International Research Journal of Multidisciplinary Scope (IRJMS), 2025; 6(1):879-891

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2025.v06i01.02864

Tomato Leaf Disease Detection with YOLOV8 Leaf Extraction, Resnet-50 Classification, and Gpt-3.5 for Treatment Recommendations

Nilesh B Korade[1]*, Mahendra B Salunke[2], Amol A Bhosle[3], Gayatri G Asalkar[1], Dhanashri M Joshi[1], Aparna S Patil[1], Sunil M Sangve[4]

[1]Department of Computer Engineering, JSPM's Rajarshi Shahu College of Engineering, Pune, Maharashtra, India, [2]Department of Computer Engineering, PCET's, Pimpri Chinchwad College of Engineering and Research, Pune, Maharashtra, India, [3]Department of Computer Science and Engineering, MIT Art Design and Technology University, Pune, Maharashtra, India, [4]Department of Artificial Intelligence and Data science, Vishwakarma Institute of Technology, Pune, Maharashtra, India. *Corresponding Author's Email: nilesh.korade.ml@gmail.com

Abstract

India produces 20 million metric tons of tomatoes annually, with 150,000 metric tons being exported to international markets. India ranks as the leading producer and exporter of tomatoes globally, and tomato farming has a significant contribution to India's agricultural economy, with millions of farmers relying on tomato farming for their livelihood. Tomatoes are in high demand during the summer, but cultivating them at this time of year is challenging because the hot climate increases the susceptibility to numerous diseases. In this study, we collected 5250 images of tomato leaves suffering from seven distinct diseases. We identified regions of interest, such as leaves, using the YOLOv8 framework for object detection and passed these to the ResNet-50 model for classification. Using classification assessment metrics, the effectiveness of this framework was evaluated and compared with other deep learning architectures, such as CNN, AlexNet, VGG-19, and EfficientNetV2B7. We provide optimal treatment recommendations based on disease identification using a comprehensive disease treatment dataset, along with detailed explanations through GPT-3.5. The results demonstrate that YOLOv8 performs well in precise and real-time recognition of objects, making it extremely efficient for detecting regions of interest such as tomato leaves, while ResNet-50's deep architecture boosts disease classification accuracy by effectively distinguishing between various tomato diseases based on extracted features. Together, they form a powerful framework for accurate tomato disease prediction.

Keywords: Classification, Disease Yolo, GPT3.5, Leaf Extraction, Prediction, Resnet, Tomato.

Introduction

Approximately 15% of India's GDP is derived from agriculture, which also provides livelihoods for 150 million farmers nationwide. Among various crops, tomatoes are a notable agricultural product, with India being one of the largest producers and exporters globally. The economic significance of tomatoes has been highlighted by India's recent tomato production, which reached beyond 20 million metric tons, of which 150,000 metric tons were exported to other countries. Tomatoes are in high demand during the summer, but extreme temperatures vary from 35°C to 50°C, making it challenging for the crops to thrive (1). These harsh conditions increase the risk of infections from various diseases. If the diseases are not discovered and treated promptly, the affected crops weaken, struggle to recover, and may eventually die. Early detection and intervention are crucial to preventing the spread of diseases and ensuring crop survival during these difficult conditions (2). The description for principal diseases that damage tomato crops is given below:

Tomato Yellow Leaf Curl Virus (TYLCV)

The TYLCV-affected tomato plants exhibit small leaves that become yellow, particularly between the veins, with the leaves curling towards the center of the leaf. The virus is spread by silverleaf whiteflies, from infected to healthy plants. Infected seedlings display stunted growth, giving them a bushy appearance. In mature plants, only new growth after infection is reduced in size. Although the virus hampers tomato production, the fruit itself often remains visually unaffected despite the plant's overall decline in health (3).

(Received 17th October2024; Accepted 23rd January 2025; Published 31st January 2025)

Septoria Leaf Spot

One of the most damaging diseases of tomato leaves is Septoria leaf spot, caused by a fungus, Septoria lycopersici. It is especially severe in places where there is prolonged, humid weather. The spots on the leaves are circular in shape and approximately 0.06 to 0.25 inches in diameter. They have dark brown edges with tan to gray centers, often containing small black fruiting structures. The disease ascends from the oldest to the youngest growth. The leaves get somewhat yellow, then brown, and finally wither if there are several leaf lesions.

Tomato Mosaic Virus (ToMV)

It is a plant pathogenic virus that damages tomatoes or many other plants worldwide, and it is a member of the family Tobamoviridae and of the genus Tobamovirus. The leaves of infected plants may have mottled or mosaic patterns of light and dark green. In addition, the leaves could stoop, distort, or develop blisters. Fruit size reduction, deformed appearance, and slowed growth are other effects of the virus (4).

Tomato Bacterial Spot

It is caused by Xanthomonas bacteria and appears on the leaves as tiny, rounded patches, often less than 0.125 inches in diameter, which may appear water-soaked or wet and can seriously harm tomato crops. Fruits with lesions may become unmarketable.

Tomato Late Blight

Potatoes and tomatoes are susceptible to the damaging fungal disease known as tomato late blight, which is brought on by Phytophthora infestans and spreads rapidly by wind and rain and flourishes in cool, damp environments. As the disease progresses, leaves, stems, and fruits develop sporadic, water-soaked lesions that eventually turn brown and dry. Large, black, greasy patches appear on infected fruit, significantly reducing yield.

Tomato Early Blight

Alternaria solani, a common fungal pathogen that causes tomato early blight, primarily impacts leaves, stems, and fruits. On older leaves, the initial symptoms develop as tiny, brown to black lesions that are frequently encircled by a yellow halo. The spots enlarge and form concentric rings that resemble a bullseye pattern as the condition worsens (5). Although the warm, humid weather is frequently good for growing tomatoes, it also encourages the growth of bacterial, viral, and fungal diseases. Early disease detection on tomatoes is essential, as small, immediate treatments can frequently result in a full recovery if they are detected early. Treating a seriously infected plant, however, may not be effective because it is more likely to die, and the disease may spread to nearby crops. Taking early intervention reduces the possibility of losing a significant amount of the harvest and helps in stopping its propagation (6). Farmers have different obstacles in managing tomato infections that severely affect crop yield and quality.

- Accurate identification frequently requires specific knowledge and skills that not all producers possess, especially in remote or resource-constrained areas.
- Many diseases exhibit similar symptoms, making it challenging to differentiate between them just by visual inspection (7).
- Since some infections spread quickly within or between fields, early detection and accurate diagnosis are crucial to managing these conditions.
- In order to properly manage and rid a crop disease, farmers must be aware of the proper remedies after it has been detected. If the disease is not properly treated, it may spread and get worse, causing more harm to the crop and possibly a loss of production (8).

This research offers an innovative approach for the identification, management, and clarification of tomato diseases, empowering farmers to act quickly before a disease outbreak. We have gathered pictures of tomato plants suffering six serious diseases. Leaf extraction is done with YOLOv8, while disease categorization is done with ResNet-50. For disorders that have been discovered, GPT-3.5 offers thorough explanations and recommended courses of action. Our approach performs better in disease prediction and treatment suggestion when compared to CNN, AlexNet, VGG-19, and EfficientNetV2B7, applying measures like classification accuracy, loss/accuracy curves, and confusion matrices. The enhanced speed, accuracy, and real-time detection capabilities of YOLOv8's sophisticated architecture allow it to perform better in object detection. Accurate leaf extraction is ensured by its efficient handling of complex environments and smaller items, such as tomato crop leaves. However, due to its deep residual learning methodology, which enables it to recognize complex patterns from the retrieved leaf data, ResNet-50 performs extremely well in the categorization of diseases. Compared to more conventional models like AlexNet or VGG-19, it is more accurate because of its deeper layers, which provide better representation of disease features. A significant problem in agriculture is crop disease, especially when it comes to identifying issues with plant leaves. The international economy depends heavily on agriculture, and India is the world's second-largest producer of tomatoes. Diseases, however, significantly decrease tomato yield and quality. Technological developments in computer vision and deep learning offer potential for proficient crop disease forecasting, facilitating prompt treatments to enhance the production of agriculture. The TomConv model (9), based on an improved Convolutional Neural Network (CNN), divides tomato leaves into ten categories. The research employs 16,000 images from the PlantVillage dataset, including both healthy and diseased photographs. A four-layer CNN with max pooling was trained on the dataset images, which were transformed to 150x150 pixels resolution and partitioned into an 80:20 ratio for training and validation. The results demonstrate that the model delivers 98.19% accuracy over 105 epochs. The machine learning techniques (10), such as CNN, fuzzy-SVM, and R-CNN, are examined for the identification of diseases in tomato plant leaves. The dataset of 735 images comprises healthy samples as well as a variety of diseases used to test the performance of classifiers. To increase accuracy, the images were compressed to 256 by 256 pixels and turned into grayscale. With an accuracy of 96.74%, the R-CNN-based model was the most effective of all the tested classifiers. R-CNN uses a two-step method: it finds sick areas first, and then it categorizes the objects within each region. Mexico's agricultural sector covers 2.5 percent of its GDP, and tomatoes have become the country's largest exported agricultural product. Using a publicly available dataset and additional images captured in Mexican fields, the researcher (11) suggests a CNN-based model for identifying and categorizing tomato leaf diseases. The public dataset consists of up to 11,000 images and 2,500 photos collected from various Mexican agriculture fields. Generative Adversarial Networks (GANs) are used for data augmentation, producing synthetic samples that share the same attributes as the training distribution. The input for the network consists of 112 × 112 color images that have been normalized to (0, 1) values. Filters with values of 16, 32, 64, and 128 were used in the four convolutional layers of the proposed neural network. The kernel size was set to a value of 3 × 3, and the rectified linear unit (ReLU) was used as the activation function for each convolved node. According to the findings, the classification of leaf diseases had a training accuracy of 99.99% and a validation accuracy of 99.64%. The corresponding disease is accurately classified by the model with an F1 score, precision, and recall of 0.99. Farmers frequently find it challenging to contact experts in remote regions to take advice on preventive actions against unusual diseases, and physically identifying tomato diseases by closely examining the plants is a time-consuming and challenging task. The enhanced CNN model proposed (12) in this work contained an input layer accepting 128 × 128 size of image, two convolution layers with filter size 3 × 3 and strides of 1, two max pooling layers, a hidden layer, and a flattening followed by an output layer. The performance of the proposed approach was evaluated against transfer learning models such as Inceptionv3, ResNet152, and VGG19. The results demonstrate that the suggested model outperforms other models with a training accuracy of 98% and a testing accuracy of 88.17%. The hybrid CNN-RNN model (13) is designed to identify tomato plant illnesses, where several pre-trained CNN models are used for feature extraction. The next stage is combining the liquid time-constant networks (LTC) model with the refined CNN model. To enable the LTC model to capture sequential relationships between the CNN model's output features, they are concatenated into a sequential format before being fed. The model's performance will be optimized by hyperparameter tuning, which will investigate variations in parameters such as batch sizes, learning rates, and the number of RNN units. To evaluate accuracy, computational cost, and training time, different model combinations are investigated. The study aims to create lightweight architectures that work in contexts with limited resources. For the automatic classification of ten varieties of tomato leaves, seven effective

Bayesian-optimized deep hybrid learning models that leverage the synergistic advantages of machine learning and deep learning were proposed (14). The study used CNN for automatic feature extraction and conventional machine learning techniques for classification, including RF, XGB, GNB, SVM, KNN, MLR, and stacking. The data augmentation performed for nine classes using the ImageDataGenerator and Boruta feature filtering layer has been implemented on the PlantVillage dataset in order to extract the statistically significant features. The Bayesian optimization technique was employed to optimize the DL model's hyperparameters. The result demonstrates that the CNN-Stacking model produced the highest classification performance among the seven hybrid models, and for an unknown dataset, this model attained accuracy values of 98.268%, which needs only 0.174 seconds of testing time. Instead of relying on expensive expert analysis, the suggested (15) deep learning-based framework to recognize tomato plant diseases by examining pictures of tomato leaves will assist farmers in classifying diseases impacting tomato-growing crops by simply taking an image of infected leaves. Conditional-GAN is utilized for image augmentation, offering synthetic images for training purposes, and the DenseNet model was trained on photos of tomato plants and tomato synthetic images in order to detect tomato illness. For 5, 7, and 10 disease class classification tasks, the proposed model's accuracy on the original PlantVillage dataset is 98.16%, 95.08%, and 94.34%, respectively, and when using the original PlantVillage + synthetic images dataset, the accuracy is 99.51%, 98.65%, and 97.11%. In the world of agriculture, deep learning is crucial for addressing the problem of plant disease identification, as it requires an extensive amount of effort, a thorough understanding of different plant illnesses, and more processing time to detect diseases in plants. Based on transformer aggregation, the researcher (16) presented an innovative approach to forecast crop pathology on a global-local feature space. The minority classes are augmented, so the effects of class imbalances are mitigated, and the dataset is normalized once it has been augmented to make it appropriate for the model's input. Training and disease classification were conducted using the Plant Village dataset and the VietNam strawberry disease dataset. A multi-level CNN model with ten layers builds the architecture where the first convolutional layer has 64 filters with a 3×3 kernel size and accepts an image input shape of 256×256×3. The fourth, sixth-, and eighth-layer output channels are designated as the local features, where the global feature is the dropout layer's output. Multiple evaluations demonstrate that the proposed strategy performs well in Plant Village and VNStr by 99.18% and 94.05%, respectively.

Methodology

After visiting various fields, we constructed tomato disease datasets for six primary diseases. The Yolov8 has been trained on annotated pictures of tomato leaves in order to extract leaves. Preprocessing techniques like resizing, normalization, and augmentation have been applied to the retrieved image. The preprocessed images are fed to ResNet-50 training after being divided into sets for training and validation. The extracted leaf from the live-captured image was categorized by the trained model. The user is given treatment recommendations, disease descriptions, their effects, etc., as illustrated in Figure 1.

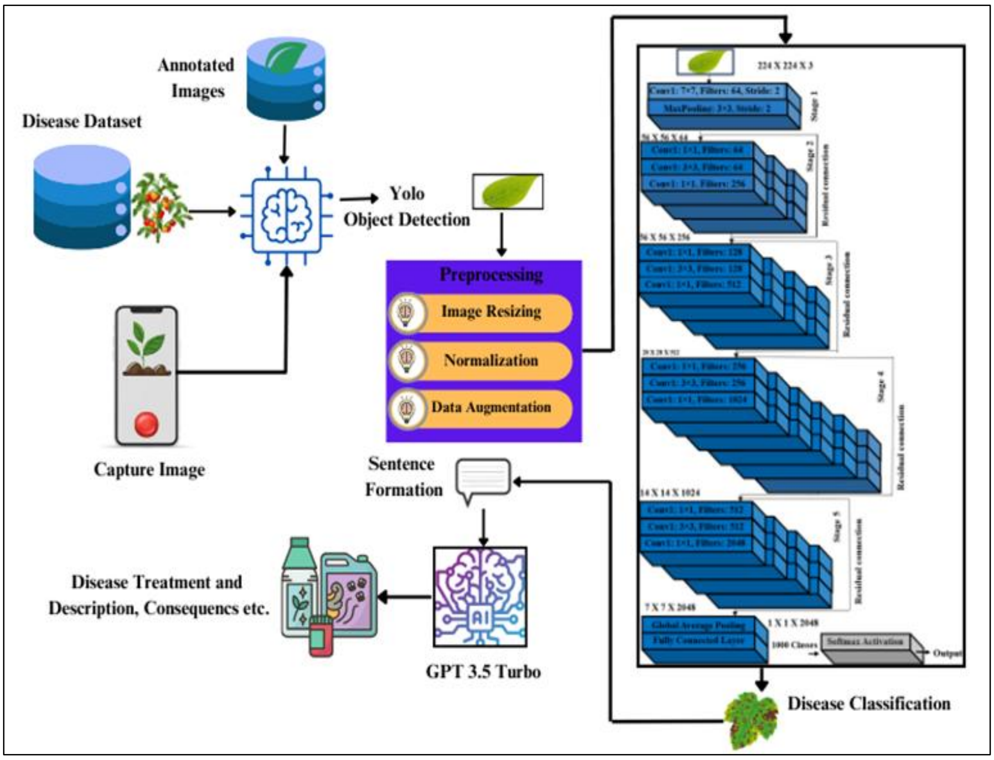


Figure 1: Architecture for Tomato Disease Detection Using YOLOv8, ResNet-50, and GPT-3.5

Dataset

The dataset contains 5250 pictures for seven categories, including six diseased and one healthy category, which were gathered from publicly available agricultural databases, research repositories, and different tomato plant agricultural fields. To ensure consistency, the photographs were taken in both controlled and natural lighting conditions, using regular cameras. The dataset captures variations in the quality of images, such as resolution, background clutter, leaf angles, and lighting conditions, to guarantee that the model is robustly trained and evaluated under multiple scenarios. Figure 2 demonstrates the type of disease and how it affects tomato leaves. We have created and annotated a separate leaf dataset containing plant pictures to enable the Yolo algorithm to extract leaves from a given plant image.

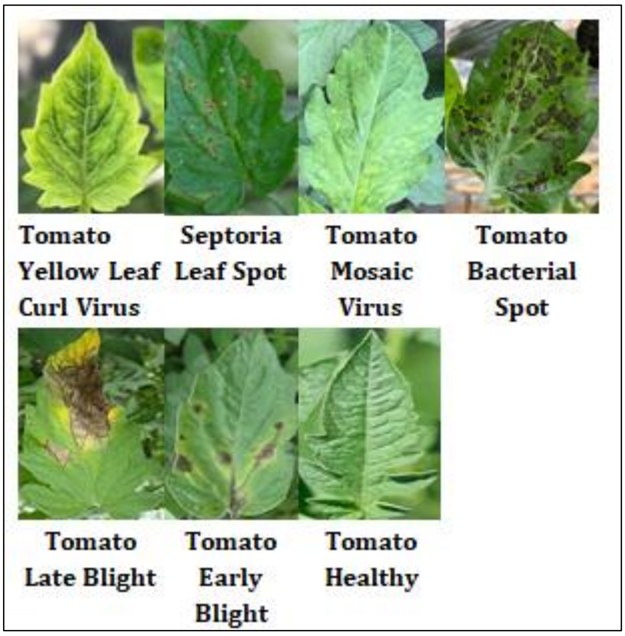


Figure 2: Tomato Leaf Dataset Representing All Disease Categories

Initial Preprocessing

Variations in leaf structure, lighting circumstances, and overlapping symptoms among diseases make it challenging to generalize the model to different tomato varieties, growing conditions, and disease severity levels. To solve this, the model needs broad, high-quality datasets encompassing an extensive range of varieties and environmental conditions, as well as data augmentation approaches to boost robustness. To make sure that the input data is in a format and quality appropriate for training and inference, image preprocessing is an essential phase before feeding images into deep learning models. Image augmentation is performed to increase the size and diversity of the dataset before employing models to improve the model's performance, which helps the model be more resilient and recognize patterns in various conditions, leading to better generalization (17). As most deep learning models require a defined input image size (for example, ResNet-50 requires 224x224x3, or AlexNet requires 227x227x3), all images are resized in order to satisfy the specifications. Normalization helps scale pixel values, which usually range from 0 to 255, to a smaller range (generally between 0 and 1), which improves training stability and improves convergence by preventing the model from dealing with large variance in pixel values (18).

Leaf Extraction

With processing an image in a single pass through a neural network that divides the image into a grid and predicts bounding boxes and class probabilities directly from the image in a single forward pass, the YOLO (You Only Look Once) architecture was designed for real-time object recognition. The YOLOv8 nano model is a lightweight form of YOLOv8 that is well suited for applications that have limited processing resources while retaining high accuracy. The three primary parts of Yolo architecture are the backbone, primarily responsible for feature extraction; the head, which carries out bounding box regression and object categorization; and the neck, which aggregates features at various scales (19).

Backbone (Feature Extraction): These layers extract essential features like edges, textures, and more intricate patterns from the input image by using convolution techniques. A modified CSPDarknet, an optimized Darknet, using Cross Stage Partial (CSP) networks, serves as the foundation for YOLOv8 that helps to decrease the burden on computing while maintaining significant spatial and contextual information (20). Let $X \in R^{HxWx3}$ stand for the input image, where H stands for height, W for width, and 3 for the RGB color channels. A number of feature maps are calculated by the backbone are as below (21):

$$F = Backbone(X) \in R^{H' x W' x C} \quad [1]$$

Where F is the feature map, C is the number of feature channels, and H′ and W' are the reduced spatial dimensions generated by the convolution and pooling layers. Residual blocks are employed in the latest versions of YOLOv8 to enhance gradient flow and enable deeper networks that enable the network to learn an identity mapping, which guarantees that crucial information is preserved across layers (22). The residual block is expressed as follows:

$$y = F(x) + x \quad [2]$$

where x is the block's input and F(x) is the transformation, the block applies, usually convolution and activation. In order to reduce computation while preserving accuracy, YOLOv8 employs CSPNet, which divides the feature map into two parts, processes one, and then combines it with the unprocessed piece (23).

Neck (Multi-scale Feature Aggregation): In order to detect objects at various scales, the neck is constructed to gather and integrate feature maps across multiple backbone layers, as it is essential for both large and small item detection. The feature pyramid networks combine data from several backbone layers to extract high-level and low-level information. High-level features record significant contextual information, while low-level features assist in the detection of small objects. Path Aggregation Network (PANet) is used to guarantee efficient feature aggregation and enhance information flow between layers.

Head (Detection and Prediction): Within each bounding box, the Yolo head classifies the object and predicts the bounding box coordinates. The YOLOv8 employs an anchor-free technique, which implies that box coordinates, abjectness scores, and class probabilities are predicted directly rather than using predefined anchor boxes (24). Below is a representation of each object's bounding box prediction:

$$\hat{x} = \sigma(x_g) + c_x \text{ and } \hat{Y} = \sigma(y_g) + c_y \quad [3]$$

YOLOv8 uses the center coordinates (x, y), width w, and height h to estimate bounding boxes. From grid cell coordinates, these values are forecasted as offsets. The coordinates of the grid cell are (cx, cy), and the predicted offsets, σ(xg) and $\sigma(yg)$, are normalized using the sigmoid function.

With the use of a sigmoid activation function, YOLOv8 predicts an abjectness score Pobj for every grid cell, indicating whether or not the grid cell includes an object (25).

$$P_{obj} = \sigma(o_g) \quad [4]$$

Where o_g represents raw score for the grid cell.

A softmax or sigmoid function is used to calculate the class probabilities (Pclass,i) for each grid cell, depending on whether the task is multi-class or multi-label classification.

$$P_{class,i} = \frac{e^{c_i}}{\sum j^{e^{c_j}}} \quad [5]$$

Where for class i, the ci represent the raw class score and the sum is taken over all classes.

Disease Classification

The ResNet-50 is a deep CNN type of residual network family with 50 layers distinguished by its creative use of residual connections, also referred to as skip connections. Convolutional and identity blocks are the primary structural components of ResNet-50, allowing the model to maintain accuracy as the network grows in depth.

Convolution layer: A CNN's fundamental component, the convolutional layer, is mainly used for feature extraction from images by applying a set of filters, or kernels, to the input data (26). These filters move across the image or other input to find patterns like edges, textures, shapes, and higher-level features. The basic function of a convolutional layer can be summarized as follows (27):

$$y_{i,j,k} = \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{c=1}^{C} x_{i+m,j+n,c} \cdot w_{m,n,c,k} + b_k \quad [6]$$

Where the bias term for the k^{th} filter is represented by b_k. This summation covers M×N spatial dimensions and all C input channels. From the c^{th} input channel, the $x_{i,j,c}$ represents the input at place [i, j]. For the k^{th} filter, $y_{i,j,k}$ represents the output at place [i, j] and $w_{m,n,c,k}$ represent the weight for the k^{th} filter at position [m, n] for channel c.

Residual Learning: The vanishing gradient problem, in which gradients are incredibly small as they are backpropagated through numerous layers, is one of the primary issues with extremely deep networks, which results in slower updates and can lead to a network that is unable to perform learning. Instead of learning the entire mapping, ResNet introduced the concept of residual learning, which enables the model to learn residuals, that is, the difference between the input and the intended output.

The residual function F(x) is calculated by passing the input x through a sequence of convolutional layers in a residual block. The residual function and the initial input x are then added to determine the output of the residual block (28).

$$y = F(x, \{Wi\}) + x \quad [7]$$

The F(x,{Wi}) stands for the learned residual function, which is parameterized by the convolutional layers' weights w' and the identity mapping, x, gets added to the output directly.

Batch Normalization: By normalizing the inputs to each layer such that they have a mean of zero and a standard deviation of one, batch normalization (BN), an essential strategy in deep neural networks, reduces the issue of internal covariate shift and stabilizes and speeds up training. The batch normalization layer estimates the mean and variance of the activations for each feature in a mini-batch represented as below (29).

$$Mean\ (\mu_B) = \frac{1}{m} \sum_{i=1}^{m} x_i \quad [8]$$

$$Variance(\sigma_B^2) = \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_B)^2 \quad [9]$$

where m is the number of samples in the mini-batch and x is the activation value for the i^{th} input. The calculated mean and variance are then used to normalize each activation x.

$$\hat{X} = \frac{x_i - \mu\beta}{\sqrt{\sigma_\beta^2 + \epsilon}} \quad [10]$$

Where μβ and σ_β^2 are the mean and variance of the batch. To avoid division by zero, a small constant ϵ gets added.

Activation Function: ResNet-50 adds non-linearity to the network by using ReLU activation after each convolutional layer and batch normalization step, as it is necessary for the network to be able to model complicated relationships between inputs and outputs, as explained below (30).

$$ReLU\ f(x) = (0, x) \quad [11]$$

Global Average Pooling: It generates a single scalar value for each feature map by calculating the average value of each feature map across its spatial dimensions, i.e., height and width.

$$yc = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{i,j,c} \quad [12]$$

where x_{ijc} is the value at location (i,j) in the c^{th} feature map, y_c is the output for the c^{th} channel, and H and W are the feature map's height and width (31).

Treatment Recommendation

The disease that was diagnosed was included in the query that was sent to GPT-3.5, along with requests for suitable agricultural remedies, disease specifications, consequences, etc. A variation of OpenAI's (32) Generative Pre-trained Transformer (GPT) series, GPT-3.5 Turbo was created especially to provide high-performance natural language creation and interpretation at a lower computing cost than its predecessors. In comparison to the more powerful GPT-4, the speed-optimized GPT-3.5 Turbo responds more quickly without compromising accuracy. The large corpus of heterogeneous text material used to train GPT-3.5 Turbo encompasses a wide range of subjects and includes books, scholarly articles, websites, and other publicly accessible text sources. The average response time for the majority of queries is between one and two seconds (33).

Performance Evaluation

We used several standard metrics that are frequently used in classification tasks, including accuracy, precision, recall, and F1-score, to evaluate the performance of our proposed approach based on the test dataset, which consists of tomato plant images representing various diseases. The most basic measure is accuracy, which shows the percentage of successfully predicted instances among all forecasts made (34). Precision estimates the percentage of predicted positive cases that resulted in actual positive (35). The model's recall, also referred to as sensitivity, and assesses its capacity to identify each true positive instance. The F1-Score is calculated by taking the harmonic mean of recall and precision (36).

$$Accuracy = \frac{[TruePositive + TrueNegative]}{Total\ Samples} \quad [13]$$

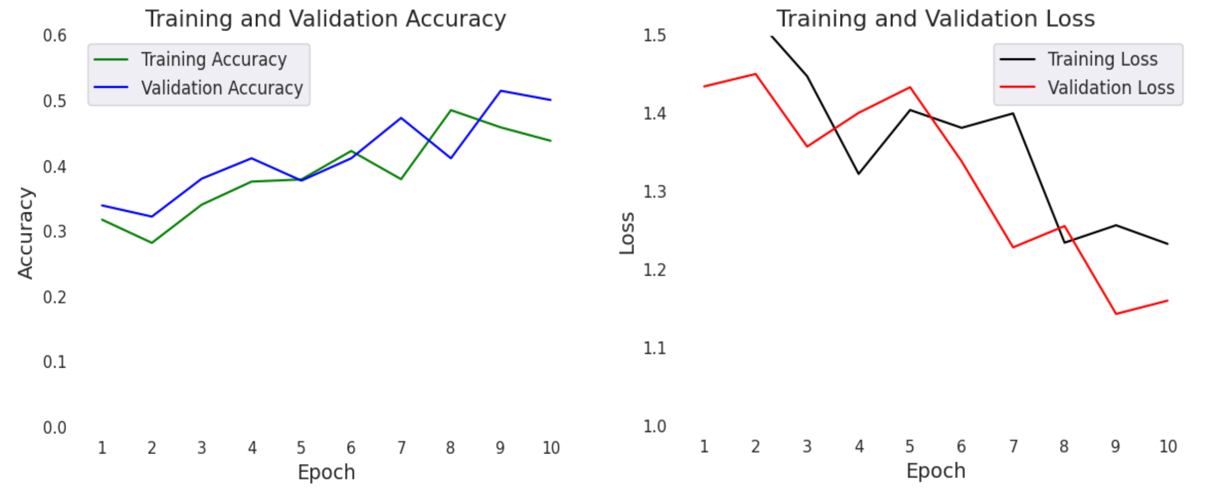
$$Precision = \frac{TruePositive}{[TruePositive + FalsePositive]} \quad [14]$$

$$Recall = \frac{TruePositive}{[TruePositive + FalseNegative]} \quad [15]$$

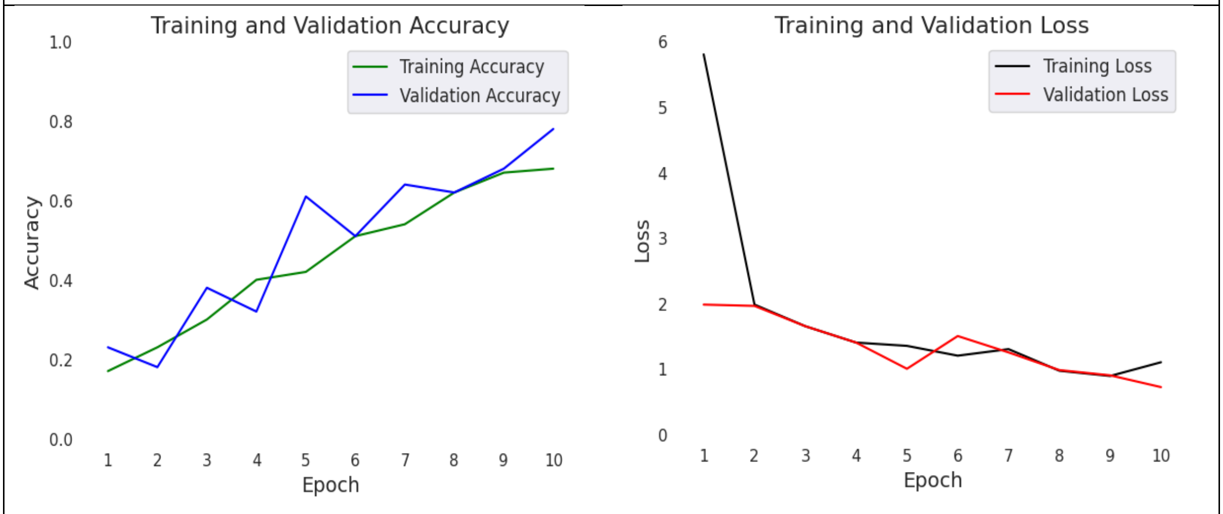
$$F1Score = 2 * \frac{[Precision * Recall]}{[Precision + Recall]} \quad [16]$$

Results and Discussion

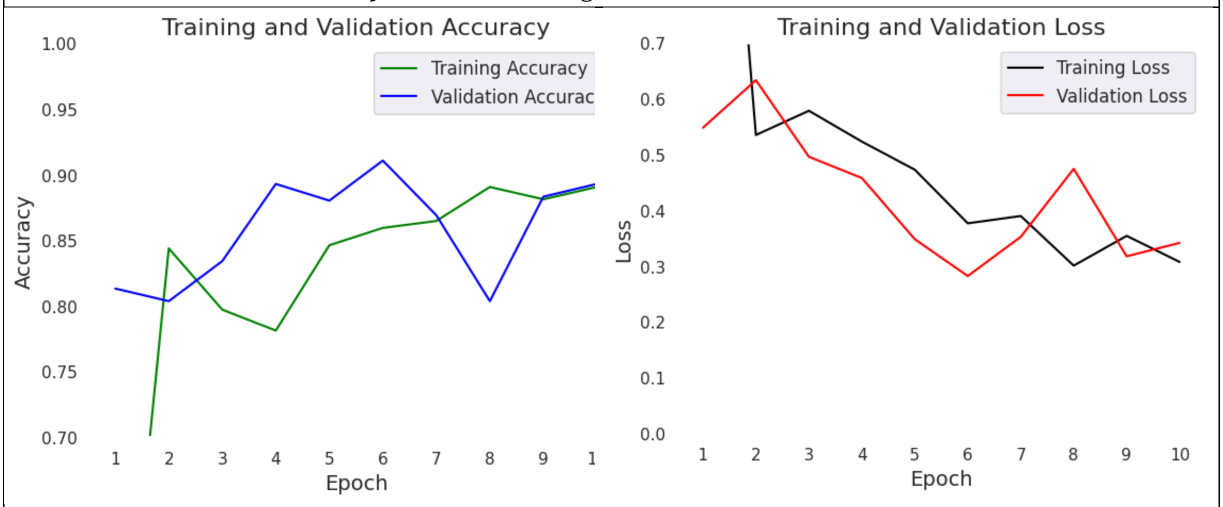
Using Google Colab, a cloud-based platform that offers an integrated environment for executing Python code, including deep learning models, evaluations for tomato leaf disease prediction have been carried out. A NVIDIA Tesla T4 GPU was used in this study to speed up the evaluation and training procedures. The enormous computational power and parallel processing capabilities of the T4 GPU, which has 16 GB of GDDR6 memory, make it an excellent choice for deep learning operations. These features also significantly reduce the amount of time required to perform model training when compared to CPU-based systems. For seven classes, including diseased and healthy, we gathered 5950 photographs of tomato-infected leaves. These were split into two sets: a training set with 4550 images and a validation set with 1400 images. Yolo8 was trained on a dataset with annotated leaves, which was used to extract leaves from pictures of tomato plants and carried out basic preprocessing like normalization, resizing, and augmentation. The photos were scaled to 640 × 640 pixels, with a learning rate of 0.001, and the Adam optimizer was employed for steady convergence. A batch size of 16 was chosen, and the model was trained for 50 epochs, with early stopping used to prevent over fitting. The VGG-19, AlexNet, EfficientNet, and ResNet-50 are examples of high-performing deep learning models that have been trained to evaluate their performance on disease classification tasks. We used the Adam optimizer with the categorical cross entropy loss function to train deep learning algorithms on the tomato disease dataset, and we monitored the model's performance using the accuracy metric. The verbosity level was set to 1, and the model was trained over 10 epochs with 65 steps per epoch. We incorporated callback functions such as learning rate scheduling (LR), early stopping (ES), and model check pointing (MC) to improve the training process. Figure 3 illustrates the performance of different models by visualizing the training and validation accuracy, loss, and number of epochs for each model.



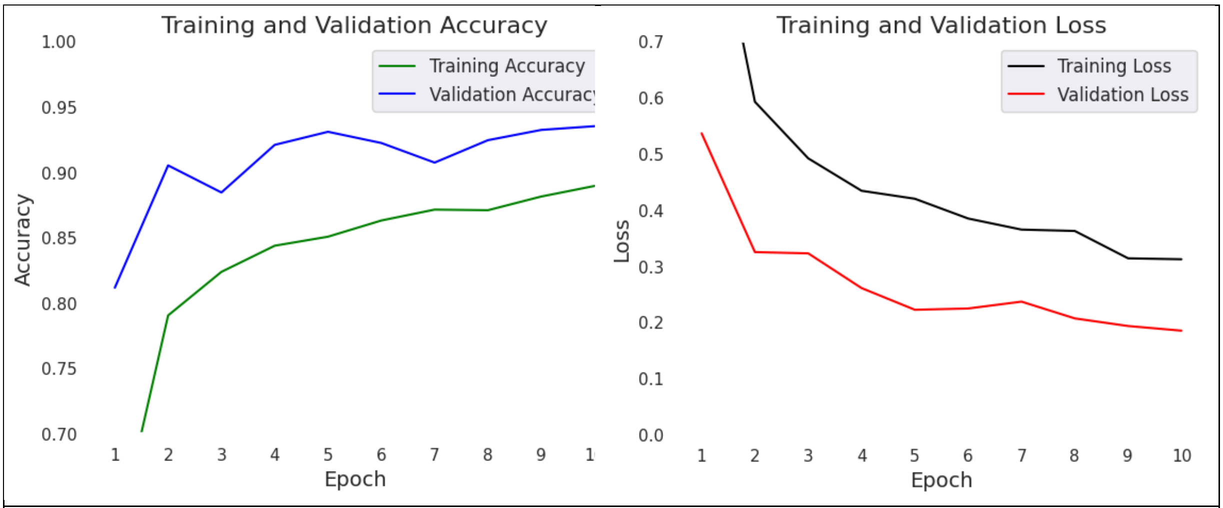
A) CNN Training and Validation Performance



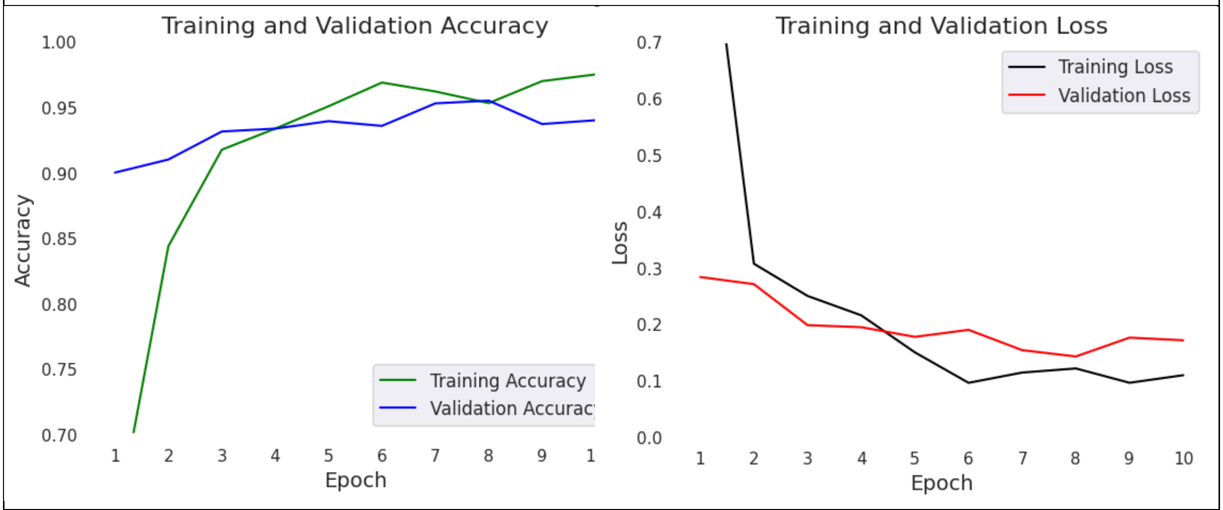
B) AlexNet Training and Validation Performance



C) VGG-19 Training and Validation Performance



D) EfficientNetV2B7 Training and Validation Performance



E) ResNet-50 Training and Validation Performance

Figure 3: Training and Validation Performance of the Different Models

We captured 140 photos in total, 20 for each category, and employed assessment measures to evaluate each model's performance. Table 1 shows the comparative performance of models based on accuracy, precision, recall, and F1-score. The confusion matrix is especially beneficial for highlighting areas where the model is likely to make incorrect classifications. The confusion matrix, demonstrating both correct and incorrect classifications for each category by each model, is presented in Table 2.

Table 1: Comparative Performance of Models Based on Accuracy, Precision, Recall, and F1-Score

Model	Accuracy	Precision	Recall	F1-Score
CNN	0.6286	0.6681	0.6286	0.6289
Alexnet	0.7286	0.7489	0.7286	0.7333
VGG-19	0.8429	0.8528	0.8429	0.8434
EfficientNetV2B7	0.9429	0.9489	0.9429	0.9407
ResNet50	0.9714	0.9727	0.9714	0.9714

Table 2: Confusion Matrix for Test Dataset

True Label	CNN							AlexNet							VGG-19							EfficientNetV2 B7							ResNet-50						
Predicted Label	Bacterial Spot	Early Blight	Late Blight	Septoria Leaf	Yellow Leaf Curl	Mosaic Virus	Healthy	Bacterial Spot	Early Blight	Late Blight	Septoria Leaf	Yellow Leaf Curl	Mosaic Virus	Healthy	Bacterial Spot	Early Blight	Late Blight	Septoria Leaf	Yellow Leaf Curl	Mosaic Virus	Healthy	Bacterial Spot	Early Blight	Late Blight	Septoria Leaf	Yellow Leaf Curl	Mosaic Virus	Healthy	Bacterial Spot	Early Blight	Late Blight	Septoria Leaf	Yellow Leaf Curl	Mosaic Virus	Healthy
Bacterial Spot	16	2	0	2	0	0	0	18	0	0	1	1	0	0	16	2	0	0	0	1	1	20	0	0	0	0	0	0	20	0	0	0	0	0	0
Early Blight	3	15	0	2	0	0	0	0	19	1	0	0	0	0	1	14	2	3	0	0	0	1	14	0	3	0	2	0	0	18	0	0	1	0	1
Late Blight	0	1	9	1	2	7	0	0	2	11	0	0	0	7	0	4	15	0	0	1	0	0	0	20	0	0	0	0	0	2	18	0	0	0	0
Septoria Leaf Spot	3	0	2	11	3	0	1	2	3	5	10	0	0	0	0	1	0	19	0	0	0	2	0	0	18	0	0	0	0	0	0	20	0	0	0
Yellow Leaf Curl Virus	4	4	0	2	8	0	2	0	0	0	3	14	3	0	0	2	0	0	18	0	0	0	0	0	0	20	0	0	0	0	0	0	20	0	0
Mosaic Virus	0	2	3	0	1	14	0	0	0	2	2	0	16	0	1	0	0	1	0	17	1	0	0	0	0	0	20	0	0	0	0	0	0	20	0
Healthy	0	2	3	0	1	0	14	0	1	3	0	0	2	14	0	0	0	1	0	0	19	0	0	0	0	0	0	20	0	0	0	0	0	0	20

According to the findings, ResNet-50 performs superior in training and validation in terms of accuracy and loss over CNN, AlexNet, VGG-19, and EfficientNetV2B7. Compared to the other models, ResNet-50 performs better and converges more smoothly and consistently, as seen by the training and validation accuracy vs. epoch and loss vs. epoch charts. This is particularly noticeable when the model is employed on leaves that have been extracted with the YOLO object detection framework, which improves classification accuracy by processing only the most essential area of the plant. The loss vs. epoch graph additionally demonstrates that ResNet-50 is more robust in interpreting unknown data, as it minimizes training and validation loss more effectively. The stronger performance of ResNet-50 during testing is further confirmed by the comparative performance table and confusion matrix table, which demonstrate that it consistently produces outstanding precision, recall, and F1-scores. ResNet-50 performed better on YOLO-extracted leaves as it was able to use residual learning, allowing the model to learn from high-quality, refined input data. ResNet-50 performed better on YOLO-extracted leaves as it was able to use residual learning, allowing the model to learn from high-quality, refined input data. GPT-3.5 was integrated into the system as a backend service that generates suggestions for treatment. The integration was carried out by using API calls offered by OpenAI's GPT-3.5 framework. Once the ResNet-50 model categorized the tomato crop disease, the label was fed into the GPT-3.5 model as an input. A structured prompt was developed to assist the model to generate precise, accurate, and feasible suggestions. For example, the prompt comprised a description of the circumstance, an in-depth discussion of its signs and symptoms, and a request for suggestions for treatment. The proposed platform is appropriate for real-time field deployment, employing YOLOv8 for rapid leaf extraction and ResNet-50 for diagnosis on edge devices such as the NVIDIA Jetson. Treatment suggestions have been developed using GPT-3.5 through a cloud-based API, ensuring speedy responses; however, offline alternative options can be explored for isolated places. The integration of drones with IoT devices may boost large-scale surveillance and customized suggestions.

Conclusion

With the objective of ensuring the correct classification, we used the YOLOv8 model to efficiently extract leaves from a dataset of 5250 tomato plant pictures covering seven disease classifications, which allowed us to isolate the most significant parts of the plant. A number of deep learning models, including CNN, VGG-19, AlexNet, EfficientNet, and ResNet-50, were then trained using the extracted leaf pictures as input. Among them, ResNet-50 outperformed the other models by a significant amount and delivered better training, validation, and testing results. Following classification, GPT-3.5 was used to further process the identified problems in order to provide comprehensive treatment recommendations, disease clarifications, and insights into the distinctive features of the disease and its possible complications. ResNet was able to successfully handle complicated features in the leaf pictures due to its robust design and residual learning. With integrating AI technologies, an extensive framework for diagnosing and treating diseases was developed, boosting its accuracy and usefulness in practical situations. Overall, the framework revealed exceptional efficiency, with ResNet-50 performing excellently in disease classification and YOLOv8 ensuring precise leaf extraction. In addition to enhancing model performance, this combination reduced errors that are frequently identified in noisy input data. Future research will concentrate on improving the model's generalizability across different kinds of plants, incorporating real-time disease monitoring, and expanding the dataset to include additional disease categories. Furthermore, future research into advanced algorithms like Vision Transformers (ViTs) may produce categorization results that are even more precise. Furthermore, utilizing more advanced GPT models to enhance the treatment recommendation system may offer farmers and agricultural workers even more customized and helpful insights. Creating a user-friendly interface, such as a mobile app, can allow a farmer to connect with the system more efficiently that allows farmers to input leaf pictures, get real-time disease detection findings, and view treatment recommendations in their native language. The integration of offline capabilities and voice help can improve accessibility in remote areas.

Abbreviations

Nil.

Acknowledgements

None.

Author Contributions

All authors contributed to the study conception and design.

Conflict of Interests

The authors declare that they have no competing interests.

Ethics Approval

Not applicable.

Funding

No funding received by any government or private concern.

References

1. Trivedi NK, Gautam V, Anand A, Aljahdali HM, Villar SG, Anand D, Goyal N, Kadry S. Early Detection and Classification of Tomato Leaf Disease Using High-Performance Deep Neural Network. Sensors (Basel). 2021; 21(23):7987.
2. Wang X, Liu J, Zhu X. Early real-time detection algorithm of tomato diseases and pests in the natural environment. Plant Methods. 2021; **17**(43):1-17.
3. Tang Z, He X, Zhou G, *et al.* A Precise Image-Based Tomato Leaf Disease Detection Approach Using PLPNet. Plant Phenomics. 2023; 5:0042.
4. Debnath A, Hasan MM, Raihan M, Samrat N, Alsulami MM, Masud M, Bairagi AK. A Smartphone-Based Detection System for Tomato Leaf Disease Using EfficientNetV2B2 and Its Explainability with Artificial Intelligence (AI). Sensors. 2023; 23(21):8685.
5. Manchanda M, Jadli K, Butola P, Sharma V, Bisht D, Purohit K. A Real Time Tomato Crop Disease Prediction using Deep Learning. International Conference on Inventive Computation Technologies (ICICT). 2024; 649-656.
6. Islam MP, Hatou K, Aihara T, Seno S, Kirino S, Okamoto S. Performance prediction of tomato leaf disease by a series of parallel convolutional neural networks. Smart Agricultural Technology. 2022; 2:100054. doi: 10.1016/j.atech.2022.100054.
7. Zhong Y, Teng Z, Tong M. LightMixer: A novel lightweight convolutional neural network for tomato disease detection. Frontier Plant Science. 2023; 14:1166296.
8. Kumar R, Chug A, Singh AP, Singh D. A Systematic Analysis of Machine Learning and Deep Learning Based Approaches for Plant Leaf Disease Classification: A Review. Journal of Sensors. 2022; (4):1:13.
9. Baser P, Saini JR, Kotecha K. TomConv: An Improved CNN Model for Diagnosis of Diseases in Tomato Plant Leaves. Procedia Computer Science. 2023; 218:1825-1833.
10. Nagamani HS, Sarojadevi H. Tomato Leaf Disease Detection using Deep Learning Techniques. International Journal of Advanced Computer Science and Applications. 2022; 13(1):305-311.
11. Guerrero-Ibañez A, Reyes-Muñoz A. Monitoring Tomato Leaf Disease through Convolutional Neural Networks. Electronics. 2023; 12(1):1-15.

12. Sakkarvarthi G, Sathianesan GW, Murugan VS, Reddy AJ, Jayagopal P, Elsisi M. Detection and Classification of Tomato Crop Disease Using Convolutional Neural Network. Electronics. 2022; 11(21):3618.
13. Le AT, Shakiba M and Ardekani I. Tomato disease detection with lightweight recurrent and convolutional deep learning models for sustainable and smart agriculture. Frontiers in Sustainability. 2024; 5:1383182.
14. Khan B, Das S, Fahim NS, *et al.* Bayesian optimized multimodal deep hybrid learning approach for tomato leaf disease classification. Scientific Reports. 2024*.* 14:21525.
15. Abbas A, Jain S, Gour M, Vankudothu S. Tomato plant disease detection using transfer learning with C-GAN synthetic images. Computers and Electronics in Agriculture. 2021; 187: 106279.
16. Ha MH, Nguyen DC, Do MT, *et al.* Plant pathology identification using local-global feature level based on transformer. Indonesian Journal of Electrical Engineering and Computer Science. 2024; 34(3):1582-1592.
17. Murcia-Gómez D, Rojas-Valenzuela I, Valenzuela O. Impact of Image Preprocessing Methods and Deep Learning Models for Classifying Histopathological Breast Cancer Images. Applied Science. 2022; 12(22):1-18.
18. Koresh HJD. Impact of the Preprocessing Steps in Deep Learning-Based Image Classifications. National Academy Science Letters. 2024; (47):645:47.
19. Jiang P, Ergu D, Liu F, Cai Y, Ma B. A Review of Yolo Algorithm Developments. Procedia Computer Science. 2022; 199: 1066-1073.
20. Vasanthi P, Mohan L. Efficient YOLOv8 algorithm for extreme small-scale object detection. Digital Signal Processing. 2024; 154. doi: 10.1016/j.dsp.2024.104682.
21. Feng F, Hu Y, Li W, Yang F. Improved YOLOv8 algorithms for small object detection in aerial imagery. Journal of King Saud University - Computer and Information Sciences. 2024; 36(6): 102113.
22. Wu T, Dong Y. YOLO-SE: Improved YOLOv8 for Remote Sensing Object Detection and Recognition. Applied Sciences. 2023; 13(24):12977.
23. Cheng G, Chao P, Yang J, Ding H. SGST-YOLOv8: An Improved Lightweight YOLOv8 for Real-Time Target Detection for Campus Surveillance. Applied Sciences. 2024; 14(12):5341.
24. Quang NH, Lee H, Kim N, *et al.* Real-time flash flood detection employing the YOLOv8 model. Earth Science Informatics. 2024;17(2024): 4809-29.
25. Tran TV, Ba QHD, Tran KT, *et al.* Designing a Mobile Application for Identifying Strawberry Diseases with YOLOv8 Model Integration. International Journal of Advanced Computer Science and Applications. 2023; 15(3):500-506.
26. Korade NB, Zuber M. Stock Price Forecasting using Convolutional Neural Networks and Optimization Techniques, International Journal of Advanced Computer Science and Applications. 2022; 13(11):378-385.
27. Korade NB, Zuber M. Boost Stock Forecasting Accuracy Using The Modified Firefly Algorithm And Multichannel Convolutional Neural Network. Journal of Theoretical and Applied Information Technology. 2023; 101(7):2668- 2677.
28. Wu D, Ying Y, Zhou M, Pan J, Cui D. Improved ResNet-50 deep learning algorithm for identifying chicken gender. Computers and Electronics in Agriculture. 2023; 205: 107622.
29. Zhang L, Bian Y, Jiang P, Zhang F. A Transfer Residual Neural Network Based on ResNet-50 for Detection of Steel Surface Defects. Applied Sciences. 2023; 13(9):5260.
30. Korade NB, Zuber M, Stock Forecasting Using Multichannel CNN and Firefly Algorithm. Proceedings of the 2nd International Conference on Cognitive and Intelligent Computing. 2023; 447-458. doi: 10.1007/978-981-99-2742-5_46.
31. Omer ZT, Abbas AH. Image anomalies detection using transfer learning of ResNet-50 convolutional neural network. Indonesian Journal of Electrical Engineering and Computer Science. 2022; 27(1): 198-205.
32. Korade NB, Salunke MB, Bhosle AA*, et al.* Strengthening Sentence Similarity Identification through OpenAI Embeddings and Deep Learning. International Journal of Advanced Computer Science and Applications. 2024. 15(4):821-829.
33. Arshad, S, Yaqoob, M, Mehmood T. The Performance of GPT-3.5 in Summarizing Scientific and News Articles. Data Science and Emerging Technologies. 2023; 191:49–6.
34. Lalmi D, Bouaraour K, Benseddik A, Badji A, Bensaha H, Kifouche A, Khodja K. Enhancing predictive accuracy of inside temperature and humidity in an agricultural greenhouse using data-driven modeling with artificial neural networks. Indian Journal of Engineering. 2024; 21(55):1-15.
35. Korade NB, Zuber M. Forecasting Stock Price Using Time-Series Analysis and Deep Learning Techniques. Data Engineering and Applications: Proceedings of the International Conference. 2024;1146. doi: 10.1007/978-981-97-0037-0_31.
36. Shukla AK, Shukla A, Singh R. Neural networks based face recognition system for biometric security. Indian Journal of Engineering. 2023; 20(53):1-9.