Solanaceae Safeguard: Cnn-Swin Fusion for Precision Disease Management

Original Scientific Paper

Jaferkhan. P *
Noorul Islam Centre For Higher Education
Tamilnadu, India
jpskhan@gmail.com

V. Amsaveni
Noorul Islam Centre For Higher Education
Tamilnadu, India
amsaveni.v78@gmail.com

*Corresponding author

Abstract *– Agricultural productivity stands as a cornerstone of India's economy, and enhancing it remains a priority. A pivotal strategy in bolstering agricultural output is the timely identification of diseases. In agriculture, disease detection and management are crucial for ensuring crop health and yield. This study proposes a novel disease detection system for Solanaceae Vegetables utilizing a hybrid deep learning approach. The system integrates SWIN Transformer architecture with Convolutional Neural Networks (CNN) to analyze and classify disease patterns in Solanaceae vegetables. The dataset used for training and evaluation is sourced from Kaggle repository, comprising comprehensive images of diseased and healthy Solanaceae plants. Through extensive experimentation, the proposed hybrid model achieves a remarkable classification accuracy of 96%. The model demonstrated high precision, recall, and F1-scores across most classes, such as Class 0 (0.92, 0.89, 0.91) and Class 14 (0.97, 1.00, 0.99).The system's high accuracy demonstrates its potential as a reliable tool for disease detection and effective management strategies in Solanaceae vegetable cultivation, thereby contributing to enhanced leaf health and productivity.*

Keywords: *SWIN Transformer, Solanaceae vegetables, Convolutional Neural Networks, Agricultural productivity*

Received: July 21, 2024; Received in revised form: September 10, 2024; Accepted: October 8, 2024

1. INTRODUCTION

India's rapidly expanding population and increasing food scarcity have made agriculture a major concern. Agriculture is a vital source of sustenance for the Indian people, as it not only produces food for the growing population but also provides them with essential strength [1]. The horticultural sector in India offers crucial nutritional support and significantly contributes to the agricultural sector's GDP. Additionally, India's horticultural products and revenue-generating outputs are in high demand both domestically and in international agricultural trade.Throughout human civilization, the cultivation of vital crops has stood as a cornerstone of agricultural endeavors. Seasonal changes, the composition of soil, and a plethora of environmental variables collectively shape the performance of agricultural production, with any alterations therein invariably leading to diminished yields [2]. Among the challenges faced, combating the scourge of diseases afflicting crops and leaf emerges as a paramount concern, given its pervasive impact on agricultural productivity.

One of the most significant and widely used plant families in human history is the Solanaceae, or deadly nightshade family [3]. Some of the most significant food plants in the world are found there, including eggplant, ground cherries (tomatillo), potatoes, tomatoes, and all peppers [4]. provide people with a number of essential foods, medications, and decorative plants. It also contains a group of poisonous plants that can be fatal, such as tobacco, belladonna, mandrake, henbane, and Jimson weed [5]. They Fig. 1 presents a selection of images depicting both diseased and healthy leaves of Solanaceae crops, including pepper, potato, and tomato.

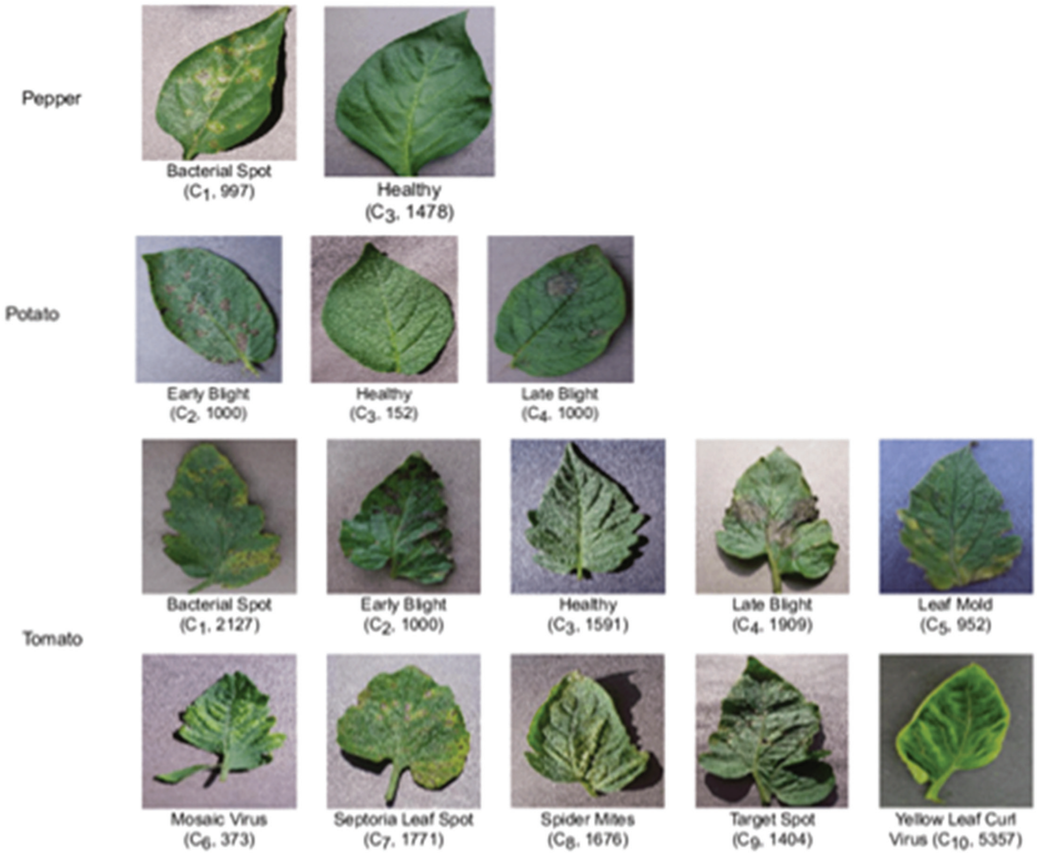


Fig. 1. Diseased and healthy leaf images of pepper, potato and tomato

Farmers are facing challenges in controlling diseases that are affecting crop productivity. Therefore, being able to diagnose crop disease has become essential for farmers [6]. Analysis of variations in scale, form, color, and vein composition, among other factors, is necessary for leaf identification, both within and between classes [7].Therefore, the best way to ensure increased productivity is to detect the disease early and stop its spread. Agriculture experts, researchers, and investigators are therefore very concerned about automated illness identification, diagnosis, classification, and recommendation of preventive measures [8]. The major contributions of the work are listed as follows

- Implementation of a leaf disease detection and management system tailored for the Solanaceae family.
- Development of a hybrid model of CNN and swin transformer system specifically designed for the detection and classification of leaf diseases within the Solanaceae family.
- Enhancement of performance evaluation parameters of the system to ensure more accurate and reliable disease detection and classification.

2. RELATED WORKS

In their study, Hidayah et al. [9] focused on utilizing CNN architecture for object detection in Solanaceae crops to aid robot vision. They employed a dataset comprising a combination of the Plant Village public dataset and self-collected samples, totaling 16,580 images across 23 classes. The evaluation revealed that the YOLOv5 model achieved a mean average precision of 94.2%, outperforming Scaled-YOLOv4. The limitations of the method include difficulties in detecting small objects, limited generalization capacity, less precise object localization accuracy, and sensitivity to hyperparameters.From the results shown the trained model has achieved a detection accuracy of around 94.12%.

Ojo and Zahid [10] explored the detection of bacterial wilt disease. Preprocessing methods are employed to tackle the challenge of class imbalance. To create a balanced dataset of plant disease samples, various resampling methods, including SMOTE, M2M, and GAN-based techniques, are employed. Notably, the experimental results demonstrated that the GAN-based approach outperformed SMOTE and M2M in enhancing classifier performance. The method achieved an average classification accuracy of 91.69% and an average F1-score of 91.62%. Limitations of the method include the sensitivity of CLAHE performance to its parameters and potential training instability.

Khalid et al. [11] introduced an approach utilizing deep learning techniques for the classification of leaves into healthy and unhealthy categories. The initial phase of the work involved the creation of a dataset comprising images of money plant leaves, which were then divided into two primary groups. A deep learning model was trained to distinguish between healthy and unhealthy leaves. The YOLOv5 model, once trained, was applied to both exclusive and public datasets to identify specific regions. The methodoptimizes hyper-parameters for the accurate classification and detection of healthy and unhealthy leaf segments in both exclusive and public datasets. The model, once trained, achieved a 93% accuracy on the test set. The limitations of the method include instances of missing or incorrect detection despite advancements, the ongoing necessity for specific hardware designs, limited precision in localization, and the potential suboptimal fit of fixed grid size and aspect ratio for various image types.

Ilyas et al. [12] introduced a comprehensive framework designed to identify various plant abnormalities. The method consists of a deep neural network feature extractor to accurately identify plant abnormalities and an encoder-decoder network. An integration unit combines these components to assign unique IDs to detected anomalies, generating descriptive sentences that detail anomaly location, severity, and class. The algorithm achieved a precision of 91.7% for abnormality detection. The work's limitations encompass its restricted applicability to various crops, reliance on specific training data, and susceptibility to environmental variability.

Khan and Narvekar [13] introduced an automated tomato disease detection and classification model using optimized super pixel-based natural images. Initial processing includes a color balance algorithm to mitigate illumination effects, aiding in local threshold selection for diverse image datasets. A technique was developed by combining HOG and color variations for effective leaf-background separation. Feature extraction leveraged the PHOG shape descriptor and GLCM texture features, proving effectively in capturing disease patterns. Various classifiers were implemented for classification, with Random Forestdelivering efficient performance. Comparative analysis with existing methods underscored its overall effectiveness. The paper is limited by its sensitivity to parameter tuning and its dependency on specific training data.Results indicate that the method achieved an accuracy of 93.12%.

Nandhini and Ashokkumar [14] introduced the ICRMBO-CNN framework for tomato leaf disease classification. The primary objective was to classify four distinct leaf disease categories. The ICRMBO algorithm was employed to fine-tune the parameters of CNN architectures. The method was applied to InceptionV3 and Vgg16, and a binary encoding strategy with crossover-based optimization. Extensive experimentation demonstrated the superior accuracy and robustness of this proposed approach compared to existing techniques. Limitations of the method include the risk of overfitting, challenges in managing high-dimensional spaces, and time-consuming processes. Magaña-Álvarez et al. [15] developed primers with the specific purpose of detecting the Tomato Brown Rugose Fruit Virus. Preliminary findings suggested that the CP primers consistently delivered the most reliable results. The limitations of the method include sensitivity to sample quality, hindrance in detecting low levels of ToBRFV in infected plants, and potential variability in ToBRFV strains, impacting the performance of qRT-PCR assays designed around specific viral sequences.

Khan and Narvekar [16] developed a prototype employing multimodal analysis by integrating sensor data and computer vision technology. The primary goal of this system is to enhance the precision of disorder detection in tomato plants by utilizing a combination of IoT, Machine Learning, Cloud Computing, and Image Processing. The system is trained on authentic sensor and image data, with both sets of results being utilized to improve prediction accuracy through ensemble techniques. The limitations of the method encompass integration complexity, privacy and cybersecurity concerns associated with collecting and sharing sensitive data from IoT devices, and a lack of generalization ability when deployed in new environments with varying growing conditions or disease patterns.

2.1. RESEARCH GAP

Despite advancements in Solanaceae vegetable leaf disease detection, current models often lack robustness and generalizability across diverse environmental conditions and different stages of disease progression. Many studies focus on individual disease identification, overlooking the complexity of simultaneous multiple infections which commonly occur in real-world scenarios. Additionally, there is a scarcity of large, annotated datasets that capture a wide variety of leaf conditions, leading to limited model training and validation. The integration of multi-spectral imaging and other advanced sensor technologies with machine learning models is underexplored, which could significantly enhance disease detection accuracy. Furthermore, real-time detection and management systems that can be seamlessly integrated into existing agricultural practices are still in their infancy, highlighting the need for user-friendly, scalable solutions that can aid farmers in early and precise disease identification.

3. MATERIALS AND METHODS

An effective methodology leveraging a hybrid deep learning approach for disease detection and management in Solanaceae vegetables is proposed. The data is sourced from the publically available Kaggle depository. Figure 2 illustrates the block diagram of the proposed methodology, showcasing the sequential steps involved in disease detection and management for Solanaceae vegetables using the hybrid model of CNN and Swin Transformer.Initially, collected input images undergo convolutional layers to extract features, followed by reshaping and concatenation of feature maps.

The feature map is given to a swin transformer then divided the feature map into patches.Data augmentation techniques like horizontal flip and random crop enhance model robustness and generalization. Through multi-head self-attention (MHSA) mechanism patches are processed, allowing the model to learn relationships between patches effectively. Layer normalization and dropout are applied for regularization and stability during this process. Subsequently, multi-layer perceptron (MLP) blocks capture complex nonlinear relationships within and between patches. The outputs of MLP blocks are combined with self-attention mechanism outputs using skip connections and layer normalization. Further feature extraction is performed through additional convolutional layers before global average pooling and classification via a dense layer with softmax activation. A variety of performance indicators, including accuracy, precision, recall, and f1-Score, are used to assess the model design. demonstrating its effectiveness in disease detection.

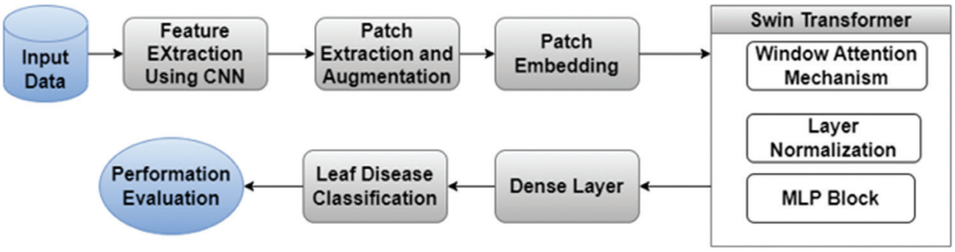

Input Data
Feature EXtraction Using CNN
Patch Extraction and Augmentation
Patch Embedding
Swin Transformer
Window Attention Mechanism
Layer Normalization
MLP Block
Dense Layer
Leaf Disease Classification
Performation Evaluation

Fig. 2. Block Diagram of Proposed Methodology

3.1. DATASET DESCRIPTION

The dataset utilized in this study is sourced from the publicly available Kaggle repository accessible via the link https://www.kaggle.com/datasets/emmarex/plantdisease/data. This dataset comprises images representing distinct leaf diseases affecting Solanaceae crops, including Tomato Spider Mites Two-Spotted Spider Mite, Tomato Early Blight, Pepper Bell Bacterial Spot, Tomato Late Blight, Potato Late Blight, Tomato Bacterial Spot, Tomato Leaf Mold, Tomato Target Spot, Tomato Yellow Leaf Virus, Tomato Mosaic Virus Potato Early Blight,Tomato Septoria Leaf Spot. Fig. 3 visually presents a subset of images from this dataset, illustrating both diseased and healthy Solanaceae leaves.

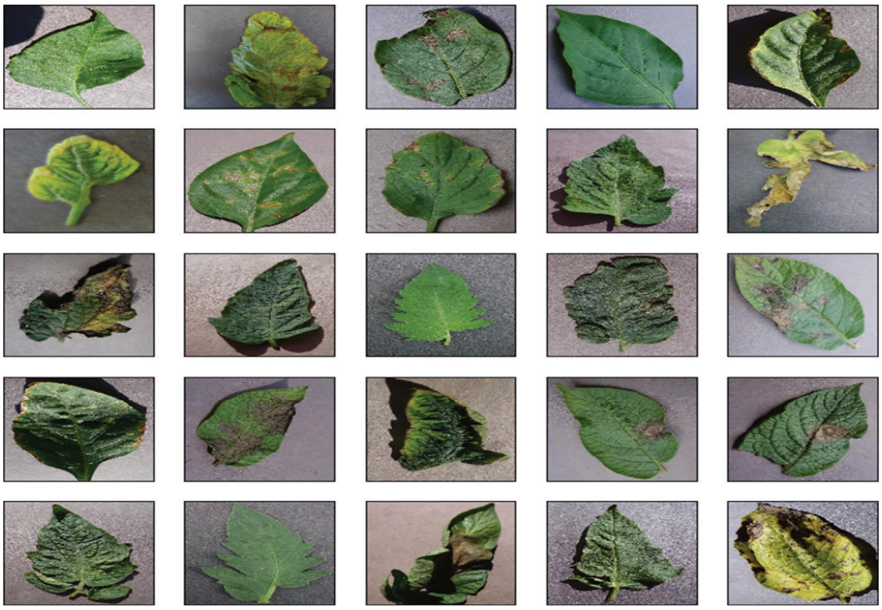


Fig. 3. Sample images from the dataset

Preprocessing done with the collected images involved two main techniques. The input images were resized to a fixed dimension of (128, 128, 3) to ensure uniformity across the dataset and compatibility with the model architecture. Additionally, pixel values were normalized, typically scaling them between 0 and 1, to enhance the model's convergence during training. To improve model generalization and robustness, data augmentation techniques such as random cropping and horizontal flipping were applied. Random cropping helps the model learn from different parts of the image, while horizontal flipping introduces variations in orientation, ensuring the model doesn't overfit to specific patterns present in the training data. Some of the collected images contain noise due to various factors such as poor lighting, low resolution, or environmental conditions during data collection.

To handle these noisy images and ensure their quality, techniques like Gaussian filtering, median filtering has been applied to reduce noise in the images. These filters help smooth the image while preserving important details, ensuring that noise doesn't interfere with feature extraction during the convolutional layers.

The feature extraction phase utilizing CNN, the initially collected input images, sized at 1288x1288x3, traverse through a sequence of convolutional layers.CNN consists of an input layer, hidden layers and an output layer. Fig. 4 illustrates the basic architecture of CNN model.

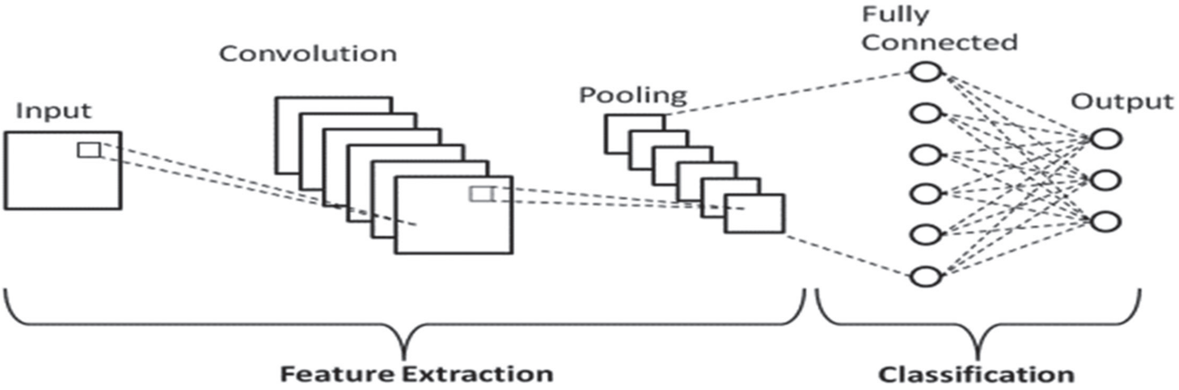


Fig. 4 Basic architecture of CNN

One or more convolutional layers are among the hidden layers in a CNN. This usually involves a layer that uses the layer's input matrix to perform a dot product of the convolution kernel. ReLU serves as the activation function. The convolution procedure creates a feature map as the convolution kernel moves along the layer's input matrix; this feature map then feeds into the input of the layer after it. Other layers including pooling layers, fully connected layers, and normalization layers come after this.The convolution operation is represented by

$$(I * K)(x, y) = \sum_{\mathrm{i}} \sum_{\mathrm{j}} I(i, j) \cdot K(x - i, y - j) \quad (1)$$

Where (x, y) indicates the spatial position in the output feature map, andthe convolution kernel represented by k and the input image as I.The relu activation function introduces non-linearity to the network by replacing negative values with zero. And it is represented by

$$\mathrm{f(x)} = \max(0, \mathrm{x}) \quad (2)$$

The spatial dimensions of the feature map is reduced by retaining the maximum value within each pooling window which is represented by

$$\mathrm{MaxPooling}(x,y)=\max(Ix,y) \quad (3)$$

The fully connected layer is represented by

$$\mathrm{y=Wx+b} \quad (4)$$

Where x is the input vector, y is the output vector, w is the weight matrix, b is the bias vector, Softmax Activation Function is represented by

$$\mathrm{Softmax(xi)} = \frac{\mathrm{e}^{\mathrm{x_i}}}{\sum_{\mathrm{j}} \mathrm{e}^{\mathrm{x_i}}} \quad (5)$$

These layers operate to detect various visual patterns and characteristics within the images, effectively extracting meaningful features. These convolutional layers are designed to capture low to mid-level features in the image, employing learnable filters, activation functions such as ReLU, and pooling layers like MaxPooling to reduce dimensionality and extract dominant features from the multi-channel arrays representing the input images. Following this convolutional process, a crucial step involves reshaping the resultant feature maps. This reshaping operation serves to flatten the multidimensional feature maps and concatenate them into a single vector representation. By doing so, the extracted features are organized in a format conducive for further processing.

3.2. THE SWIN TRANSFORMER

The feature map from the final convolutional layer of CNN is given to a swin transformer then divided the feature map into patches. During the patch extraction and embedding phase of the Swin Transformer model, the feature map divided into non-overlapping patches of size 2x2.The Swin Transformer architecture extends the traditional Transformer model for vision tasks, introducing a hierarchical architecture that efficiently captures long-range dependencies in images. Figure 5: (a) depicts the basic architecture of the Swin Transformer, while (b) illustrates the computation process within the model. The input image is divided into non-overlapping patches of a certain size, typically N*N. Let X denote the input image, and Pi denote the ith patch. Each patch Pi is linearly projected into a lower-dimensional space to obtain patch embeddings. This projection is represented by a learnable weight matrix Wpatch.

$$\text{Patch Embedding(Pi)} = \text{Pi.Wpatch} \tag{6}$$

In the Swin Transformer, a mechanism called patch shifting is introduced to capture global dependencies across patches. Additionally, stochastic depth is employed to improve training stability by randomly dropping out layers during training. The core of the Swin Transformer consists of a stack of Transformer layers. Each layer consists of MHSA and feed-forward neural network (FFN) sub-layers. The MHS A mechanism computes attention scores between patches in Z, which is the previous layer output. These scores are then used to aggregate information across patches. Given Z, the attention output A is computed as follows:

$$\mathrm{A} = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

Query, key, and value matrices obtained from Z are represented by Q, K, and V are the, the dimension of the key vectors denoted by , softmax is the function applied along the patch dimension.

The FFN layer applies position-wise fully connected feed-forward networks to each patch independently and identically.

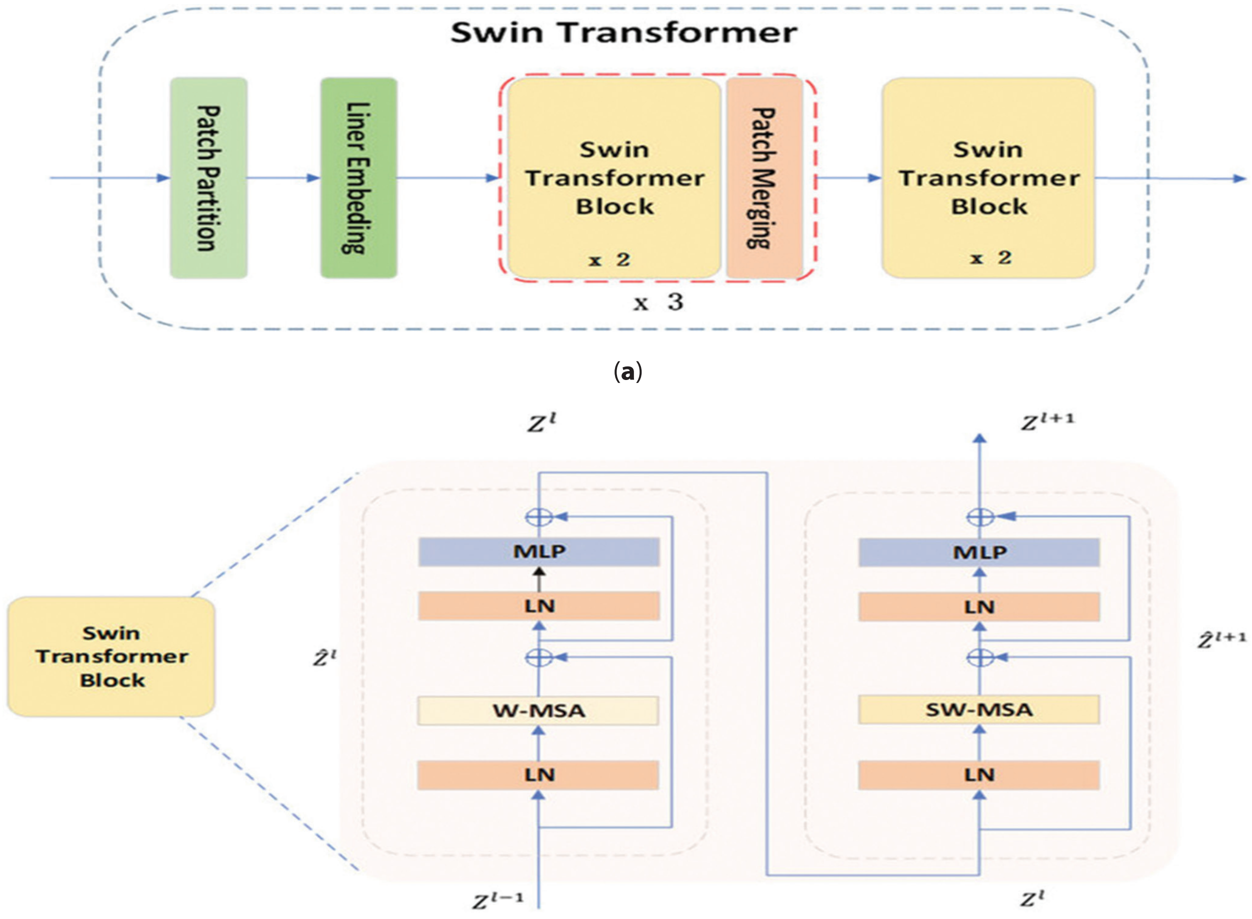
It is typically composed of two linear transformations followed by a non-linear activation function like ReLU.

Around each sub-layer, layer normalization and residual connections are applied to enhance regularization and stability:

$$LayerNorm\ (x+Sublayer(x)) \tag{8}$$

Where x represents the input to the sub-layer, and Sub layer represents either the MHS A or FFN sub-layer. After the transformer layer, the sequence of patch embeddings is aggregated, typically through mean pooling or another aggregation mechanism.

The aggregated representation is then fed into a multi-layer perceptron (MLP) head for classification.



(a)

(b)

Fig. 5. (**a**) Basic block diagram of Swin Transformer (**b**) Computation Process

Our approach divides the feature map from the last convolutional layer into non-overlapping patches of size 2x2 during the patch extraction and embedding stage. Subsequently, data augmentation techniques like random crop and horizontal flip are applied to bolster the model's robustness and generalization. Each patch undergoes linear projection into a lower-dimensional space, resulting in patch embeddings. These embeddings are then reshaped into sequences to be inputted into the subsequent Transformer layers.In the Swin Transformer, the transfer block constitutes a stack of transformer layers, each comprising multiple attention heads that compute attention scores between patches. These scores facilitate the aggregation of information across patches, thereby capturing global dependencies. Within each layer, the main sub-layers include the MHSAlayer, which learns linear relationships between patches, and the FFN layer, which applies position-wise fully connected feed-forward networks to each patch independently and identically. Layer normalization and residual connections are applied around each sub-layer to ensure regularization and stability. Additionally, features undergo further processing through multi-layer perceptron (MLP) blocks to capture nonlinear relationships which consist of dense layers with ReLU activation. These Transformer layers are pivotal in capturing long-range dependencies and relationships between patches.

Shifted Window Transformer blocks implement a mechanism for down sampling, which involves shifting the window for self-attention in the spatial dimensions. By doing so, the Swin blocks effectively reduce the spatial resolution of the feature maps while increasing the number of channels, achieving down sampling in a computationally efficient manner. This unique approach allows the Swin Transformer to capture hierarchical information across different scales while maintaining computational scalability.

Table 1. Parameters of proposed Swin Transformer model

Parameter	Values
Patch Size	(2,2)
Image_dimension	128
Label_smoothing	0.1
num_mlp	64
Learning rate	1e-3
Dropout rate	0.2
num_heads	8
Embed_dim	64
Shift_size	1
Window_size	2
Batch size	32
Weight decay	0.0001
Total Parameters	1172527
Trainable Parameters	1172527
Non- Trainable Parameters	0

In the final classification step, the outputs of the MLP blocks and the self-attention mechanism are combined through skip connections and layer normalization. The resulting feature maps undergo reshaping and additional convolutional layers before being globally pooled. This pooled representation is then fed into a dense layer for classification. Following this, the model utilizes a softmax activation function to output class probabilities based on the logits obtained from the classification head. These logits represent the raw predictions for each class, and the softmax function is applied to derive the final class probabilities.

Effectiveness of Convolutional Layers progressively captured different levels of abstraction, from low-level edges and textures to high-level structures and patterns. The multi-head self-attention mechanism allowed the model to capture spatial relationships between different patches of the image. This significantly improved the model's ability to focus on important regions, enhancing classification performance. The combination of convolutional layers, self-attention mechanisms, and MLP blocks helped the model learn both local and global features from the input images, leading to better overall classification performance.

3.3 HARDWARE AND SOFTWARE SETUP

The model was created and trained on Google Collaboratory, where the entire process was completed using Python and TensorFlow. The hardware setup primarily consisted of a system equipped with a high-performance processor and GPU to efficiently execute the computational tasks involved in training and evaluating the deep learning models. An advanced processor, Intel Core i9 or AMD Ryzen was employed to handle the computational load effectively. A powerful GPU, NVIDIA GeForce RTX, was utilized to accelerate the training of deep neural networks, which typically involves intensive matrix operations. The optimizer utilized during training is Adam, and the loss function employed is categorical crossentropy.

A batch size of 32 samples per iteration is utilized for training, and the training process is conducted over 100 epoch. The software stack would have included Python as the primary programming language due to its widespread adoption and extensive libraries for machine learning and deep learning tasks. TensorFlow on Google Collaboratory, a cloud-based platform offering free access to GPU resources, facilitated collaborative coding and experimentation with deep learning models in a web-based environment.

4. RESULT AND DISCUSSION

The performance of the model is evaluated through the following parameters: precision, recall, accuracy, and F1-score.These metrics provide insights into the model's ability to correctly classify instances and handle imbalances between classes.

$$\text{Accuracy} = \frac{\text{True}_{\text{Pos}} + \text{True}_{\text{Neg}}}{\text{True}_{\text{Pos}} + \text{True}_{\text{Neg}} + \text{False}_{\text{pos}} + \text{False}_{\text{Neg}}} \quad (8)$$

$$\text{Precision} = \frac{\text{True}_{\text{Pos}}}{\text{True}_{\text{Pos}} + \text{False}_{\text{pos}}} \quad (9)$$

$$\text{Recall} = \frac{\text{True}_{\text{Pos}}}{\text{True}_{\text{Pos}} + \text{False}_{\text{Neg}}} \tag{10}$$

$$\text{F1} - \text{Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

These metrics collectively offer a comprehensive understanding of the model's performance and are essential for evaluating its effectiveness in various scenarios. An accuracy plot visualizes the model's performance over time by showing the accuracy on the training and validation datasets for each epoch. Fig. 6 displays the accuracy plot of the proposed model.Initially, the model starts with a low accuracy of around 6-7% in the first few epochs but quickly learns and improves to approximately 38% by Epoch 10. As training progresses, the accuracy continues to rise, reaching about 66% by Epoch 20 and 78-80% by Epoch 30. This consistent improvement reflects the model's increasing ability to extract and understand relevant features from the data.In the advanced training phases, accuracy surpasses 85% by Epoch 40 and reaches around 94% by Epoch 50. The final epochs show the model achieving nearly perfect accuracy, hovering around 98-99% and ultimately nearing 100% by Epoch 100. Minor fluctuations in accuracy, especially noticeable in epochs 81 and 93, are typical as the model fine-tunes its parameters. These results indicate that the hybrid model is highly effective for disease detection in Solanaceae vegetables.

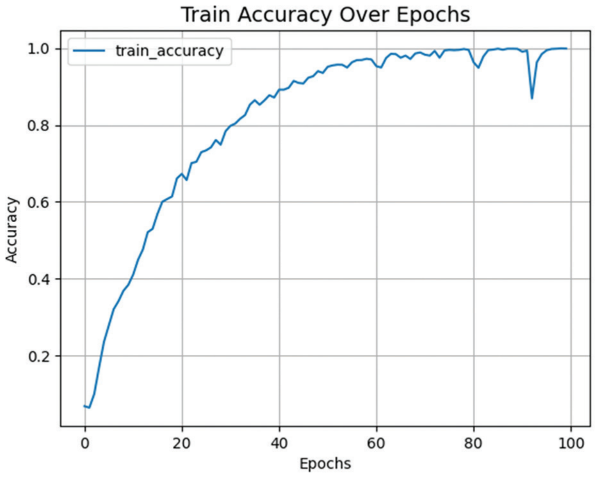


Fig. 6. Accuracy Plot of Proposed Model

A loss plot displays how the loss function, which measures the error between predicted and actual values, changes over each epoch during the training process. Fig. 7 displays the loss plot of the proposed model, highlighting the changes in loss over the training and validation phases.In the initial epochs, the model exhibits high loss of 2.7134 as it begins to learn from the dataset. By the final epochs, loss significantly decreases to 0.6476 at Epoch 99, and accuracy increases dramatically , indicating effective learning and optimization. Throughout training, loss generally decreases, but fluctuations occur, such as a slight increase from 1.3765 at Epoch 21 to 1.4065 at Epoch 22, due to factors like learning rate adjustments, data variability, and complexity of patterns. Despite these fluctuations, the model ultimately achieves stable, low loss and high accuracy, demonstrating successful learning.

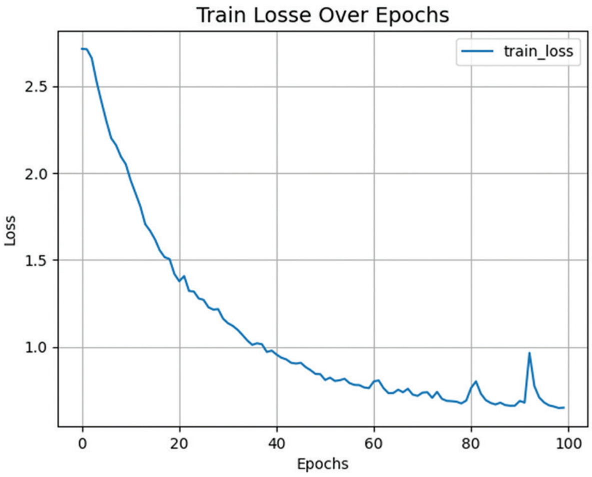
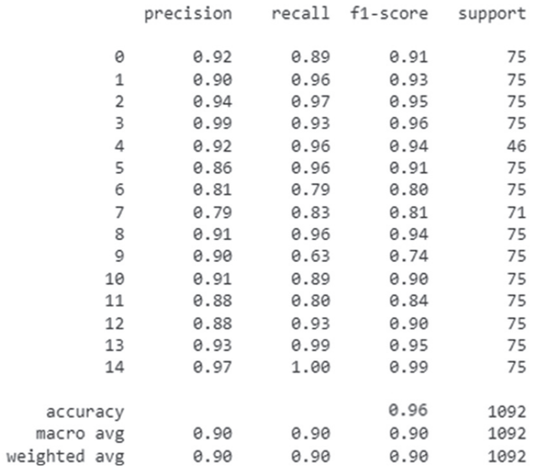


Fig. 7. Loss Plot of Proposed Model

The classification report shows the model's performance on a multi-class classification task, achieving an overall accuracy of 96%. Fig. 8 presents the classification report, summarizing the performance metrics of the proposed model, including precision, recall, and F1-score for each class.Key metrics for each class include precision, recall, F1-score, and support. Most classes have high precision, recall, and F1-scores, such as for Class 0 (0.92, 0.89, 0.91) and Class 14 (0.97, 1.00, 0.99), indicating accurate predictions. However, Class 6 (0.81, 0.79, 0.80) and Class 9 (0.90, 0.63, 0.74) show lower



```
              precision    recall  f1-score   support

           0       0.92      0.89      0.91        75
           1       0.90      0.96      0.93        75
           2       0.94      0.97      0.95        75
           3       0.99      0.93      0.96        75
           4       0.92      0.96      0.94        46
           5       0.86      0.96      0.91        75
           6       0.81      0.79      0.80        75
           7       0.79      0.83      0.81        71
           8       0.91      0.96      0.94        75
           9       0.90      0.63      0.74        75
          10       0.91      0.89      0.90        75
          11       0.88      0.80      0.84        75
          12       0.88      0.93      0.90        75
          13       0.93      0.99      0.95        75
          14       0.97      1.00      0.99        75

    accuracy                           0.96      1092
   macro avg       0.90      0.90      0.90      1092
weighted avg       0.90      0.90      0.90      1092
```

Fig. 8. Classification Report

Performance, highlighting areas for improvement. The macro and weighted averages for precision, recall, and F1-score, reflecting balanced and effective overall performance. Fig. 9 presents the confusion matrix, illustrating the true versus predicted classifications of the proposed model for each class. Table 2 showcases a comparison between the proposed model with the existing approaches, highlighting key performance metric such as accuracy.

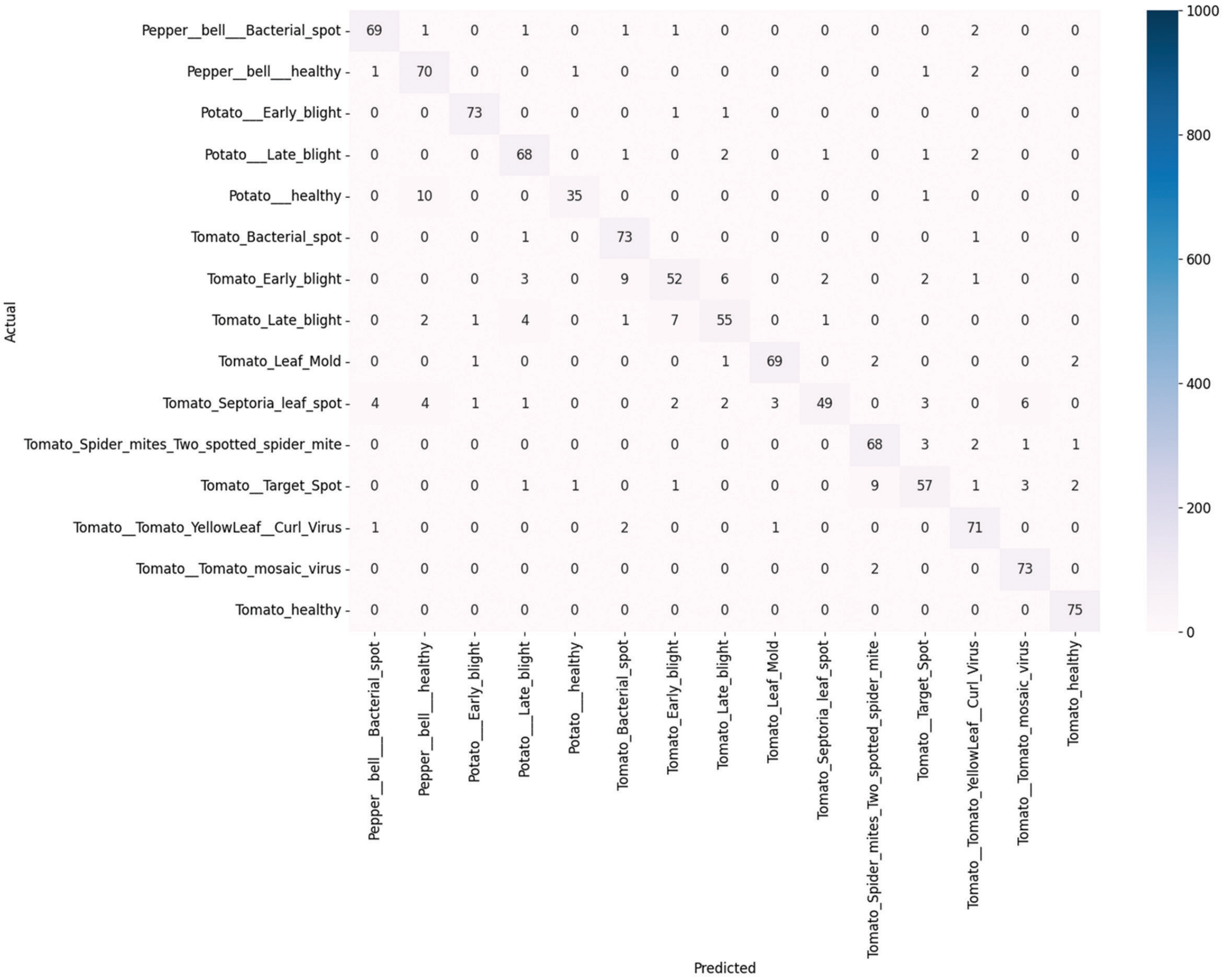


Fig. 9. Confusion Matrix

The table presents a comparison of different methodologies and their respective accuracies, highlighting the performance of various models, including the proposed method. Specifically, Hidayah et al. used CNN and achieved an accuracy of 94.2%. Mahnoor Khalid et al. employed the YOLOv5 model and obtained an accuracy of 93%. Ilyas et al. developed a deep learning-based hybrid model and reached an accuracy of 91.7%. Saiqa Khan and Meera Narvekar utilized a deep learning method and reported an accuracy of 93.12%. The proposed method, which is a hybrid model of CNN and Swin transformer, achieved the highest accuracy of 96%. This comparison demonstrates that the proposed hybrid model outperforms other models listed, showcasing its superior accuracy in disease detection and classification for Solanaceae Vegetables.

Table 2. Comparison of the proposed model with the existing approaches

Author	Methodology	Accuracy
Hidayah et al	Convolutional Neural Network	94.2%.
Mahnoor Khalid et al.	YOLOv5 model	93%
Ilyas et al	Deep learning-based Hybrid model	91.7%
Saiqa Khan et al Meera Narvekar	Deep learning method	93.12%
Proposed Method	Hybrid model of CNN and Swin transformer	96%

The proposed method addresses the complexity of simultaneous multiple infections in the current work by leveraging a multi-head self-attention mechanism that enables the model to attend to different regions of the leaf image simultaneously, capturing the relationships between various disease-affected areas. Additionally, the combination of convolutional layers and MLP blocks allows the model to extract both local and global features, enhancing its ability to detect and classify multiple infections occurring concurrently in different parts of the leaf.

5. CONCLUSION

Detection of plant leaf diseases is a critical issue in agriculture, impacting yield and crop production significantly. Solanaceae vegetables, including tomatoes, potatoes, and peppers, are vital components of the global food supply, playing a crucial role in ensuring food security and meeting global nutritional needs. This research proposes a hybrid deep learning-based system for the early and accurate identification of leaf diseases in Solanaceae vegetables, leveraging a CNN-Swin Transformer model. The integration of deep learning techniques into leaf disease detection systems allows for the development of smart, automated solutions capable of continuous monitoring and assessment of crop health. The proposed model has been evaluated on the Plant Village dataset and has demonstrated superior performance of 96% accuracy, highlighting its potential for enhancing agricultural productivity and sustainability. The major contribution of this work lies in the hybrid integration of convolutional layers, multi-head self-attention mechanisms, and MLP blocks, creating a balanced architecture that effectively captures both local and global features, unlike traditional Swin Transformers which rely solely on patch-based self-attention. This approach combines the strengths of CNNs in capturing fine-grained local details and self-attention mechanisms for global context, offering a novel solution with improved feature extraction and regularization for image classification tasks, including plant disease detection, while being computationally more efficient than full transformer-based models. This advanced detection capability can significantly aid farmers in implementing timely and targeted interventions, enhancing overall crop health and productivity. The findings of this study underscore the potential of hybrid deep learning models in developing smart, automated solutions for continuous crop health monitoring, thereby contributing to sustainable agricultural practices and improved food security. The computational complexity of integrating self-attention mechanisms with convolutional layers increases training time and resource requirements, limiting the scalability of the model for very large datasets.

5. REFERENCE

[1] S. K. Patel, A. Sharma, G. S. Singh, "Traditional agricultural practices in India: an approach for environmental sustainability and food security", Energy, Ecology and Environment, Vol. 5, No. 4, 2020, pp. 253-271.

[2] C. R. Barrett, "Overcoming global food security challenges through science and solidarity", American Journal of Agricultural Economics, Vol. 103, No. 2, 2021, pp. 422-447.

[3] S. Knapp, L. Bohs, M. Nee, D. M. Spooner, "Solanaceae—A Model for Linking Genomics with Biodiversity", Comparative and Functional Genomics, Vol. 5, No. 3, 2004, pp. 285-291.

[4] M. Añibarro-Ortega, J. Pinela, A. Alexopoulos, S. A. Petropoulos, I. C. Ferreira, L. Barros, "The powerful Solanaceae: Food and nutraceutical applications in a sustainable world", Advances in Food and Nutrition Research, Vol. 100, 2022, pp. 131-172.

[5] T. Chamroy, "Production Technology of Underutilized Vegetables of Solanaceae Family", Production Technology of Underutilized Vegetable Crops, Springer, 2023, pp. 151-161.

[6] B. Richard, A. Qi, B. D. Fitt, "Control of crop diseases through Integrated Crop Management to deliver climate-smart farming systems for low-and higinput crop production", Plant Pathology, Vol. 71, No. 1, 2022.pp. 187-206.

[7] M. Jiao, Y. Wang, F. Yang, Z. Zhao, Y. Wei, R. Li, Y. Wang, "Dynamic fluctuations in plant leaf interception of airborne microplastics", Science of The Total Environment, Vol. 906, 2024, p. 167877.

[8] J. A. Wani, S. Sharma, M. Muzamil, S. Ahmed, S. Sharma, S. Singh, "Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: Methodologies, applications, and challenges", Archives of Computational methods in Engineering, Vol. 29, No. 1, 2022. pp. 641-677.

[9] A. N, Hidayah et al. "Disease Detection of Solanaceous Crops Using Deep Learning for Robot Vision", Journal of Robotics and Control, Vol. 3, No. 6, 2022, pp. 790-799.

[10] M. O. Ojo, A. Zahid, "Improving Deep Learning Classifiers Performance via Preprocessing and Class Imbalance Approaches in a Plant Disease Detection Pipeline", Agronomy, Vol. 13, No. 3, 2023, p. 887.

[11] M. Khalid, M. S. Sarfraz, U. Iqbal, M. U. Aftab, G. Niedbała, H. T. Rauf, "Real-Time Plant Health Detection Using Deep Convolutional Neural Networks", Agriculture, Vol. 13, No. 2, 2023, p. 510.

[12] T. Ilyas, H. Jin, M. I. Siddique, S. J. Lee, H. Kim, L. Chua, "DIANA: A Deep Learning-Based Paprika Plant Disease and Pest Phenotyping System with

Disease Severity Analysis", Frontiers in Plant Science, Vol. 13, 2022, p. 983625.

[13] S. Khan, M. Narvekar, "Novel Fusion of Color Balancing and Superpixel Based Approach for Detection of Tomato Plant Diseases in Natural Complex Environment", Journal of King Saud University-Computer and Information Sciences, Vol. 34, No. 6, 2022, pp. 3506-3516.

[14] S. Nandhini, K. A. Kumar, "Improved Crossover Based Monarch Butterfly Optimization for Tomato Leaf Disease Classification Using Convolutional Neural Network", Multimedia Tools and Applications, Vol. 80, 2021, pp. 18583-18610.

[15] A. A. Magaña-Álvarez et al. "Detection of Tomato Brown Rugose Fruit Virus (ToBRFV) in Solanaceous Plants in Mexico", Journal of Plant Diseases and Protection, Vol. 128, 2021, pp. 1627-1635.

[16] S. Khan, M. Narvekar, "Disorder Detection of Tomato Plant (Solanum Lycopersicum) Using IoT and Machine Learning", Journal of Physics: Conference Series, Vol. 1432, No. 1, 2020, p. 012086.