Fusion of Color, Shape and Optimization Techniques for Tomato Plant Leaf Disease Detection Using Machine Learning

Puja Dipak Saraf[1*], Jayantrao B. Patil[2], Nitin N.Patil[3]

[1,2,3]Department of Computer Engineering, R.C. Patel Institute of Technology, Shirpur, Maharashtra-425405, India
Email: pujasaraf20@gmail.com
*Corresponding Author

Received: 17.04.2024 Revised : 10.05.2024 Accepted: 28.05.2024

ABSTRACT
Tomato leaf diseases stance a major threat to tomato cultivation, causing substantial yield losses and economic damage. Early and precise identification of leaf diseases is vital for appropriate intrusion and disease control. This study investigates the effectiveness of ML techniques MLP Neural Network in classifying tomato leaf diseases using shape and color features extracted from leaf images. Four different optimization algorithms (SGD, RMSProp, AMSGrad and Momentum SGD) were evaluated across three learning rates (0.5, 0.01 and 0.00001) for training a neural network model. The results demonstrated that RMSProp and AMSGrad outperformed SGD and Momentum SGD in terms of classification accuracy. RMSProp achieved the highest accuracy at a learning rate of 0.5, while AMSGrad excelled at both learning rates of 0.01 and 0.00001 with 95.00% accuracy. The proposed method using shape and color features and the AMSGrad optimizer can be an effective tool for early detection and control of tomato leaf diseases.

Keywords: Tomato leaf disease classification, shape features, color features, machine learning, optimization algorithms.

1. INTRODUCTION

Tomato (Solanum lycopersicum) cultivation is of paramount importance in global agriculture, providing a vital source of nutrition and income for millions of people worldwide. Tomato leaf diseases pose a significant threat to tomato cultivation, causing substantial yield losses and economic damage [1]. However, the tomato crop is consistently threatened by a number of diseases that can considerably jeopardize crop yield and quality. The appropriate and exact detection of diseases affecting the leaves of tomato plants is pivotal for operative disease managing and the preservation of agricultural productivity [2]. Recent years have witnessed a transformative shift in the agricultural landscape, marked by the integration of advanced technology, specifically the application of machine learning and computer vision techniques, to tackle the challenging issue of plant disease detection. Machine learning methods have emerged as powerful tools capable of automating disease identification processes, offering a multitude of advantages, including rapid and non-invasive assessment, reduced reliance on manual labor and the facilitation of precision agriculture. This research paper is dedicated to the expansion and deployment of a machine learning-based approach for the detection of diseases on tomato plant leaves. Early and accurate identification of leaf diseases is crucial for timely intervention and disease control [3]. The primary objective is to provide a reliable and effective method for the early identification of diseases, thereby supporting farmers and agrarian specialists in making well-informed verdicts to control disease spread and mitigate crop damages. Experts' visual assessment is the basis of traditional disease diagnosis techniques, which can be laborious, subjective, and prone to human error. Machine learning techniques offer a promising alternative for automated disease detection and classification [4]. In this study, we explore the significance of early disease detection in tomato farming, investigate existing disease detection methods and present a novel approach that combines the power of machine learning with several optimization techniques. Specifically, we incorporate Stochastic Gradient Descent (SGD), AMSGrad, Momentum SGD and RMSprop, aiming to improve the exactness and efficiency of disease detection. These optimization techniques play a vital role in fine-tuning the machine learning models to achieve better convergence and more precise results. Throughout the following sections, we delve into the significance of early disease detection in tomato farming, assess the limitations of existing methods and introduce our machine learning-based approach, which leverages the aforementioned optimization techniques. Additionally, we discuss the dataset employed for model training and testing, the selection of machine learning algorithms and the criteria used for evaluating the model's performance. This research contributes not only to the field of agriculture but also aligns with the broader context of applying cutting-edge technology to advance sustainable and precision agricultural practices [5].

2. LITERATURE REVIEW

The detection of plant diseases, particularly those affecting tomato plants, is of utmost importance in the field of agriculture to ensure high crop yield and quality. With the advancement of machine learning and computer vision, researchers and practitioners have been exploring innovative approaches for early and accurate disease detection. In this section, we review some of the key studies and methodologies related to tomato plant leaf disease detection using machine learning techniques.

Recent advances in agricultural technology have enabled plant disease detection using spectroscopy [6, 9]. Plant nitrogen levels in the field can be determined using ground-level reflectance spectra. [10, 11]. Leaf spectral characteristics have been analyzed to predict crop yield, monitor leaf area index variations [13], describe the biophysical characteristics of agricultural crops parameters [14] and discriminate diseases [12, 15]. Distinct diseases often manifest as specific physiological and visual changes in their host plants. Studies have demonstrated the effectiveness of non-destructive methods for detecting leaf diseases on certain varieties [16, 17]. Technology used in computer vision is another efficient non-destructive approach for plant detection, offering environmental benefits and cost-effectiveness. Probably the most prominent symptoms of plant disease is leaf scarring. Diseased leaves exhibit uneven leaf texture or colour compared to healthy leaves. Additionally, the disease spot shape on diseased leaves differs. In lab settings, several imaging techniques and stable lighting environments have been studied. Investigators have explored different imaging techniques Innovative methods for extracting disease features, utilizing scientific methods to capture leaf images and construct classification models [18, 19]. Currently, laboratory-based machine vision technology can achieve a classification accuracy of 100% for plant diseases [20], equaling the highest accuracy achieved by spectral techniques. However, image feature extraction or selection is computationally intricate. Precise structures provide high accuracy in characteristic convinced plant or disease types, but when the plant diversity or disease class's changes, the feature extraction steps that need to be changed are spectral processing and image segmentation. Consequently, the accuracy of disease classification can decrease when dealing with new disease types. Presently, Plant disease classification problems make extensive use of deep learning (DL) methods, especially those that are based on convolutional neural networks (CNNs), a subset of DL [21, 22].

Here, our goal is to expand on the current knowledge and advance the field of detecting leaf diseases in tomato plants by leveraging machine learning techniques and optimization algorithms. Our approach is grounded in the valuable insights gained from previous research and our own experimental findings.

3. RESEARCH METHODOLOGY

3.1 Dataset Used

The Plant Village dataset [23] was utilized to obtain images of tomato diseases. In tomatoes, there are nine main categories of diseases, namely: 1) Target Spot, 2) Mosaic virus, 3) Bacterial spot, 4) Late blight, 5) Leaf Mold, 6) Yellow Leaf Curl Virus, 7) Spider mites (Two-spotted spider mite), 8) Early blight and 9) Septoria leaf spot. In our proposed work, the training dataset comprises 10,000 images, the validation dataset includes 7,000 images and the testing dataset consists of 500 images. Among the 10,000 training images, 1,000 belong to the healthy category, while 1,000 images are allocated to each specific tomato disease category mentioned earlier. Each class is represented by 700 photographs in the validation set, and each class is represented by 50 images in the test set. 50 randomly chosen photographs from each training set class were taken out of their respective folders for testing.

We created our project training dataset from the leftover training dataset by keeping the same number of pictures (1,000) in each class.

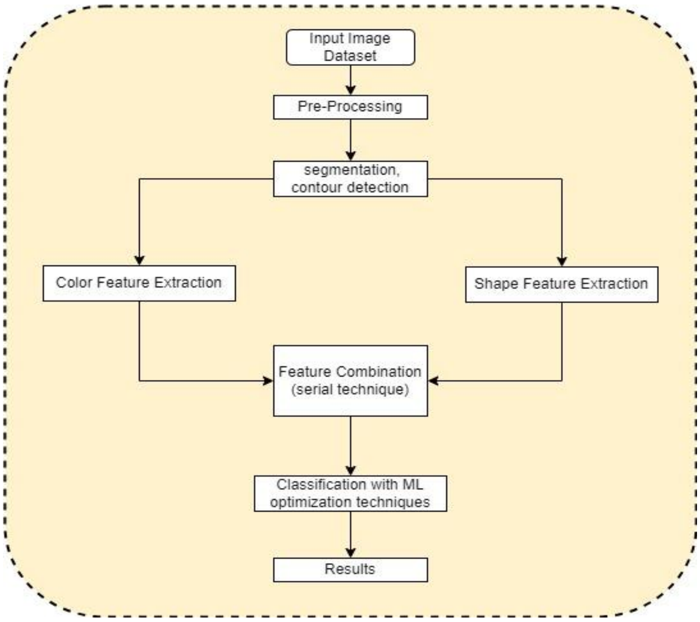

Figure 1. Proposed Research Methodology

When there were fewer than 1,000 photographs in a class, we created more images using data augmentation techniques. Using the Python Augmentor module, augmentation entails scaling, rotating, flipping, and cropping previous images to produce new ones that are similar. In cases where the number of When the number of images in a given class in the training dataset surpassed 1,000, we chose the top 1,000 images. This same process was trailed aimed at the validation dataset, ensuring that every class possessed exactly 700 images all. Here, meticulous approach is essential to avoid bias toward any particular class during the training of the Artificial Neural Network (ANN). The size of all images is set at 512 × 512 pixels and the format is JPEG.

Figure 2. Plant Village dataset samples.

A4 (210mm x 297mm) portrait page set-up should be used. The left, right, top margins should be 25mm and bottom margins should be 15mm. Do not use any headers, footers or footnotes. No page numbers. Single column. All main text paragraphs, including the abstract, must be fully (left and right) justified. All text, including title, authors, headings, captions and body, will be Times New Roman font.

3.2 Data Pre-Processing

As shown in Figure 1. The acquired dataset comprises images with minimal noise, obviating the necessity to eliminate any noise. The resolution of the dataset images was adjusted to 512 x 512, enhancing the efficiency of the model training process and facilitating computational benefits. Normalization, whether applied to input or output variables, is crucial for expediting the training process. This process enhances numerical conditions related to normalization issues. Moreover, normalization aids in bringing all pixel values of images into a specific range by utilising the mean and standard deviation values. Image flattening is achieved through the utilization of a Gaussian Filter, which is a linear filter commonly employed for noise reduction and edge detection. The Gaussian filter proves beneficial in blurring, reducing disparity and softening boundaries.

Otsu's thresholding method for adaptive image thresholding is implemented. Image thresholding is a simple yet efficient method for dividing an image into the foreground and background. This method, a form of image segmentation, transforms grayscale images into binary images, effectively segregating objects. Additionally, morphological transformations are employed for closing holes in binary images. Morphological transformations, consisting of basic operations based on image shape, are typically executed on binary images. The process involves two inputs: the original image and the structural component or kernel that determines how the process is conducted.

Feature Extraction

Algorithms are employed in this section on image processing to identify and separate different desired areas or forms (features) of digitised images. This technique made use of a variety of plant leaf characteristics, including colour and shape. Features such as surface area, surface perimeter, and disfigurement are examples of shape features. Shape features are crucial since they reduce the quantity of data collected and offer several ways to define an image using its most important components. The contour approach can be used to extract shape features. The variation in red, green, and blue is one of the colour features. The model has little trouble classifying photos using colour features. For every image, the quantity of pixels with the same colour value is calculated. As a result, using colour features for image classification is typical. The image's red, green, and blue channels can each be used to extract colour details. When it comes to optical character recognition or classification, feature extraction is very crucial.

Color Features

1. **Color Mean:** Calculate the mean value of pixel intensities in the red, green and blue channels using eq 1.

$$Mean_{\sum_{i=1}^{n}\frac{Pixel}{n}} \qquad \text{Eq.1.}$$

2. **Color Variance:** Measure the spread of pixel intensities around the mean in each color channel calculated using eq 2.

$$Variance_{\sum_{i=1}^{n}}$$

3. **Color Histogram:** Construct histograms for red, green and blue channels to represent the distribution of pixel intensities calculated by eq 3.

Eq.3.

$$Histogram_{(i)=No.of\,Pixelswithintensityi}$$

Shape Features

1. **Contour-based Features:** Compute the perimeter of the leaf contour by summing the distances between consecutive points and calculated using eq 4.

$$Perimeter_{shape} = \sum_{i=n}^{n} Distance(P_i, P_{i+1}) \qquad \text{Eq. 4.}$$

2. **Area-based Features:** Calculate the area of the leaf contour using the shoelace equation 5.

$$Area_{shape} = \frac{1}{2}\left|\sum_{i=1}^{n-1}(x_i * y_{i+1} - x_{i+1} * y_i) + x_n * y_1 - x_1 y_n\right| \qquad \text{Eq. 5.}$$

3. **Elongation:** Measure how stretched or elongated the leaf shape is calculated using Eq. 6.

$$Elongation_{shape} = \frac{MajorAxisLenght}{MinorAxisLenght}$$ Eq. 6.

3.3 Feature Combination

Combining features into a single feature vector often involves concatenating the individual feature values. The combined feature vector F can be represented as follows:

F = [Meancolor, Variancecolor, Histogramcolor, Perimetershape,Areashape, Elongationshape]

This is a simple concatenation of the individual feature values. If you have m color histogram bins and n features, the total length of the feature vector will be 3×m+n.

4. RESULT AND DISCUSSION

Optimization techniques play a vital role in training machine learning models, enabling them to converge efficiently towards optimal solutions. Here, Stochastic Gradient Descent (SGD), AMSGrad, Momentum SGD and RMSprop are some of the prominent optimization algorithms that address different challenges during training.

4.1 Stochastic Gradient Descent (SGD)

First we apply Stochastic Gradient Descent (SGD) is a modified of the gradient descent optimization algorithm that is frequently employed in training machine learning models. In contrast to conventional gradient descent, which computes the gradient using the full dataset, SGD updates the model parameters for each iteration based on a single randomly chosen training example. This presents randomness into the optimization process, which can lead to faster convergence and reduced computational requirements, particularly for large datasets [24]-[25].

Algorithm for Stochastic Gradient Descent (SGD):

1. **Input:** Training dataset, learning rate, number of epochs.
2. **Output:** Optimized model parameters.
3. **Algorithm Steps:**
   a. Initialize model parameters randomly.
   b. Shuffle the training dataset randomly.

For each epoch:

   c. Iterate over each mini-batch (a subset of the shuffled dataset).
      i. **For each mini-batch:**
         1. Calculate the loss function's gradient using the current mini-batch in relation to the parameters.
         2. Update the parameters expending the gradient and learning rate: parameter=parameter−learning_rate×gradientparameter=parameter−learning_rate×gradient.
   d. Repeat the mini-batch iteration for a specified number of mini-batches.
   e. Repeat the entire process for the specified number of epochs.
4. **Output:** Optimized model parameters after training.

We obtained the following results for Stochastic Gradient Descent at 0.5, 0.01 and 0.00001 Learning rate.

Table 1. SGD at 0.5, 0.01 and 0.00001 learning rate

	Learning Rate	**1st run**	**2nd run**	**3rd run**	**Mean**
SGD	0.5	0.93	0.92	0.85	0.9000
	0.01	0.92	0.91	0.85	0.8933
	0.00001	0.94	0.86	0.89	0.8967

4.2 RMSprop (Root Mean Square Propagation)

It addresses the problem of declining rates of learning in traditional GD. It splits the learning rate by the squared gradient average that decays exponentially, allowing for better handling of different parameter scales. This can lead to more stable training, particularly in deep neural networks [26].

Algorithm for RMSProp (Root Mean Square Propagation):

1. **Input:** Training dataset, learning rate, decay rate, number of epochs.
2. **Output:** Optimized model parameters.
3. **Algorithm Steps:**
   i. Initialize model parameters randomly.

ii. Initialize the moving the squared gradients' average for each parameter to zero.

For each epoch:

a. Iterate over each mini-batch (similar to SGD).

i. **For each mini-batch:**

1. Calculate the loss function's gradient using the current mini-batch in relation to the parameters.
2. Update the moving average of squared gradients using exponential decay: $squared_{gradient} = decay_{rate} * squared_{gradient} + (1 - decay_{rate}) * gradient^2$
.
3. Calculate the scaling factor for the moving average of squared gradients-based learning rate: $slicing_{factor} = \sqrt{\frac{1}{squared_{gradient} + epsilon}}$ (epsilon is a small constant for numerical stability)
4. Update the parameters using the scaled learning rate: $parameter = parameter - learning_{rate} * scaling_{factor} * gradient$

b. Repeat the mini-batch iteration for a specified number of mini-batches.

c. Repeat the entire process for the specified number of epochs.

4. **Output:** Optimized model parameters after training.

We obtained the following results for Root Mean Square propagation at 0.5, 0.01 and 0.00001 Learning rate.

Table 2. RMSProp at 0.5, 0.01 and 0.00001 learning rate

	Learning Rate	1st run	2nd run	3rd run	Mean
RMSProp	0.5	0.96	0.87	0.92	0.9167
	0.01	0.93	0.86	0.93	0.9067
	0.00001	0.92	0.85	0.93	0.9000

4.3 AMSGrad (Adaptive Moment Estimation with a Stable Gradient) Algorithm

It is an adaptation of the well-liked optimization technique known as the Adam optimizer, which is employed in deep neural network training. AMSGrad aims to address a potential issue in the original Adam optimizer where the learning rates for some parameters can become very small, leading to slow convergence in some cases [27].

Initialization

i. Initialize the model's parameters θ.
ii. Initialize the first and second moments for each parameter to zero.
iii. Initialize a small constant ε to prevent division by zero.
iv. Set the time step.
v. Choose hyperparameters: learning rate α (e.g., 0.001), β1 (normally close to 1, e.g., 0.9), β2 (normally close to 1, e.g., 0.999).

Repeat Until Convergence

i. Increment the time step
ii. Determine the loss's gradient in relation to the parameters..
iii. Update the first moment approximation:
iv. Update the second moment estimate with a new element-wise maximum operation:
v. Correct for bias in the first and second moment estimates:
vi. Compute the update for each parameter:
vii. Update the parameters:
viii. Repeat the above steps until convergence or for a predetermined number of times.

We obtained the following results for Adaptive Moment Estimation with a Stable Gradient at 0.5, 0.01 and 0.00001 Learning rate.

Table 3. AMSGrad at 0.5, 0.01 and 0.00001 learning rate

	Learning Rate	1st run	2nd run	3rd run	Mean
AMSGrad	0.5	0.91	0.86	0.94	0.9033
	0.01	0.94	0.94	0.95	0.9433
	0.00001	0.96	0.96	0.93	0.9500

4.3 Momentum Stochastic Gradient Descent (Momentum SGD) optimization algorithm

Momentum SGD (Stochastic Gradient Descent) is a popular optimisation approach for deep neural network and other machine learning model training. It is an extension of the basic SGD algorithm that helps accelerate convergence, especially when dealing with high-dimensional and non-convex optimization problems.

Initialization:

i. Initialize model parameters randomly.
ii. Set the initial velocity vector to zero. The velocity vector has the same dimensions as the model's parameters.

Hyperparameters:

i. Choose a momentum factor beta (normally a value between 0 and 1).
ii. Choose a learning rate learning_rate.

Iteration:

i. For each iteration of training:
ii. Sample a mini-batch of training data and determine the loss's gradient in relation to the model's parameters.
iii. Update the velocity vector:
iv. Velocity = beta * velocity + learning_rate * gradient.
v. Update the model constraints by the velocity:
   a.Parameters = parameters – velocity.

The momentum term (beta * velocity) acts as a "memory" that accumulates the historical gradients and guides the updates in the direction that has shown consistent gradients over time. This facilitates the removal of local minima and accelerates convergence in the optimisation process. In summary, the Momentum SGD algorithm introduces a velocity term that helps the optimization process to accumulate past gradients and move more consistently in the direction of the gradients, enabling faster convergence and better handling of irregular optimization landscapes. The hyperparameters beta and learning_rate play a vital character in defining the behavior of the optimization process. We obtained the following results for Momentum SGD at 0.5, 0.01 and 0.00001 Learning rate.

Table 4. Momentum SGD at 0.5, 0.01 and 0.00001 learning rate

	Learning Rate	1st run	2nd run	3rd run	Mean
Momentum SGD	**0.5**	0.89	0.93	0.88	0.90
	0.01	0.91	0.89	0.85	0.8333
	0.00001	0.93	0.90	0.88	0.9033

Results & Evaluation: Here the comparative analysis of SGD, RMSProp, AMSGrad and Momentum SGD are given.

Table 5. Comparative Analysis SGD, RMSProp, AMSGrad and Momentum SGD

	Learning Rate	1st run	2nd run	3rd run	Mean
SGD	**0.5**	0.93	0.92	0.85	0.9000
	0.01	0.92	0.91	0.85	0.8933
	0.00001	0.94	0.86	0.89	0.8967
RMSProp	**0.5**	0.96	0.87	0.92	0.9167
	0.01	0.93	0.86	0.93	0.9067
	0.00001	0.92	0.85	0.93	0.9000
AMSGrad	**0.5**	0.91	0.86	0.94	0.9033
	0.01	0.94	0.94	0.95	0.9433

	0.00001	0.96	0.96	0.93	0.9500
Momentum SGD	**0.5**	0.89	0.93	0.88	0.90
	0.01	0.91	0.89	0.85	0.8333
	0.00001	0.93	0.90	0.88	0.9033

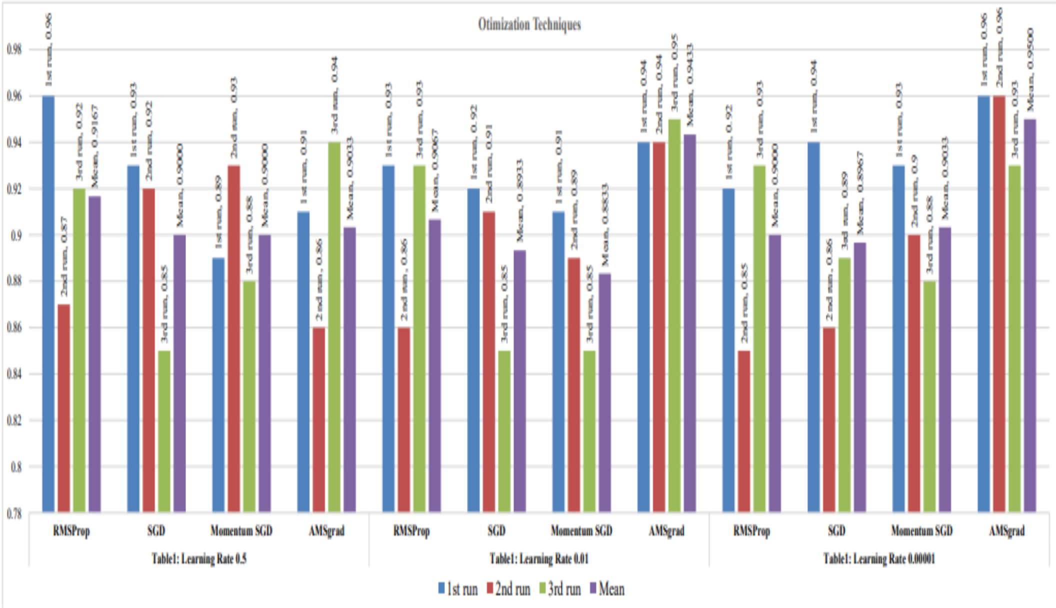

Figure 3. Comparative Analysis SGD, RMSProp, AMSGrad and Momentum SGD at different learning rate 0.5, 0.01 and 0.00001.

After Applying and evaluating above mentioned existing optimization techniques we can concluded as:

1. The accuracies obtained from training the model for 15 epochs, 3 times for each learning rate and each optimizer as well as the mean of the results from the 3 runs.
2. In this series of experiments, we investigated the performance of four different optimization algorithms (SGD, RMSProp, AMSGrad and Momentum SGD) across three dissimilar learning rates (0.5, 0.01 and 0.00001) in the context of training a neural network.
3. RMSProp optimizer achieves higher accuracy at learning rate 0.5. AMSGrad optimizer achieves the higher accuracy at learning rate 0.01 while Momentum SGD and SGD performs badly due to over fitting.
4. Again AMSGrad optimizer achieves the higher accuracy at learning rate 0.00001. It converge faster to the local minima to find out the optimal solution.

Further explore the performance of the proposed method, we compare its results with those of several established techniques in Table 6.

Table 6. Comparison of Tomato Leaf Disease Detection Accuracy across Different Approaches

Paper	Approach	Dataset	Diseases	Accuracy
28	VGG-19, VGG-16, ResNet, InceptionV3	Laboratory & Field	Multiple (4)	75%-90% (Lab), 60%-75% (Field)
29	AlexNet, GoogLeNet, ResNet	N/A	Multiple (3)	97.28% (ResNet)
30	BRBFNN	N/A	Multiple (7)	99.4%
31	DCGAN + GoogLeNet	Plant Village	Multiple (5)	94.33%
32	CNN with K-Means, SVM, RBF, MLP, NN, BPNN	N/A	Multiple (7)	99.4%
33	CNN	Dataset of 14240 images	Multiple (9)	95.53%
34	Deep CNN	Dataset of 39 plant species	Multiple (39)	83.12%
35	Residual CNN & Attention	Plant Village	Multiple (3)	98%

	CNN			
36	MFORS-based SVM	N/A	N/A	91.5%
37	Lightweight CNN	Plant Village	Multiple (10)	98.4%
38	Ensemble of VGG16 & VGG19	N/A	Multiple (2)	98.7%
39	Gabor Wavelet Transform & SVM	Custom dataset	Powdery mildew & Early blight	99.5%
40	DNN with PCA & PCA-WOA	N/A	N/A	90% (testing)
41	Modified-Xception (Transfer Learning)	Plant Village	Multiple (10)	99.55%
Proposed Method	Color & Shape features + Optimization (Multi-Layer Perceptron (MLP) Neural Network)	Plant Village	Multiple (9)	95.00%

Table 6. Compares the accuracy of various approaches for detecting tomato leaf diseases. It show cases models achieving up to 99.55% accuracy, utilizing diverse techniques like deep learning, color/shape features and optimization algorithms. Notably, proposed work using Multi-Layer Perceptron (MLP) Neural Network based on shape and color features achieves 95% accuracy, demonstrating its competitiveness while potentially offering an efficient and interpretable approach compared to some complex deep learning models.

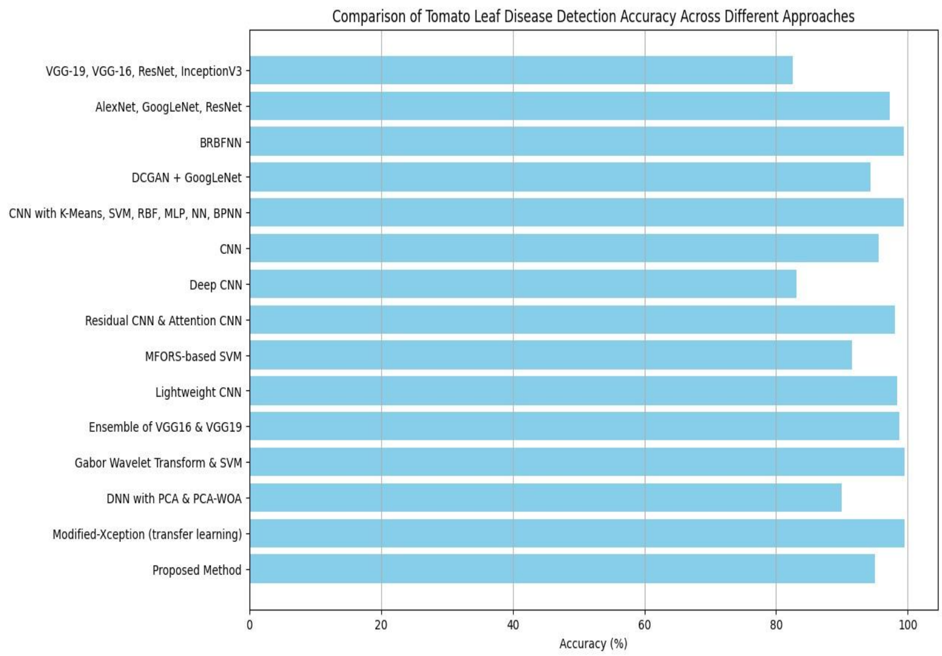

Figure 4. Comparison of proposed Tomato Leaf Disease Detection Accuracy.

We have proposed an innovative approach for the early detection and control of tomato leaf diseases, centering on the integration of shape and color features coupled with optimization techniques in neural network training. The research investigates the performance of four optimization algorithms—SGD, RMSProp, AMSGrad and Momentum SGD—across three different learning rates (0.5, 0.01 and 0.00001). The results underscore the pivotal role of selecting an appropriate optimization algorithm and learning rate to achieve optimal accuracy in disease identification. Notably, our work contributes a unique emphasis on optimization techniques, complementing existing research that employs diverse methodologies, including transfer learning, GANs and novel CNN strategies, for disease detection in tomato plants. The comparative analysis with other studies, such as those employing Adam, RMSprop and SGD optimizers, as observed in related works [2, 8 and 14], showcases the significance of our approach in the broader landscape of agricultural disease detection.

The significance of our work lies in its comprehensive exploration and estimation of various advanced convolutional neural network models for the automatic detection and classification of diseases in tomato plants. By conducting a meticulous comparative analysis of models such as VGG-19, VGG-16, ResNet and

Inception V3 on both controlled laboratory and real-world field datasets, our research not only provides insights on the advantages and disadvantages of many architectures but likewise addresses the challenges posed by varying environmental conditions. Moreover, our findings highlight the importance of parameter tuning over feature extraction, shedding light on a crucial aspect of model optimization. The emphasis on the practical applicability of our results, particularly in the context of real-world field data, distinguishes our work, making it an important addition to the area of agricultural disease detection. The proposed extension of optimizing these models for improved concert in actual field-oriented scenarios underscores the forward-looking nature of our research, offering a roadmap for future activities in enhancing the efficiency of automated plant disease identification systems.

5. CONCLUSIONS

The proposed method using shape and color features and the optimization techniques for effective and early detection and control of tomato leaf diseases, potentially contributing to improved tomato crop yields and reduced economic losses. This study investigated the performance of four optimization algorithms (SGD, RMSProp, AMSGrad and Momentum SGD) across three learning rates (0.5, 0.01 and 0.00001) in the context of training a neural network. The results demonstrated that RMSProp and AMSGrad outperformed SGD and Momentum SGD in terms of accuracy. RMSProp achieved the maximum accuracy at a learning rate of 0.5, while AMSGrad excelled at both learning rates of 0.01 and 0.00001. The superior performance of AMSGrad is attributed to its ability to reach the local minima more quickly and find the optimal solution. These findings highlight the significance of choosing the right optimization algorithm and learning rate for attaining optimal performance in neural network training.

ACKNOWLEDGEMENTS

We are thankful to Dr. Jayantrao B.Patil and Dr. Nitin N. Patil for their encouragement and moral support in preparing the manuscript.

REFERENCES

[1] Sobur, A., Kabir, M. H., Hossain, M. Z., Hossain, A., & Rana, I. C. (2024). Enhancing Tomato Leaf Disease Detection in Varied Climates: A Comparative Study of Advanced Deep Learning Models with a Novel Hybrid Approach. International Journal of Creative Research Thoughts| IJCRT, 12(2).

[2] Thakur, P. S., Khanna, P., Sheorey, T., & Ojha, A. (2022). Trends in vision-based machine learning techniques for plant disease identification: A systematic review. Expert Systems with Applications, 208, 118117.

[3] Upadhyay, S. K., & Kumar, A. (2022, April). An Accurate and Automated plant disease detection system using transfer learning based Inception V3Model. In 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) (pp. 1144-1151). IEEE.

[4] Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., & Stefanovic, D. (2019). Solving current limitations of deep learning based approaches for plant disease detection. Symmetry, 11(7), 939.

[5] Luna-Benoso, B., Martínez-Perales, J. C., Cortés-Galicia, J., Flores-Carapia, R., & Silva-García, V. M. (2021). Detection of diseases in tomato leaves by color analysis. Electronics, 10(9), 1055.

[6] Sankaran, S., Mishra, A., Ehsani, R., & Davis, C. (2010). A review of advanced techniques for detecting plant diseases. Computers and electronics in agriculture, 72(1), 1-13.

[7] Xing, J., Symons, S., Shahin, M., & Hatcher, D. (2010). Detection of sprout damage in Canada Western Red Spring wheat with multiple wavebands using visible/near-infrared hyperspectral imaging. Biosystems Engineering, 106(2), 188-194.

[8] Jones, C. D., Jones, J. B., & Lee, W. S. (2010). Diagnosis of bacterial spot of tomato using spectral signatures. Computers and electronics in agriculture, 74(2), 329-335.

[9] Ye, H., Huang, W., Huang, S., Cui, B., Dong, Y., Guo, A., & Jin, Y. (2020). Identification of banana fusarium wilt using supervised classification algorithms with UAV-based multi-spectral imagery. International Journal of Agricultural and Biological Engineering, 13(3), 136-142.

[10] Stroppiana, D., Boschetti, M., Brivio, P. A., & Bocchi, S. (2009). Plant nitrogen concentration in paddy rice from field canopy hyperspectral radiometry. Field crops research, 111(1-2), 119-129.

[11] Zhao, D., Reddy, K. R., Kakani, V. G., & Reddy, V. R. (2005). Nitrogen deficiency effects on plant growth, leaf photosynthesis, and hyperspectral reflectance properties of sorghum. European journal of agronomy, 22(4), 391-403.

[12] Panda, S. S., Ames, D. P., & Panigrahi, S. (2010). Application of vegetation indices for agricultural crop yield prediction using neural network techniques. Remote sensing, 2(3), 673-696.

[13] Brantley, S. T., Zinnert, J. C., & Young, D. R. (2011). Application of hyperspectral vegetation indices to detect variations in high leaf area index temperate shrub thicket canopies. Remote Sensing of Environment, 115(2), 514-523.
[14] Thenkabail, P. S., Smith, R. B., & De Pauw, E. (2000). Hyperspectral vegetation indices and their relationships with agricultural crop characteristics. Remote sensing of Environment, 71(2), 158-182.
[15] Schubert, T. S., Rizvi, S. A., Sun, X., Gottwald, T. R., Graham, J. H., & Dixon, W. N. (2001). Meeting the challenge of eradicating citrus canker in Florida—again. Plant disease, 85(4), 340-356.
[16] Swarbrick, P. J., SCHULZE-LEFERT, P. A. U. L., & Scholes, J. D. (2006). Metabolic consequences of susceptibility and resistance (race-specific and broad-spectrum) in barley leaves challenged with powdery mildew. Plant, Cell & Environment, 29(6), 1061-1076.
[17] Orchi, H., Sadik, M., & Khaldoun, M. (2021). On using artificial intelligence and the internet of things for crop disease detection: A contemporary survey. Agriculture, 12(1), 9.
[18] López-Maestresalas, A., Keresztes, J. C., Goodarzi, M., Arazuri, S., Jarén, C., & Saeys, W. (2016). Non-destructive detection of blackspot in potatoes by Vis-NIR and SWIR hyperspectral imaging. Food control, 70, 229-241.
[19] Karuppiah, K., Senthilkumar, T., Jayas, D. S., & White, N. D. G. (2016). Detection of fungal infection in five different pulses using near-infrared hyperspectral imaging. Journal of Stored Products Research, 65, 13-18.
[20] Lu, J., Ehsani, R., Shi, Y., de Castro, A. I., & Wang, S. (2018). Detection of multi-tomato leaf diseases (late blight, target and bacterial spots) in different stages by using a spectral-based sensor. Scientific reports, 8(1), 2793.
[21] Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture, 161, 272-279.
[22] Chen, J., Chen, J., Zhang, D., Sun, Y., & Nanehkaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. Computers and Electronics in Agriculture, 173, 105393.
[23] Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture, 161, 272-279.
[24] Mohamad, S., Alamri, H., & Bouchachia, A. (2022). Scaling up stochastic gradient descent for non-convex optimisation. Machine Learning, 111(11), 4039-4079.
[25] Sharma, A. (2018). Guided stochastic gradient descent algorithm for inconsistent datasets. Applied Soft Computing, 73, 1068-1080.
[26] Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.
[27] Zhang, J., Yang, X., Zhang, K., & Wen, C. (2021). An Adaptive Deep Learning Optimization Method Based on Radius of Curvature. Computational Intelligence and Neuroscience, 2021(1), 9882068.
[28] Ahmad, I., Hamid, M., Yousaf, S., Shah, S. T., & Ahmad, M. O. (2020). Optimizing pretrained convolutional neural networks for tomato leaf disease detection. Complexity, 2020(1), 8812019.
[29] Zhang, K., Wu, Q., Liu, A., & Meng, X. (2018). Can deep learning identify tomato leaf disease?. Advances in multimedia, 2018(1), 6710865.
[30] Chouhan, S. S., Kaul, A., Singh, U. P., & Jain, S. (2018). Bacterial foraging optimization based radial basis function neural network (BRBFNN) for identification and classification of plant leaf diseases: An automatic approach towards plant pathology. Ieee Access, 6, 8852-8863.
[31] Wu, Q., Chen, Y., & Meng, J. (2020). DCGAN-based data augmentation for tomato leaf disease identification. IEEE access, 8, 98716-98728.
[32] Vadivel, T., & Suguna, R. (2022). Automatic recognition of tomato leaf disease using fast enhanced learning with image processing. Acta Agriculturae Scandinavica, Section B—Soil & Plant Science, 72(1), 312-324.
[33] Kokate, J. K., Kumar, S., & Kulkarni, A. G. (2023). Classification of Tomato Leaf Disease using a Custom Convolutional Neural Network. Current Agriculture Research Journal, 11(1), 11(1), 2023.
[34] Towfek, S. K., & Khodadadi, N. (2023). Deep convolutional neural network and metaheuristic optimization for disease detection in plant leaves. Journal of Intelligent Systems and Internet of Things, 10(1), 66-75.
[35] Karthik, R., Hariharan, M., Anand, S., Mathikshara, P., Johnson, A., & Menaka, R. (2020). Attention embedded residual CNN for disease detection in tomato leaves. Applied Soft Computing, 86, 105933.
[36] Hassanien, A. E., Gaber, T., Mokhtar, U., & Hefny, H. (2017). An improved moth flame optimization algorithm based on rough sets for tomato diseases detection. Computers and electronics in agriculture, 136, 86-96.

[37] Agarwal, M., Gupta, S. K., & Biswas, K. K. (2020). Development of Efficient CNN model for Tomato crop disease identification. Sustainable Computing: Informatics and Systems, 28, 100407.
[38] Darwish, A., Ezzat, D., & Hassanien, A. E. (2020). 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.
[39] Mokhtar, U., Ali, M. A., Hassenian, A. E., & Hefny, H. (2015, December). Tomato leaves diseases detection approach based on support vector machines. In 2015 11th International computer engineering conference (ICENCO) (pp. 246-250). IEEE.
[40] Gadekallu, T. R., Rajput, D. S., Reddy, M. P. K., Lakshmanna, K., Bhattacharya, S., Singh, S., ... & Alazab, M. (2021). A novel PCA–whale optimization-based deep neural network model for classification of tomato plant diseases using GPU. Journal of Real-Time Image Processing, 18, 1383-1396.
[41] Thangaraj, R., Anandamurugan, S., & Kaliappan, V. K. (2021). Automated tomato leaf disease classification using transfer learning-based deep convolution neural network. Journal of Plant Diseases and Protection, 128(1), 73-86.