of Technology, Silchar

Contents

List of Figures

List of Tables

1 Abstract

In agricultural ecosystems, the maximization of crop yields is heavily dependent on disease management. The recent advancements in technology have aimed to suggest better alternatives for plant disease diagnosis to minimize the economic and aesthetic damage caused by plant diseases. This paper discusses methods based on the principles of automated leaf-based disease diagnosis using computer vision. The primary steps involved are: Data Acquisition, Image pre-processing and augmentation, hyperparameter optimization using metaheuristics like the Grey Wolf Optimization(GWO) algorithm, Whale Optimization Algorithm(WOA) and Slime Mould Algorithm(SMA), and training and testing the CNN model based on the hyperparameters obtained. The experimental results obtained using 3600 images each of Early Blight, Healthy and Late Blight, and the classification accuracy along with detailed performance analysis are the main areas of discussion of this paper.

Keywords Leaf-based Disease Diagnosis, Potato Blight, Convolutional Neural Network, Grey Wolf Optimizer, Whale Optimization Algorithm, Slime Mould Algorithm

2 Introduction

Potatoes are considered to be one of the most important food crops worldwide. They are known to be a good source of carbohydrates and fiber, also containing nutrients such as Potassium, Magnesium, Vitamin C and is rich in antioxidants. In India alone, potatoes are cultivated on a large scale. The production quantities roughly add up to 52 million tonnes. Considering the widespread dependence on the crop, it becomes important to study and diagnose different types of diseases which can hamper the production quality and quantity. Two common leaf diseases in potato plants are Early Blight and Late Blight. Early Blight is caused by the fungal pathogen *Alternaria Solani* while Late Blight is caused by the pathogen *Phytophthora Infestans*. Both of these diseases if left unchecked and untreated can cause substantial loss in production. Traditional and the most practiced approach for detection of disease in plants is continuous onsite monitoring performed by naked eye observation by the farmers and/or the local experts. This approach on such a large scale proves to be unfeasible and impractical while also consuming a lot of time and resources. Hence, in order to overcome this problem, use of emerging technologies to automate the disease detection process is a need of the hour.

Early and Late Blight symptoms can be visually identified as characteristic spots of discoloration/pigmentation in the leaves of the infected plants. This paper is a performance analysis of various Convolutional Neural Networks (CNN) pre-trained on a dataset of images of Healthy, Early Blight infected and Late Blight infected plants. Deep Learning algorithms such as this have seen widespread application in different sectors as well. The CNNs leverage the recent advancements in image segmentation for enhanced classification of the images. Further, processes such as feature extraction, selection and classification are done in the mid and deep layers of the deep neural architecture itself. In this paper, OptCoNet [1] has been used as the primary deep learning architecture along with other pre-trained networks such as GoogLeNet and ResNet50.

CNNs have a variety of hyperparameters that have a considerable effect on the network's performance. Hyperparameters are the variables that determine network training options and how the network is trained. Using manual trial and error methods to determine the optimal set of values to be used as the hyperparameters for any CNN network is challenging as well as computationally expensive. Additionally, hyperparameters are found to be dataset and network specific, meaning hyperparameters that adapt well for a particular dataset or network may not adapt well to another dataset and/or network. These reasons bring up the need for hyperparameter optimization. In this paper, the use of metaheuristics such as Grey Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA) and Slime Mould Algorithm (SMA) have been carried out for this purpose. GWO is based on the behavioral and hunting patterns of grey wolves. It tests different sets of values of hyperparameters and returns the best most suitable set to be used for training. WOA is based on the spiral bubble-net feeding maneuver of humpback whales, while SMA models the adaptive search strategy of the slime mould in nature. It is worth noting that any other optimization algorithm could have been chosen for this purpose. As the No Free Lunch (NFL) [2] theorem proves that no algorithm is best suited for solving all optimization problems, or in other words, all optimization algorithms perform equally well when their performance is averaged across all possible problems.

Broadly, the methodology consists of the following major steps: Data Acquisition, Image pre-processing and augmentation, hyperparameter optimization using metaheuristics like the Grey Wolf Optimization(GWO) algorithm, Whale Optimization Algorithm(WOA) and Slime Mould Algorithm(SMA), training and testing the CNN model based on the hyperparameters obtained.

This paper is organized as follows: Section III lists the literature survey of the prominent work done in the concerned field. Section IV discusses Hyperparameter Optimization and the algorithms used in this paper. Section V elucidates the methodology and a brief description of the steps taken to obtain the necessary results. Section VI presents a brief comparison of the experimental results and performance of the networks. Section VII concludes the paper and provides scope for future work.

3 Literature Review

This section briefly covers the ongoing attempts at leaf-based classification of healthy and diseased plants, primarily focusing on potato leaves. Works [3–5] proposed mechanisms to detect diseased leaves using Support Vector Machines.

Ramya et al. [3] converted the images to grayscale and performed segmentation using Otsu's threshold algorithm [6]. Then, they applied a mean filter to improve the image quality and extracted features such as area and perimeter from the binary segmented images. From the resulting data, affected leaves were classified using SVM. Prakash et al. [4] proposed a four-part framework consisting of Image pre-processing, segmentation(using K-means clustering), Feature Extraction, and classification. Finally, the classification amongst healthy/diseased was carried out using an SVM classifier. Monzurul Islam et al. [5] integrated image processing and machine learning algorithms on the PlantVillage dataset. Their demonstration included segmentation of images, extraction of Region of Interest(ROI) followed by training a multi-class SVM classifier with the obtained significant properties which are distinguishable. Both [4] and [5] used the Gray Level Co-occurrence Matrix (GLCM) for extracting statistical texture features like contrast, correlation, energy and homogeneity.

Iqbal et al. [7] demonstrated the importance of image segmentation and feature extraction before classification. During image processing, the images were converted into HSV colour space and then thresholded accordingly to separate the region of interest(ROI) from the image. They used multiple classifiers such as Naive Bayes, Linear Discriminant Analysis, Random forest, etc. for performing multi-class classification and compared the results. Random Forest (RF) algorithm gave the best results with 97% accuracy.

Most of the literature reviews mentioned so far have used SVM as the main judgement tool along with some form of image preprocessing. With current advances in technology, it has been observed that a well trained artificial neural net also gives high classification accuracy.

Oppenheim et al. [8] proposed the use of CNN via a partially modified VGG architecture for multiclass classification of diseases in potato tubers. They trained the model for different volumes of training and test data and compared the accuracies and errors for each case. It was observed that for a 90%-10% to 70%-30% training-test volume division the accuracies were almost the same. While potato tubers also develop blight symptoms, using leaf-based early detection has a clear advantage over tuber-based detection as appropriate steps can be taken to reduce the damage caused and hence, increase productivity.

Ananthi et al. [9] proposed a system that uses CNN to identify potato leaves into healthy, early blight, late blight classes. They discuss the Contrast-Limited Adaptive Histogram Equalization (CLAHE) and Gaussian Blur algorithms which play a major role in performing feature extraction on the pre-processed image, and finally use a CNN with five hidden layers and eight iterations (epoch), to obtain an accuracy of 98.54% on a dataset containing 500 images. Rakesh. S et al. [10] demonstrated the use of deep learning models for plant disease diagnosis emphasising on Deep Transfer Learning with Optimal Kernel Extreme Learning Machine(DTL-OKELM). Their proposed method consisted of a thresholding based technique where the DTL based inception v3 model was used as a feature extractor and the OKELM model was used as a classifier. With slight pre-processing for noise removal and contrast management their proposed model obtained an accuracy of 98.58% for citrus and 99.32% for tomato. Trong-Yen Lee et al. [11] emphasized the use of CNN architecture with Adam optimizer over SVM based judgment for leaf-based disease detection. They used Gaussian filtering for noise removal, RGB normalization to reduce light impact followed by extraction of the ROI and reconstruction of the image to be used for training. They tested pre-trained networks such as VGG16 and VGG19 each giving a training accuracy of 98.15% and 48.55%, along with the proposed CNN which had a training accuracy of 99.53%.

Darwish et al. [12] proposed an ensemble model of two pre-trained convolutional neural networks (CNNs) namely VGG16 and VGG19 to get a better predictive performance of healthy/diseased classification compared to a single model. The hyperparameters of each of the networks were optimized using Orthogonal Learning Particles Swarm Optimization (OLPSO). OLPSO algorithm optimized the minibatch size and the dropout rate of the CNN architecture. The accuracy achieved by the optimized VGG16, optimized VGG19, and the ensemble model was 97.9%, 97.7% and 98.2% respectively. The results were then compared with other pre-trained networks such as InceptionV3 and Xception both yielding accuracies of 96.6% and 96.5% respectively.

Table 1: Summary of state of the art CNN methods for leaf Disease Detection

Reference	Task	No. of Images	Method	Accuracy	Recall	Precision	F1 Score
Ramya et. al [3]	Various leaves	169	SVM	92	-	-	-
Prakash et. al [4]	Various leaves	70	SVM	90	-	-	-
Ishlam et. al [5]	Potato Leaf	300	SVM	95	95	95	95
Iqbal et. al [7]	Potato Leaf	450	Random Forest*	97	97	97	97
Oppenheim et. al [8]	Tubers	2465	Modified VGG	95.85	-	-	-
Ananthi et. al [9]	Potato Leaf	500	CNN	98.54	-	-	-
Rakesh et. al [10]	Citrus Tomato	609 5452	DTL-OKELM	98.58 99.32	98.77 99.43	98.44 99.21	98.32 99.05
Lee et. al [11]	Various leaves		CNN	99.53	-	-	-
Darwish et. al [12]	Various leaves	12332	VGG16,VGG19	98.2	97	98	97

4 Hyperparameter Optimization

Hyperparameters play a crucial role in determining the accuracy and convergence of the CNN. Selecting the network's hyperparameters is essential and is highly specific to the application for which the CNN is used. The **learning rate**, **number of epochs**, **momentum**, and **regularization coefficient** are the most common CNN training hyperparameters. The learning rate controls the gradient descent algorithm's speed, and the momentum controls the influence of the update of previous weights on the update of current weights. The number of epochs determines the number of times the learning algorithm will update the network parameters according to the training dataset. Regularization overcomes the issue of overfitting in the network. Therefore, to address all these settings, optimizing these hyperparameters is required to help the network yield the most accurate results.

For tuning the hyperparameters of the CNN Network, three meta-heuristic optimization algorithms have been implemented to maximize *accuracy* as the objective function. The Increasing Complexity in optimization problems in recent decades has led to the application of Meta-heuristic Algorithms in many disciplines because of higher performance and lower requirement of computing capacity. Meta-heuristic optimization algorithms are becoming more and more popular in engineering applications because they: rely on rather simple concepts and are easy to implement; do not require gradient information; can bypass local optima, and thus can be utilized in a wide range of problems covering different disciplines.

4.1 Grey Wolf Optimization

The Grey Wolf Optimizer (GWO) [13], a recently developed swarm intelligence algorithm, has proven to be a reliable optimization algorithm compared to conventional evolutionary and swarm-based algorithms. The grey wolf belongs to the Canidae family and is considered a high-level predator and dwells at the top of the food chain. They live in a pack that comprises 5–12 maintaining an exact predominance order- the pack is driven by alphas and trailed by betas, the subordinate wolves who are mindful to help the alpha. The beta wolf strengthens the alpha's orders all through the pack and offers input to the alpha. In the interim, the lowest rung among the gray wolves is the omega, who generally assumes the scapegoat's job. They are the last wolves allowed to eat from the prey. On the off chance that a wolf is not alpha, beta, or omega, the individual in question is known as a delta. Delta wolves act as scouts, sentinels, seniors, trackers, and guardians. The motivation for proposing the use of GWO algorithm for Potato leaf-disease diagnosis is twofold: The GWO has been applied to problems from varying research fields: such as feature selection, economic load dispatch problems, and flow scheduling problems. Additionally, the GWO algorithm also benefits from avoiding high local optima, which leads to avoidance of overlapping features in the problem of feature selection, which is essential to the heart of the proposed CNN based methods.

4.2 The Whale Optimization Algorithm

The Whale Optimization Algorithm (WOA) [14] mathematically models the spiral bubble-net feeding maneuver of humpback whales in order to perform optimizations on a fitness function. Whales, considered to be the biggest mammals in the world are found in 7 different main species: killer, Minke, Sei, humpback, right, finback, and blue. Mostly considered as predators, they have common cells in certain areas of their brains similar to those of humans called spindle cells [15]. These cells, which are responsible for judgment, emotions, and social behaviors in humans, exist in almost twice the number for Whales, which has led to studies investigating whales' thinking, learning, judgement and communication techniques. An interesting aspect of the humpback whales is their special *bubble-net feeding method* [16] hunting method. It has been observed that this foraging is done by creating distinctive bubbles along a circle or 9-shaped path. In 2011, Goldbogen et al. [17] investigated this behavior utilizing tag sensors by capturing 300 tag-derived bubble-net feeding events of 9 individual humpback whales and found two maneuvers associated with bubble and named them *upward-spirals* and *doubleloops*. In the former maneuver, humpback whales dive around 12 m down and then start to create bubbles in a spiral shape around the prey and swim up toward the surface. The later maneuver includes three different stages: coral loop, lobtail, and capture loop. It is worth mentioning here that bubble-net feeding is a unique behavior that can only be observed in humpback whales.

Algorithm 1: GWO Algorithm

Input: Number of Search Agents, Fitness Function
Output: Best Search agent X_α

Initialize the grey wolf population $X_i (i = 1, 2, \ldots, n)$
Initialize coefficient vectors a, A, and C
Calculate the fitness of each search agent
X_α = the best search agent
X_β = the second best search agent
X_δ = the third best search agent
while $t <$ *Max number of iterations* **do**
    for *each search agent* **do**
        Update the position of the *current search agent* according to GWO update equations [13]
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update X_α, X_β, and X_δ
    $t = t + 1$
end while
return X_α

Humpback whales can recognize the location of prey and encircle them. Since the position of the optimal design in the search space is not known a-priori, the WOA algorithm assumes that the current best candidate solution is the target prey or is close to the optimum. After the best search agent is defined, the other search agents will hence try to update their positions towards the best search agent. The same concept can be extended to an $n-$dimensional search space, with the extension that the search agents here move in hyper-cubes around the best solution obtained so far.

4.3 The Slime Mould Algorithm

The Slime Mould Algorithm (SMA) [18], a recently developed MA has proven to be computationally efficient as compared to deterministic algorithms in various optimization algorithms. The SMA stimulates the behaviours and changes of slime mould *Physarum polycephalum* during its quest for find food and doesn't represent its complete lifecycle. The algorithm mainly consists of the following components: initialization, fitness evaluation, sorting ,weight update and location update. T. Latty et al. [19] presented that slime mould has the ability of making foraging arrangements based on optimization theory. The slime mould can approach food according to the odour in the air which includes searching individuals to search in all possible directions near the optimal solution, thus simulating the circular sector structure of slime mould when approaching food. The adaptive search strategy of the slime mould is based on dynamic adjustment of search patterns to the quality of food. In case of high quality food source, it uses a region-limited search method whereas in case of low quality it leaves the food source for an alternative source in the region. The unique mathematical model in the SMA relies on the use of weights in SMA to simulate the positive and negative feedback generated by slime mould during foraging. In contrast to deterministic algorithms, where the algorithm sinks to a local optimum during the later stages, the random factors in SMA makes the algorithm search for all fit solutions in the search space thus avoiding the local optimum.

Algorithm 2: WOA Algorithm

Initialize the whales population $X_i (i = 1, 2, \ldots, n)$
Calculate the fitness of each *Search Agent*
X^* = the best *Search Agent*
Initialize coefficient vectors a, A, C, l and p
while $t <$ *Max number of iterations* **do**
  for *each Search Agent* **do**
    Update a, A, C, l, and p
    if $p < 0.5$ **then**
      if $|A| < 1$ **then**
        Update the position of the current *Search Agentt* as:

$$D = |C \cdot X^*(t) - X(t)|$$
$$X(t+1) = X^*(t) - A \cdot D$$

      else
        Select a random search agent X_{random}
        Update the position of the current *Search Agent* as:

$$D = |C \cdot X_{random} - X|$$
$$X(t+1) = X_{random} - A \cdot D$$

      end if
    else
      Update the position of the current *Search Agent* as:

$$D' = |X^*(t) - X(t)|$$
$$X(t+1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t)$$

    end if
  end for
  Check if any *Search Agent* goes beyond the search space and amend it
  Calculate the fitness of all *Search Agents*
  Update X^*
  $t = t + 1$
end while
return X^*

Algorithm 3: SMA Algorithm

Initialize the parameters *pop_size*, $Max\_iteration$
Initialize the positions of slime mould $X_i (i = 1, 2, \ldots, n)$
while $t < Max\_iteration$ **do**
  Calculate the fitness of all slime mould
  update bestFitness, X_b
  Calculate: $\overline{W(\text{SmellIndex}(t))} = \left\{ \begin{array}{ll} 1 + r \cdot \log\left(\frac{bF - S(i)}{bF - wF} + 1\right), & \text{condition} \\ 1 - r \cdot \log\left(\frac{bF - S(i)}{bF - wF} + 1\right), & \text{others} \end{array} \right\}$
  for *each Search Portion* **do**
    Update p, vb, vc
    Update positions as: $\overrightarrow{X^*} = \begin{cases} \text{rand} \cdot (UB - LB) + LB, \ \text{rand} < z \\ \overline{X_b(t)} + \overrightarrow{vb} \cdot \left(W \cdot \overrightarrow{X_A(t)} - \overline{X_B(t)}\right), r < p \\ \overrightarrow{vc} \cdot \overrightarrow{X(t)}, r \geq p \end{cases}$
  end for
  $t = t + 1$
end while
return *bestFitness*, X_b

5 Methods

5.1 Dataset

The potato leaf images used for training and testing the networks were collected from online repositories [20]. A subset containing 3600 images was randomly chosen from the main dataset for training.

In order to monitor the model's accuracy during the training phase, a validation set was created where 30% of the total images were set aside. The remaining 70% of the images were used for training. Table 2 shows the number of samples used for training and validation under each class.

Category	No. of Images	
	Training Set	Validation Set
Healthy	720	360
Early Blight	720	360
Late Blight	720	360

Table 2: Number of samples in each class used in training and validation set

Before deep learning, image standardization was performed. The images were downsized to 224 by 224 and data augmentation was performed to reduce the chance of overfitting in the learning process. The values corresponding to each data augmentation technique is listed in Table 3. Figure 1 shows the original images of the dataset and the Figure 2 shows the same images after the augmentation.

Data Augmentation Technique	Value
Rotation	[-30,30]
X Translation	[-5,5]
Y Translation	[-5,5]
Horizontal Flip	True

Table 3: Data Augmentation Techniques and corresponding values

(a) Early Blight

(b) Healthy

(c) Late Blight

Figure 1: Original Images

(a) Early Blight (b) Healthy (c) Late Blight

Figure 2: Augmented Images

5.2 WorkFlow

5.2.1 Convolutional Neural Networks

CNN architecture, as illustrated in figure 3 primarily comprises of two blocks, the Convolutional and the Classifier sections. The Convolutional block mainly extracts the various information of the image and is composed of Convolutional Layers, Pooling Layers, Activation Layer and Batch Normalization Layers. The Classifier block predicts the class of the input image based on the features extracted in the previous layers and is composed of the Fully Connected layer and the Softmax Layer.

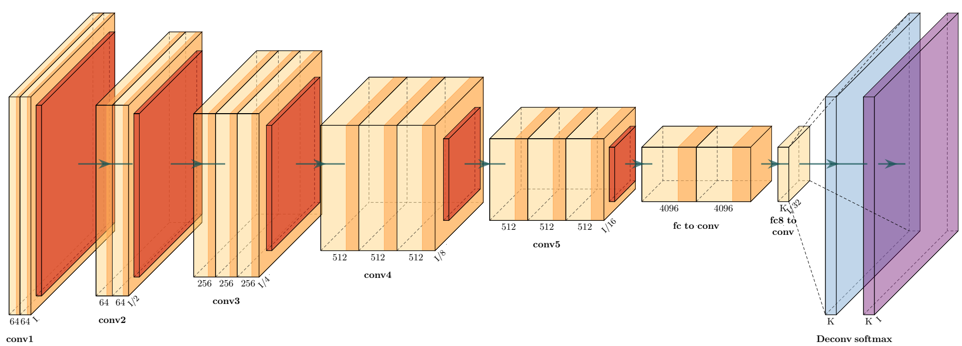


Figure 3: CNN Architecture

- **Convolutional Layer(CL):** Also known as the feature extraction layer, it is the first layer to extract certain patterns or features from an input image using a group of filters. The output of the convolution between the image and the filter or kernel is termed as the feature map which gives the characteristics information about the input image such as the corners and edges. Later, this feature map is fed to further layers to learn several other features of the input image.
- **Pooling Layer(PL):** A CL is followed by a pooling layer. This layer is mainly used to reduce the dimensionality of each feature map without losing important information. Reducing the dimensions of the map helps in decreasing the number of parameters and computation in the network. Common examples of pooling techniques are Max Pooling, Average Pooling, Sum Pooling.
- **Batch Normalization Layer(BNL):** Instead of just normalizing the input to the network, output of each layer is normalized for each batch size. Here normalization has to be done separately for each dimension over the mini-batches, and not altogether with all dimensions. BNL helps train networks

faster as the network converges more quickly. It also allows higher learning rates and initializing the weights becomes easier.

- **Activation Layer(AL):** The activation function or transfer function in a neural network defines how the weighted sum of the input is transformed into the output. Some activation functions are non-linear and these are also used to add non-linearity. The most popular activation functions are sigmoid, tanh, and Rectified Linear Unit (ReLU). Among them, ReLU is most widely used for deeper networks, as sigmoid and tanh cause vanishing gradient problem but ReLu has a constant gradient for positive input.

- **Fully Connected Layer(FCL):** From this layer, the classification process starts. The feature matrix generated in previous layers is flattened to generate a vector and is fed into the FCL. Here every neuron in one layer is connected to every neuron in the next layer. The major drawback of a FCL is that it includes a lot of parameters that need complex computations and can cause overfitting in the training dataset. This problem is overcome by randomly dropping a subset of neurons and their connections. FCLs are followed by a dropout layer except for the last FCL, which is followed by the classification layer.

- **Softmax Layer(SL):** The activation function applied to the output of the last FCL is completely different from those applied to the other fully connected layers where the sigmoid or softmax function is generally used. For multiclass classification, a softmax function is used to turn logits into probabilities by taking the exponents of each output and then normalizing each number by the sum of those exponents so the entire output vector adds up to one.

5.2.2 OptCoNet

The OptCoNet architecture consists of several deep layers which include many Convolutional Layers(CL), Max Pooling layers(MPL) and Fully connected layers(FCL). The architecture workflow is divided into two submodules: feature extraction and classification. Feature extraction is performed using several layers, including a CL followed by a MPL. Classification consists of a FCL and a softmax layer to classify the input images into a particular class. Training the CNN involves adjusting the hyperparameters of the network and individual adjusting and training is computationally expensive. Therefore, optimization algorithms such as the Grey Wolf Optimizer(GWO), Whale Optimization Algorithm(WOA) and Slime-Mould Algorithm(SMA) are used to optimise the hyperparameters.

5.2.3 Pre-trained CNNs

ResNet50 ResNet was the winner of the ILSVRC 2015 classification with an error rate of 3.57% using an ensemble model. The core idea of ResNet is the identity shortcut connection which gives non-linear layers. It overcomes the accuracy saturation and degradation problem with an increase in depth of the network due to vanishing gradient, using the Deep Residual learning framework. It uses mostly $3 * 3$ filters with stride 2. It uses a global average pooling layer and a fully connected layer with softmax in the end for classification.

GoogleNet GoogleNet was the winner of the 2014 ILSVRC champion model with an error rate of 6.7%. This model introduced the idea to increase the depth(Number of layers) or the width (Number of layers of core or neurons) of the model to get high accuracy using the inception module, which has different sizes and types of convolution for the same input. It contains the $1 * 1$ convolution at the middle of the network which is used as a dimension reduction module to reduce the computation, so that depth and width can be increased. It uses global average pooling at the end of the network instead of using a fully connected layer.

6 Experiment

6.1 Implementation Details

The algorithms were implemented in MATLAB 2021a and executed using Windows 10 Home with a 4 GB RAM Nvidia GPU.

6.1.1 Training

The dataset is split randomly into training and test subsets, 70% of the total set was used for training and the rest of the 30% was used for testing the neural networks. All the images are resized to 224x224x3 using data augmentation. The training set images are fed into the untrained CNN. The output generated is used to calculate the value of cost function which is to be minimized. Backpropagation using Stochastic Gradient Descent with Momentum was used for minimizing the cost function. During backpropagation, the internal weights and biases of the network are optimized so as to reduce the difference between actual and predicted values.

6.1.2 Testing

First, the images are resized to 224x224x3 using data augmentation and used accordingly for training the neural networks. Next, the test images were fed into the trained CNN where all the internal weights and biases are already optimized. The first block of the CNN extracts the image features while the second block classifies them into the appropriate class.

Table 4: Evaluation Metrics

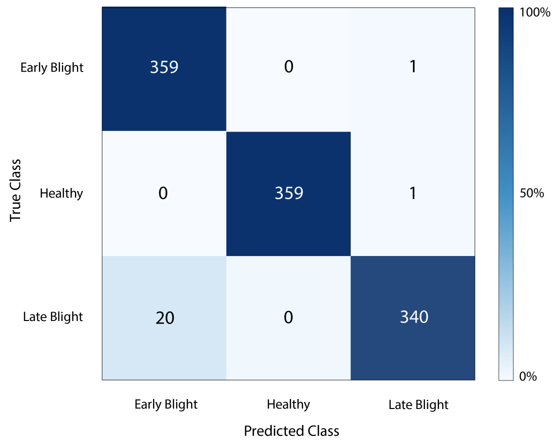
Parameters	Formula
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity or recall	$\frac{TP}{TP+FN}$
Precision	$\frac{TP}{TP+FP}$
F1 Score	$2\frac{\text{Precision x Recall}}{\text{Precision + Recall}}$
True Positive Rate	$\frac{TP}{TP+FN}$
False Positive Rate	$\frac{FP}{FP+TN}$

6.2 Performance Indicators and Evaluation Metrics

The proposed method is validated using the performance metrics of accuracy, sensitivity, precision, and F1-score, and receiver operating characteristic(ROC) analysis. The performance metrics equations are summarized in Table 4, where **TP** indicates True Positive; **TN** indicates True Negative; **FP** indicates False Positive and FN indicates False Negative. Accuracy evaluates the ability of the classifier to differentiate between healthy, early blight and late blight classes. A **TP** is where the model adequately predicts a positive case. Therefore, a **TN** is where the model viably predicts a negative instance. An **FP** is where the model erroneously predicts a positive case, and a **FN** is where the model mistakenly predicts a negative situation. Sensitivity gauges the proportion of correctly classified positive. instances. Precision measures the fraction of relevant cases among the retrieved cases and is also known as the positive predictive value. The F1-score measures a test's accuracy and is defined as the weighted harmonic mean of the test's precision and recall. The receiver operating characteristic (ROC) curve is the characteristic representation of the classification method performance executed for all values and is drawn as the relationship between $1-$specificity and sensitivity.

6.3 Experimental Results

The generated ROC and Confusion matrix for every single model are given in figures 4 and 5 respectively. For a total of 1080 testing images (360 per class), the pre-trained networks: GoogLeNet and ResNet50 achieved testing accuracies of 98.15% and 99.72% respectively. Out of the misclassified images, majority were from the late blight category, with very few from either of Healthy/Early Blight categories. OptCoNet achieved an accuracy of 95.09%, while it's optimized variants with GWO, WOA, and SMA performed better, reaching testing accuracies of 97.41%, 97.59% and 97.96% respectively.

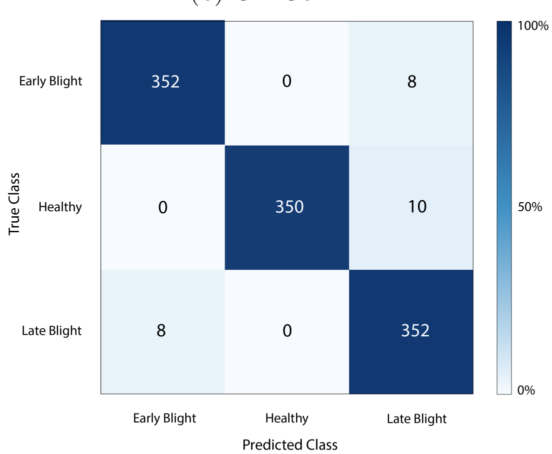


(A) GoogLeNet

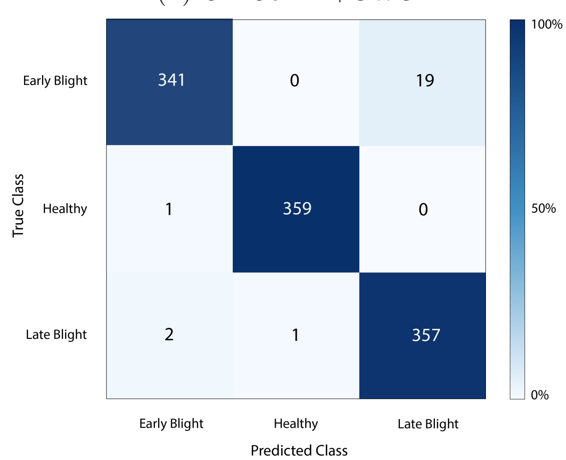
(B) ResNet50

(C) OptCoNet

(D) OptCoNet+GWO

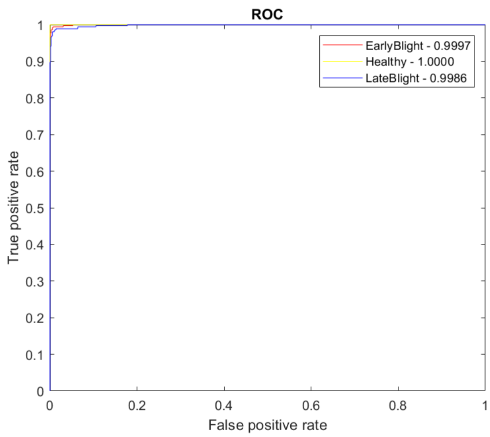


(E) OptCoNet+SMA

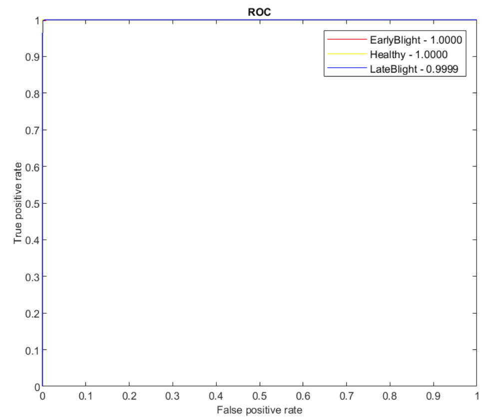


(F) OptCoNet+WOA

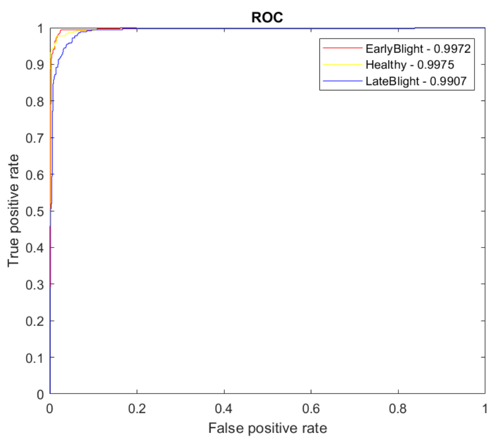
Figure 4: Confusion Matrices of the different networks and optimizers



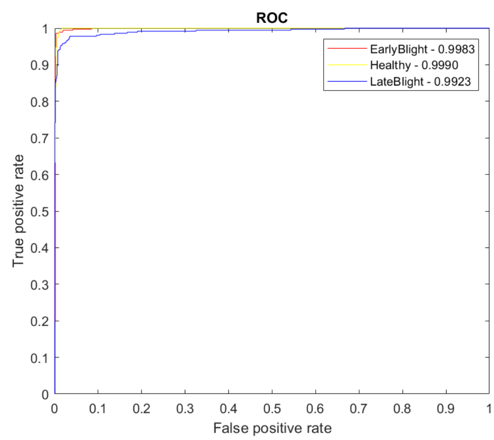
(A) GoogleNet



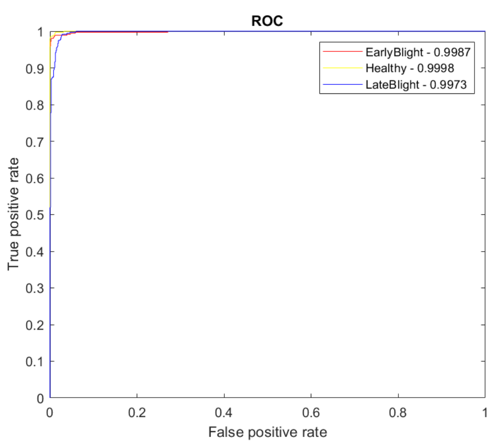
(B) ResNet50



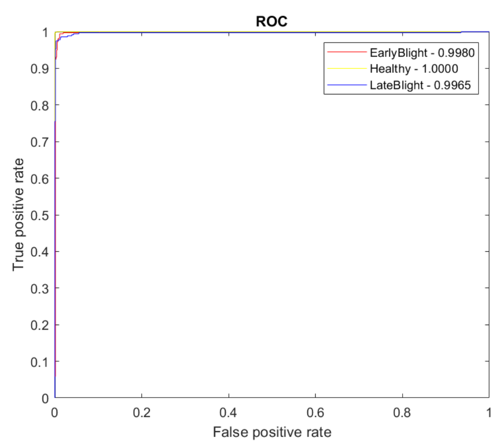
(C) OptCoNet



(D) OptCoNet+GWO



(E) OptCoNet+SMA



(F) OptCoNet+WOA

Figure 5: Receiver Operating Characteristic (ROC) curves of different networks and optimizers

6.4 Performance Analysis

For training the DL networks, 70% of the data are used; the remaining 30% of the data are used for testing. A comparison of all these networks in terms of the metrics accuracy, precision, recall, and F1-score is given in Table 5. Comparisons in terms of ROC curves and confusion matrixes are illustrated in figures 4 and 5 respectively. The results of a comparative study indicate that OptCoNet works best when optimized by WOA, with its accuracy(98.58) being at par with state-of-the-art methods shown in Table 1, as well as with pre-trained networks such as GoogLeNet and ResNet50, which achieved testing accuracies of 98.15% and 99.72% respectively.

An overall comparison of the Area Under Curve(AUC) for different networks is given in Table 6. For OptCoNet with an accuracy of 95.09%, AUC was found to be 99.72% for Early Blight, 99.75% for Healthy and 99.07% for Late Blight. But with the help of hyperparameter optimization, the accuracy of the OptCoNet increased substantially. With GWO as an optimizer, OptCoNet produced an accuracy of 97.14% which resulted in AUC values for Early Blight, Healthy and Late Blight being 99.87%, 99.98% and 99.73% respectively. Similarly, SMA with an accuracy of 97.59% gave AUC values for Early Blight, Healthy and Late Blight as 99.87%, 99.98% and 99.73% respectively. Finally, using WOA to optimize hyperparameters increased the overall accuracy to 97.96% and gave the AUC value for Early Blight, Healthy and Late Blight as 99.80%, 100.0% and 99.65% respectively, thus achieving the best results amongst the three optimizers. On the other hand, pre-trained networks such as GoogLeNet gave AUC values for Early Blight,Healthy and Late Blight gave value as 99.97%,100.00% and 99.86 respectively and similarly ResNet gave the AUC values for Early Blight,Healthy and Late Blight as 100.00%, 100.00% and 99.99% respectively. Comparing these two pre-trained networks' accuracy values with the best optimized OptCoNet network, ResNet50 gave the highest accuracy.

Table 5: Performance Metrics of Different CNN's

Network	Accuracy	Precision	Recall	F1 Score
ResNet50	99.45	0.99	0.99	0.99
Googlenet	98.64	0.98	0.98	0.98
Optconet	96.98	0.95	0.95	0.96
Optconet+GWO	98.09	0.97	0.97	0.97
Optconet+SMA	98.39	0.98	0.98	0.98
Optconet+WOA	98.58	0.98	0.98	0.98

Table 6: Area under curve(AUC) of the various networks and optimizers

	Early Blight	Healthy	Late Blight
GoogleNet	0.9997	1.0000	0.9986
ResNet50	1.0000	1.0000	0.9999
OptCoNet	0.9972	0.9975	0.9907
OptCoNet+GWO	0.9983	0.9990	0.9923
OptCoNet+SMA	0.9987	0.9998	0.9973
OptCoNet+WOA	**0.9980**	**1.0000**	**0.9965**

7 Conclusion

This work analyzes the performance of OptCoNet, GoogLeNet and ResNet50 for the automatic diagnosis of Potato Blight using leaf images. The pre-trained networks perform better than OptCoNet. However, optimizing OptCoNet with WOA yielded better performance metrics such as accuracy, F1 Score, Precision and Recall than the pre-trained networks. As diseases in agricultural plants continue to remain a major concern, further research in this field with modern state-of-the-art methods is needed.

References

[1] Tripti Goel, Murugan Raman, Seyedali Mirjalili, and Deba Chakrabartty. Optconet: an optimized convolutional neural network for an automatic diagnosis of covid-19. *Applied Intelligence*, 51:1–16, 03 2021.

[2] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[3] V Ramya and M Anthuvan Lydia. Leaf disease detection and classification using neural networks. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(11):207–210, 2016.

[4] R Meena Prakash, GP Saraswathy, G Ramalakshmi, KH Mangaleswari, and T Kaviya. Detection of leaf diseases and classification using digital image processing. In *2017 international conference on innovations in information, embedded and communication systems (ICIIECS)*, pages 1–4. IEEE, 2017.

[5] Monzurul Islam, Anh Dinh, Khan Wahid, and Pankaj Bhowmik. Detection of potato diseases using image segmentation and multiclass support vector machine. In *2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE)*, pages 1–4. IEEE, 2017.

[6] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[7] Md Asif Iqbal and Kamrul Hasan Talukder. Detection of potato disease using image segmentation and machine learning. *2020 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, pages 43–47, 2020.

[8] Dor Oppenheim and Guy Shani. Potato disease classification using convolution neural networks. *Advances in Animal Biosciences*, 8(2):244, 2017.

[9] N Ananthi, K Kumaran, et al. Detection and identification of potato plant leaf diseases using convolution neural networks. *European Journal of Molecular & Clinical Medicine*, 7(4):2753–2762, 2020.

[10] Rakesh. S and Sudhakar. P. Deep transfer learning with optimal kernel extreme learning machine model for plant disease diagnosis and classification. *International Journal of Electrical Engineering and Technology (IJEET)*, 11(9):160–178, 2020.

[11] Trong-Yen Lee, Jui-Yuan Yu, Yu-Chun Chang, and Jing-Min Yang. Health detection for potato leaf with convolutional neural network. In *2020 Indo–Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN)*, pages 289–293. IEEE, 2020.

[12] Ashraf Darwish, Dalia Ezzat, and Aboul Ella Hassanien. An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis. *Swarm and Evolutionary Computation*, 52:100616, 2020.

[13] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 2014.

[14] Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in Engineering Software*, 95:51–67, 2016.

[15] Patrick R. Hof and Estel Van Der Gucht. Structure of the cerebral cortex of the humpback whale, megaptera novaeangliae (cetacea, mysticeti, balaenopteridae). *The Anatomical Record*, 290(1):1–31, 2007.

[16] William A. Watkins and William E. Schevill. Aerial Observation of Feeding Behavior in Four Baleen Whales: Eubalaena glacialis, Balaenoptera borealis, Megaptera novaeangliae, and Balaenoptera physalus. *Journal of Mammalogy*, 60(1):155–163, 02 1979.

[17] Jeremy A. Goldbogen, Ari S. Friedlaender, John Calambokidis, Megan F. McKenna, Malene Simon, and Douglas P. Nowacek. Integrative Approaches to the Study of Baleen Whale Diving Behavior, Feeding Performance, and Foraging Ecology. *BioScience*, 63(2):90–100, 02 2013.

[18] Shimin Li, Huiling Chen, Mingjing Wang, Ali Asghar Heidari, and Seyedali Mirjalili. Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems*, 111:300–323, 2020.

[19] Tanya Latty and Madeleine Beekman. Food quality and the risk of light exposure affect patch-choice decisions in the slime mold physarum polycephalum. *Ecology*, 91(1):22–27, 2010.

[20] Alex Lavaee. Plantifydr dataset, version 3. 2021.