Identifying Selected Diseases of Leaves using Deep Learning and Transfer Learning Models

Afsana Mimi[1], Sayeda Fatema Tuj Zohura[1], Muhammad Ibrahim[2*], Riddho Ridwanul Haque[1], Omar Farrok[3], Taskeed Jabid[1], Md Sawkat Ali[1]

[1] *Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh*
[2] *Department of Computer Science and Engineering, University of Dhaka, Dhaka, Bangladesh*
[3] *Ahsanullah University of Science and Technology, Dhaka, Bangladesh*

*alim@ewubd.edu, afsana.mimi0304@gmail.com, syedafatema58@gmail.com, *ibrahim313@du.ac.bd (corresponding author), riddhohaque@gmail.com, omar.eee@aust.edu, taskeed@ewubd.edu, alim@ewubd.edu*

Abstract. Leaf diseases may harm plants in different ways, often causing reduced productivity and, at times, lethal consequences. Detecting such diseases in a timely manner can help plant owners take effective remedial measures. Deficiencies of vital elements such as nitrogen, microbial infections and other similar disorders can often have visible effects, such as the yellowing of leaves in *Catharanthus roseus* (bright eyes) and scorched leaves in *Fragaria*×*ananassa* (strawberry) plants. In this work, we explore approaches to use computer vision techniques to help plant owners identify such leaf disorders in their plants automatically and conveniently. This research designs three machine learning systems, namely a vanilla CNN model, a CNN-SVM hybrid model, and a MobileNetV2-based transfer learning model that detect yellowed and scorched leaves in *Catharanthus roseus* and strawberry plants, respectively, using images captured by mobile phones. In our experiments, the models yield a very promising accuracy on a dataset having around 4000 images. Of the three models, the transfer learning-based one demonstrates the highest accuracy (97.35% on test set) in our experiments. Furthermore, an Android application is developed that uses this model to allow end-users to conveniently monitor the condition of their plants in real time.

Keywords: convolutional neural network, transfer learning, leaf disease detection, image classification

1. Introduction

Visually observing the condition of the leaves of a plant is a good way to monitor its health and well-being. Yellowing and scorching of leaves are often symptoms of serious underlying conditions such as the deficiency of vital elements like nitrogen, surplus of chloride particles [14], or microbial infections caused by viruses, bacteria and fungi [25]. When such changes occur, it becomes necessary to identify and address the underlying issues as soon as possible. Microbial infections may often be contagious [29] and hence an early detection and removal of infected plants and leaves is crucial. Unless infected plants and leaves are identified and isolated in a timely manner, the infections may spread through a large number of plants in the vicinity, leading to drastically reduced production and having dire financial consequences [28].

Scientific evidence suggests that *C. roseus* leaves can be used for medicinal purposes [20], while the ubiquitous demand for strawberry as a fruit has led to its widespread industrial production [5]. Thus, both these plants are commonly cultivated in gardens and plantations, quite often in large amounts. Hence, manually monitoring these leaves for possible diseases is not a trivial task for gardeners and farmers. Automated tools can help in continuous monitoring of the health of these plants and in giving warning their users about signs of malnutrition and infectious diseases. The proposed approach takes a step in this direction and identifies signs of yellowing in the leaves of *C. roseus* and scorching in Strawberry plants. Our proposed system leverages machine learning techniques to detect signs of disease in plant leaves from their images.

1.1. Motivation and contribution

Machine learning is increasingly being applied to sectors like economy [19], social well-being [23], and agriculture [3]. Developing machine learning, in particular, deep learning models to identify leaf defects has been an area of interest in computer vision research for many years [8]. Despite this, to the best of our knowledge, there remains a lack of computer vision approaches that specifically focus on the detection of leaf damage in Bright Eyes and Strawberry plants. With a specific focus on detecting signs of yellowing and scorching in the leaves of these plants, we design, train, validate and compare the efficacy of multiple deep learning approaches for classifying plant leaves based on their images. The three approaches that we investigate are based on: (1) Convolutional Neural Network (CNN), (2) a hybrid architecture that combines CNN and Support Vector Machine (SVM), and (3) a transfer learning-based approach in which a MobileNetV2 model [13] pre-trained on the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) dataset [17]. The experimental evaluation conducted on a real-world dataset consisting of both pre-existing and self-taken images shows the transfer-learning model to be the most effective among the three. The model is then loaded into an android application that facilitates easy offline access to plant owners around the world.

In this paper, a transfer-learning based approach is exploited for automated detection of yellow leaves in *Catharanthus roseus* (bright eyes) plants and scorched leaves in *Fragaria* × *ananassa* (strawberry) plants from camera images. The key contributions of this paper are as follows:

- A novel dataset consisting of 3804 images is developed, 2239 of which are manually captured by us.
- Three approaches are designed to identify yellow and scorched leaves in Bright Eyes and Strawberry plants respectively. The approaches are: a vanilla CNN model, a CNN-SVM hybrid model, and a MobileNetV2-based transfer learning model. We also examine the effect of varying the number of training epochs on the accuracy of each approach.

- An Android application is developed that facilitates gardeners and farmers to easily utilize the transfer learning model for identification of unhealthy leaves in their plants.

Of the three models, the transfer learning-based one demonstrates the highest accuracy in our experiments which is 97.35% accuracy on test dataset.

The remainder of this paper is organized as follows. In Section 2 we provide a review of the relevant existing literature. Section 3 provides an overview of the proposed approach and describes the dataset and models. Section 4 describes the experimental results. Section 5 concludes the paper with directions for future research.

2. Related works

Use of deep learning models for detecting plant diseases has been a major focus of computer vision research over the years [3]. Sladojevic et al. [24] propose an approach based on CNN for detecting a variety of plant diseases. Das et al. [9] employ SVM to recognize leaf diseases in plants. Regarding hybrid architectures, CNN and SVM have been used in tandem by Ahlawat et al. [2] to identify handwritten characters.

In parallel with such generalized approaches, other techniques that are tailored to suit applications on particular species of plants are also studied in the literature. Kurtulmucs et al. [16] use three deep learning architectures, namely AlexNet [15], GoogLeNet [26] and ResNet [12], to classify sunflower seed images. Chen et al. [7] successfully use deep learning models to identify diseases in rice plants with satisfactory accuracy. Mokhtar et al. [18] compare different kernel functions in SVM for detecting tomato leaf diseases. Zhong et al. [30] use deep CNN for detecting leaf diseases in apple trees. Amara et al. [4] use CNN for detecting leaf diseases in banana trees.

Often, pre-processing the images before the training phase is vital for increasing the accuracy of machine learning models. Extracting features from datasets before training and testing models for image classification has been shown to be useful for improving the accuracy of the trained models. For example, Tiwari et al. [27], improve the accuracy of logistic regression model by using transfer learning-based feature extraction for potato disease classification. A summary of such specialized approaches which are designed to identify leaf diseases in specific types of plants is given in Table 1.

As Table 1 shows, deep learning models trained specifically to detect diseases in certain types of plants have shown relatively high levels of accuracy. Despite this, we think that the benefits of deep learning and transfer learning models are not fully harnessed in the existing literature. This research aims to bridge this gap by designing and evaluating several such models to identify yellow and scorched leaves in Bright Eyes and Strawberry plants respectively.

Tab. 1. Existing approaches to identifying leaf diseases in different types of plants.

Author	Algorithm	Plant Name	Accuracy
Kurtulmucs et al. [16]	DCNN	Sunflower	95%
Chen et al. [7]	Deep transfer learning	Rice	98.63%
Mokhtar et al. [18]	SVM	Tomato	99.83%
Tiwari et al. [27]	Transfer learning	Potato	97.8%
Zhong et al [30]	DCNN	Apple	93.71%
Amara et al. [4]	CNN	Banana	92%

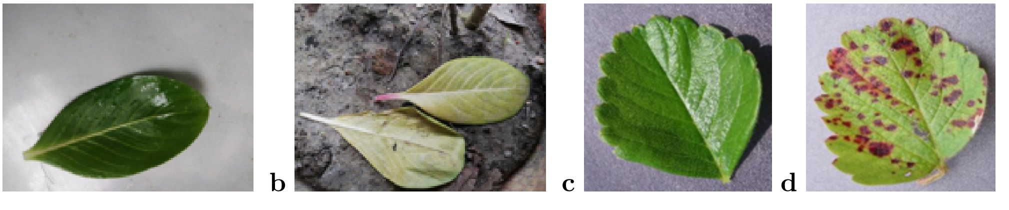
3. Learning models

As with most typical machine learning-based prediction systems, our employed models consist, broadly, of two phases: (1) dataset preparation and processing, and (2) training and testing the models. This section discusses these details.

We prepare the dataset by first acquiring images of Bright Eyes and Strawberry leaves, by manually taking photographs using mobile phone cameras (2239 images) and from publicly available online sources (1565 images). Some sample images are shown in Figure 1. The images are resized and then compiled into a single dataset. Afterwards, the images are partitioned into training and test sets. Then, three CNN-based models are trained on this dataset: a vanilla CNN, a hybrid of CNN and SVM, and a transfer learning model, specifically, the MobileNetV2 model pre-trained on ILSVRC2012 dataset. All models are trained using the images in the training set and used to classify images in the test sets. The flowchart of the data preparation process along with learning modules is shown in Figure 2.

3.1. Data collection and preparation

As mentioned above, the dataset used in the experiments consists of 3804 images of *C. roseus* and strawberry leaves, of which 2239 are captured by us, and the rest 1565



a **b** **c** **d**

Fig. 1. Sample images of leaves from the dataset: (**a**) green *C. roseus*; (**b**) yellow *C. roseus*; (**c**) healthy strawberry; (**d**) scorched strawberry.

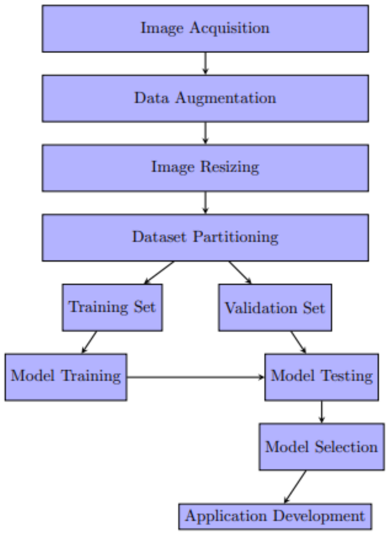


Fig. 2. Flow chart of the proposed methodology.

images are selectively chosen from the Internet. The images are then manually labeled by us into four types – green and yellow *C. roseus* leaves, and healthy and scorched strawberry leaves. Table 2 shows the number of images in each type of the dataset.

Data Augmentation

Since large training datasets help improve the accuracy of deep neural networks, we expand the size of the dataset using simple data augmentation techniques. Aside from increasing the number of training instances, data augmentation helps to diversify the dataset thereby allowing the trained model to be more robust to unnecessary variations and perturbations. All the images are re-scaled to 224×224 pixels before training using the ImageDataGenerator API provided in the Keras package [10]. These raw images are sheared, zoomed in, rotated and/or horizontally flipped. To do this, for every image of

Tab. 2. Statistics of different types of leaves in the dataset.

Image Type	Class Label	Number of Images
C. roseus Green	C0	1053
C. roseus Yellow	C1	1186
Strawberry healthy leaf	C2	456
Strawberry scorched leaf	C3	1109

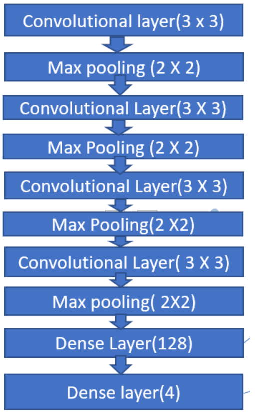


Fig. 3. Structure of employed vanilla CNN model (Model 1).

the dataset, we create three real-time tensor images using Keras platform: `shear_range`, `zoom_range`, `horizontal_flip`. However, the actual dataset does not contain these sheared, zoomed, and flipped images, rather we create those augmented images in real-time just before training.

All the images have three channels denoting red, green, and blue intensity levels of each of their pixels. Thus the total size of the dataset stands at 3804 images. The dataset is split into the training and test sets in 90-10 percentages (3426-378 images) respectively. We note here that in this paper we do not perform any hyper-parameter tuning using a separate validation dataset (which is left for future work). However, the platform we use for implementing the algorithms, namely Keras, uses the term *validation set* to indicate the test set. Therefore, throughout the paper we use the two terms *validation* and *test* interchangeably as there is no difference between these two terms in this paper.

3.2. Employed models

As mentioned earlier, in this investigation we utilize three machine learning models which are described below.

3.2.1. Model 1: CNN

The structure of the employed CNN model is illustrated in Figure 3 where a 3×3 convolutional layer followed by a 2×2 max-pooling layer are used. In total, four 3×3 convolutional layers and four 2×2 max-pooling layers are used. The popular activation function ReLU is used where a neuron is only activated when the output is greater than zero, so it does not activate all the neurons simultaneously. ReLU is popular because it reduces the chance of facing the vanishing gradient problem, and often achieves better performance. Mathematically, ReLU is defined as $y = \max(0, x)$.

At the later stage, a hidden dense layer of 128 neurons is used with ReLU activation. Finally, as loss function the SoftMax activation along with the categorical cross entropy is utilized at the dense layer because it is able to re-scale the output. The formula for SoftMax activation function is

$$\mathrm{SoftMax}(i) = \frac{e^i}{\sum_j e^j} ,$$

where j stands for the total number of neurons in the last layer. The model is compiled using a categorical cross-entropy loss function, to classify among multiple classes. The Adam optimizer is used to reduce the losses by its stochastic optimization approach.

The pseudo-code of the training phase of CNN is described below in Algorithm 1. The data preparation phase includes resizing the input image in 224×224 pixels and applying data augmentation to the input data.

Algorithm 1: CNN

```
begin
    for i ← 1 to 4 do
        pass the augmented and resized data into convolutional layer;
        pass the output of the last convolutional layer into max pooling layer;
    end
    pass the output of the previous layers into a flatten layer;
    pass the output of the flatten layer into a hidden later with 128 neurons;
    pass the output of the hidden layer using SoftMax activation function;
    calculate the loss using the categorical_crossentropy function;
end
```

Here is how it works: every input image is passed through four slices of convolutional and max pooling layers. Each slice consists of a 3×3 convolutional layer and a 2×2 max pooling layer. The output of these slices is passed through a flatten layer which converts the 2D array into a 1D array. After that the array is passed through a hidden layer consisting of 128 neurons. Finally, the output is passed through the final dense layer with 4 neurons.

3.2.2. Model 2: Hybrid of CNN and SVM

A hybrid of model consisting of an SVM with a mixture of a deep neural network is developed. Although there are some research on the hybrid CNN–SVM learning model in the existing literature, this research is still in an early phase, so it can be further improved by fine tuning its structure and parameters [21]. The features of input images are extracted using a CNN model and then fed into an SVM classifier in the last layer as

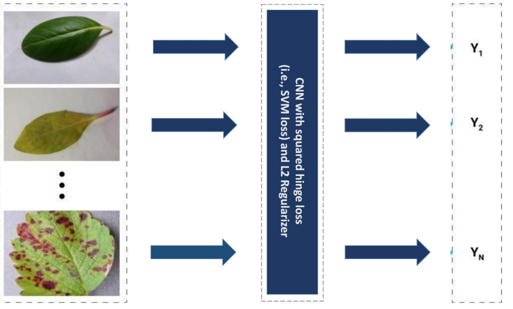


Fig. 4. Structure of hybrid CNN-SVM model (Model 2).

shown in Figure 4. The algorithm of the CNN-SVM training is described in Algorithm 2. The data preparation phase includes resizing the input image in 224×224 pixels and applying data augmentation to the input data.

Algorithm 2: Hybrid CNN-SVM

---

begin
    for $i \leftarrow 1$ **to** 4 **do**
        pass the augmented and resized data into convolutional layer;
        pass the output of the last convolutional layer into max pooling layer;
    end
    pass the output of the previous layers into a flatten layer;
    pass the output of the flatten layer into a hidden later with 128 neurons;
    pass the output of the hidden layer using SoftMax activation function;
    calculate the loss using the `squared_hinge` function (i.e., SVM loss) function with L2 regularizer;
end

Here is how it works: the images are passed through four slices like the previous CNN model. The output of these four slices are passed through a dense layer of 128 neurons. In this layer the ReLU is used as an activation function. In the last layer SoftMax activation function is applied.

The squared hinge loss function – also known as the SVM loss – is used, which enables us to draw a fine-tuned decision boundary between the classes. The formula of the squared hinge function is

$$\mathrm{L2}(y_1, y_2) = \sum_{i=1}^{n} \{\max(0, 1 - y_{1i} \cdot y_{2i})^2\}\,.$$

3.2.3. Model 3: Transfer Learning (MobileNetV2)

Transfer learning is a method through which the knowledge gained by training a model in one application area is utilized for classifying data in a different but related domain. The advantages of using pre-trained models include reduced training time and transfer of domain knowledge in terms of learnt weights of the network. Transfer learning models are particularly useful in scenarios where the target application contains a limited amount of training data. As a popular approach employed in various fields of data science, transfer learning has also been found to be effective in automated plant disease detection [30].

Our employed transfer learning model is based on the MobileNetV2 architecture [13]. The model is pre-trained on the ImageNet (ILSVRC-2012-CLS) dataset. The MobileNetV2 architecture is chosen due to its suitability for usage in mobile devices. The architecture minimizes the number of mathematical operations required, thus lessening the requirement of computational power. It uses Depthwise Separable Convolutions which makes it more efficient in comparison to other neural network architectures [22]. Thus MobileNetV2 is used to create a base model, and then a convolutional layer, dropout layer, global average pooling layer, and a dense layer are added on top of this model.

In the first layer, the pre-trained MobileNetV2 architecture model is invoked. A 2D convolution layer is then added on top of it. In the convolutional layer ReLU is used as a non-linear activation function. A dropout layer is then added in the model with a dropout rate of 0.2. On top of that was a 2D global average pooling layer which is used in lieu of a fully-connected layer. The final layer is a dense layer consisting of 4 neurons and a softmax function used for the activation. The dense layer produced the final output. We use the implementation of Keras library.

Pseudo-code of the transfer learning-based model is described below in Algorithm 3. The data preparation phase includes resizing the input image in 224×224 pixels and applying data augmentation to the input data.

Algorithm 3: Transfer learning (MobileNetV2) based model

```
begin
    load the MobileNetV2 as a base model and execute the top layer;
    apply convolutional layer to the output of the top layer;
    drop 20% of neurons by applying a dropout layer;
    pass the output of the last layer into a GlobalAveragePooling2D layer;
    pass the output of the global average pooling layer using SoftMax activation function;
end
```

The intermediate layer of MobileNetV2 is used for feature extraction. After that a classifier is added on top of it. The model consists of a 3×3 convolutional layer,

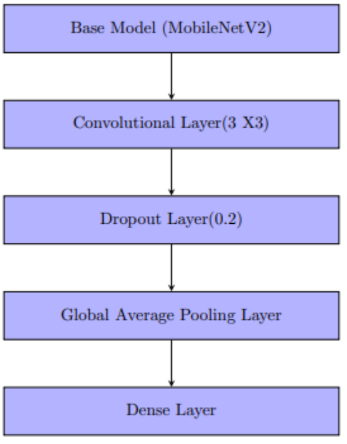


Fig. 5. Architecture of the employed transfer learning model (Model 3).

a dropout layer, a global average pooling layer and a dense layer consisting of 4 neurons. Adam optimizer and categorical-cross-entropy loss are applied while compiling this model.

3.3. Android application for leaf disease detection

After training and testing the transfer learning model, an Android application named "Go Greener" is implemented and deployed which allows users to easily use the system in their day-to-day gardening activities. The trained model is loaded into the mobile application using the Tensorflow Lite model. The application is developed using JAVA 15.0.1. The basic user interface of the application is shown in Figure 6.

The classification of the leaf's image is shown along with a confidence score. To detect the yellow leaves in *Catharanthus roseus* (bright eyes) plants and scorched leaves in *Fragaria* × *ananassa* (strawberry) plants from the "Go Greener" application, the user will simply have to open the app and place the phone on the plant leaf. The app shall then redirect the user to an interface where details on the status of the health of the leaf will be shown. Figure 7 pictorially presents the scenario.

4. Experimental results

All models are implemented in TensorFlow platform [1] which is an open source framework for developing machine learning projects in Python language. The number of training epochs varies from 1 to 10. A batch size of 64 is used. The training process is

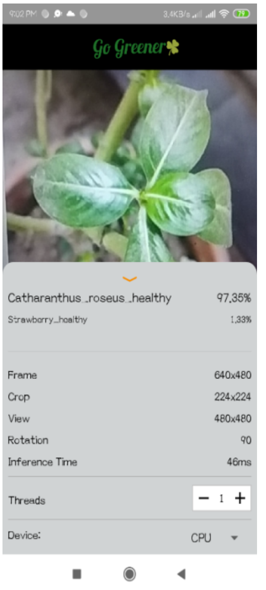


Fig. 6. Smartphone application layout.

Fig. 7. Use case diagram of the application.

accomplished on the Radeon Vega Mobile Gfx with CPU AMD Ryzen 5 3550H. All the three models are trained in the Windows 10 operating system, the CPU utilization is varying from 88% to 96% and the CPU clock speed is 2.19 GHz.

4.1. Model 1: CNN

Our CNN model gains a training accuracy of 96.47% and a validation accuracy of 95.77% after training for 10 epochs. The training phase takes 2 seconds per step. 10 training epochs are applied in the model. The influence of the number of epochs is displayed in Figure 8.

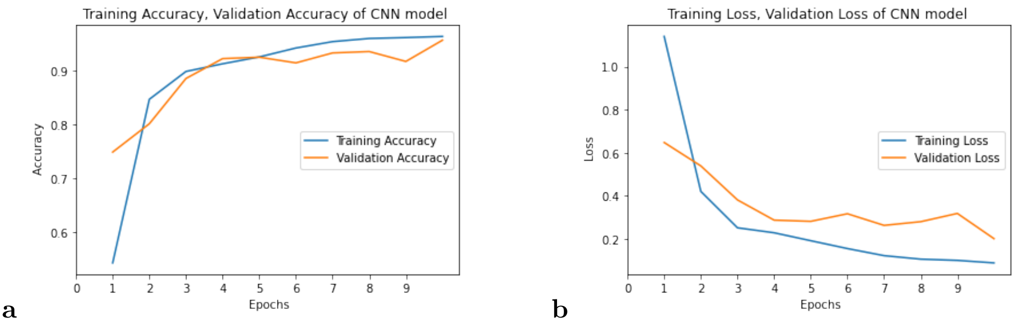


Fig. 8. CNN model – evolution of quality metrics during training: (**a**) accuracy; (**b**) loss.

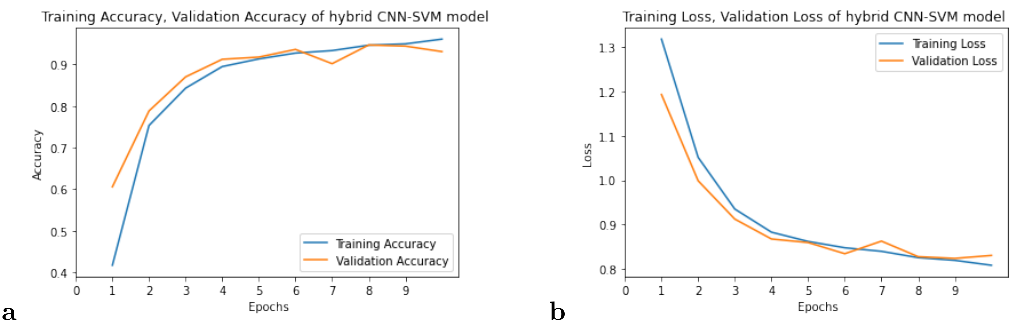


Fig. 9. Hybrid CNN-SVM model – evolution of quality metrics during training: (**a**) accuracy; (**b**) loss.

4.2. Model 2: Hybrid CNN-SVM

The CNN-SVM model gives us 96.12% training accuracy and 93.12% validation accuracy for 10 epochs. The validation loss also significantly changes in this model, as shown in Figure 9 where the Squared Hinge loss function mostly reduces the large errors and it gives a computationally effective result.

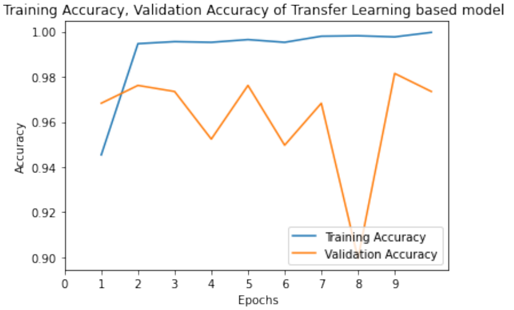
4.3. Model 3: Transfer Learning (MobileNet)

The employed transfer learning approach based on the MobileNetV2 architecture, outperforms the previous models. A training accuracy of 99.97% and validation accuracy of 97.35% are obtained using this model, making it the most effective among all three models. Table 3 describes the configuration of this model.

Figure 10 describes the changes in training and validation accuracy and loss with respect to the number of epochs for the transfer learning approach.

Tab. 3. Configuration of employed transfer learning-based model.

Parameter	Value
Convolutional Layer	1 (filter size: 3×3)
Global Average Pooling Layer	1 (filter size: 2×2)
Activation Function (Dense)	Softmax
Batch Size	64
Loss Function	`categorical_crossentropy`
Dropout	0.2
Number of Epochs	10



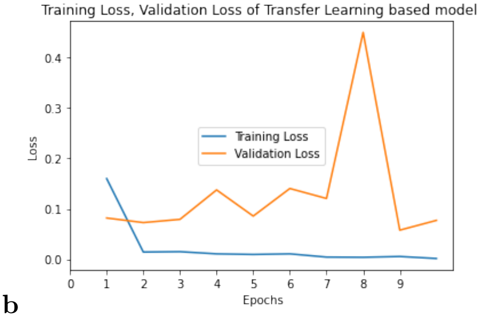


Fig. 10. Transfer learning model – evolution of quality metrics during training: (**a**) accuracy; (**b**) loss.

4.4. Comparison among all three models

The training time taken by the vanilla CNN, hybrid CNN-SVM and the MobileNetV2-based transfer learning approach are recorded to be 15, 17 and 22 minutes, respectively. The resulting training and test accuracy metrics acquired using each model are compared in Table 4 and visualized in Figure 11.

All three investigated models perform impressively, achieving test accuracy of over 90%. However, the pre-trained transfer learning model outperforms the other two as can be seen in Figure 11a. Figure 11b shows how the training and test accuracy of each model varies with the number of training epochs.

Tab. 4. Accuracy comparison between the three models.

Algorithm	CNN	SVM	Transfer Learning
Training accuracy	96.47%	96.12%	99.97%
Test accuracy	95.77%	93.12%	97.35%

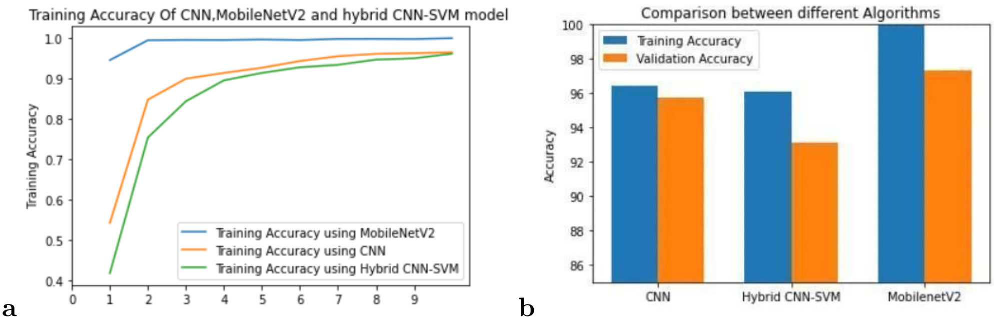


Fig. 11. Comparisons across three models, CNN, hybrid CNN-SVM, and transfer learning: (**a**) evolution of training accuracy versus training epochs; (**b**) final training and validation accuracies.

Based on precision, recall, F1-score, and accuracy, the performances of each model have been measured. Precision refers to the number of true positives divided by the total number of positive predictions. Recall is the number of true positives which are detected. F1-score is the balance between the precision and the recall which is computed by the following formula:

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}\,.$$

These metrics are shown in Table 5.

If N is the number of classes to predict, the confusion matrix is an $N \times N$ matrix which is used to evaluate the performance of the classification of a model. As shown in Figure 12, using this matrix we can compare the actual target values with the predicted ones. In the case of the CNN model (Figure 12a), it is found that the rate of true positive value is high. It can be concluded that the rate of wrong predictions of the CNN model is quite acceptably low. In Figure 12b it is found that the SVM-CNN model results in good performance though it has a lower accuracy than the CNN model. The confusion matrix for MobileNetV2 is shown in Figure 12c where it is found that the transfer learning model gives us wrong predictions 10 times. All of these three models work satisfactorily as the true positive value is high for all the models.

Tab. 5. Precision, recall and F1-score measures of each model.

Algorithm	CNN	SVM	Transfer Learning
Precision	96.00%	95.00%	97.00%
Recall	96.00%	93.00%	97.00%
F1-score	96.00%	93.00%	97.00%

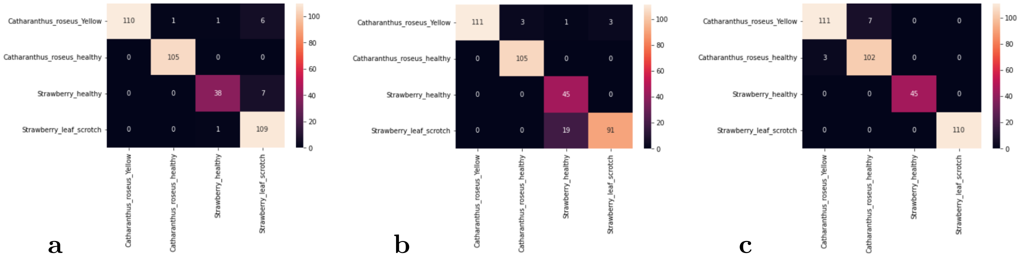


Fig. 12. Confusion matrices for (**a**) CNN model, (**b**) SVM model, (**c**) transfer learning model.

5. Conclusion

Computer vision is increasingly proving itself to be very effective to enhance the quality and production of plants. In this paper, a MobileNetV2-based transfer learning model is employed along with two other CNN-based models to classify the green and yellow leaves in *Catharanthus roseus* (bright eyes) leaves and also to classify healthy and scorched leaves of *Fragaria* × *ananassa* (strawberry) plants. Satisfactory accuracy has been achieved in the experiments. Once trained, the models can be loaded and used in smartphone applications, which can be used by end-users to classify images of leaves in real time using their mobile phone cameras. The model is expected to be helpful for cultivators in detecting diseases within bright eyes and strawberry plants. Future research in this field may generate several interesting directions. The deployed transfer learning approach can be tweaked further to improve performance and can be trained to identify diseases in other plants with even greater economic significance. Other transfer learning approaches can be considered including ResNet, GoogLeNet and EfficientNet [11]. Since the developed dataset is somewhat imbalanced in terms of class distribution, re-sampling techniques [6] can be applied to improve the accuracy.

References

[1] M. Abadi, A. Agarwal, P. Barham, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. `https://www.tensorflow.org/`. Software available from tensorflow.org.

[2] S. Ahlawat and A. Choudhary. Hybrid CNN-SVM classifier for handwritten digit recognition. *Procedia Computer Science*, 167:2554–2560, 2020. doi:10.1016/j.procs.2020.03.309.

[3] S. I. Ahmed, M. Ibrahim, Md. Nadim, et al. MangoLeafBD: A comprehensive image dataset to classify diseased and healthy mango leaves. *Data in Brief*, 47:108941, 2023. doi:10.1016/j.dib.2023.108941.

[4] J. Amara, B. Bouaziz, and A. Algergawy. A deep learning-based approach for banana leaf diseases classification. In B. Mitschang et al., editors, *Datenbanksysteme für Business, Technologie und Web (BTW 2017) – Workshopband*, pages 79–88, Bonn, 2017. Gesellschaft für Informatik e.V. `https://dl.gi.de/handle/20.500.12116/944`.

[5] L. E. C. Antunes and N. A. Peres. Strawberry production in Brazil and South America. *International Journal of Fruit Science*, 13(1-2):156–161, 2013. doi:10.1080/15538362.2012.698147.

[6] L. E. C. Antunes and N. A. Peres. Sampling non-relevant documents of training sets for learning-to-rank algorithms. *International Journal of Machine Learning and Computing*, 10(3):406–415, 2020. doi:10.18178/ijmlc.2020.10.3.950.

[7] J. Chen, D. Zhang, Y. A. Nanehkaran, and D. Li. Detection of rice plant diseases based on deep transfer learning. *Journal of the Science of Food and Agriculture*, 100(7):3246–3256, 2020. doi:10.1002/jsfa.10365.

[8] S. S. Chouhan, U. P. Singh, and S. Jain. Applications of computer vision in plant pathology: A survey. *Archives of Computational Methods in Engineering*, 27(2):611–632, 2020. doi:10.1007/s11831-019-09324-0.

[9] D. Das, M. Singh, S. S. Mohanty, and S. Chakravarty. Leaf disease detection using Support Vector Machine. In *Proc. 2020 Int. Conf. Communication and Signal Processing (ICCSP)*, pages 1036–1040, Chennai, India, 28-30 Jul 2020. IEEE. doi:10.1109/ICCSP48568.2020.9182128.

[10] Keras Special Interest Group. Keras. simple. flexible. powerful. `https://keras.io`.

[11] Md. M. Hasana, M. Ibrahim, and Md. S. Ali. Speeding up EfficientNet: Selecting update blocks of convolutional neural networks using genetic algorithm in transfer learning. *arXiv*, 2023. arXiv:2303.00261v1. doi:10.48550/arXiv.2303.00261.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 2016)*, pages 770–778, Las Vegas, NV, USA, 27-30 Jun 2016. doi:10.1109/CVPR.2016.90.

[13] A. G. Howard, M. Zhu, B. Chen, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv*, 2017. arXiv:1704.04861v1. doi:10.48550/arXiv.1704.04861.

[14] A. J. Keutgen and E. Pawelzik. Impacts of NaCl stress on plant growth and mineral nutrient assimilation in two cultivars of strawberry. *Environmental and Experimental Botany*, 65(2-3):170–176, 2009. doi:10.1016/j.envexpbot.2008.08.002.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25 – Proc. 25th Conf. NIPS 2012*, volume 25, pages 1097–1105, Lake Tahoe, NV, USA, 3-6 Dec 2012. Curran Associates, Inc. `https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`.

[16] F. Kurtulmuş. Identification of sunflower seeds with deep convolutional neural networks. *Journal of Food Measurement and Characterization*, 15:1024–1033, 2021. doi:10.1007/s11694-020-00707-7.

[17] Large Scale Visual Recognition Challenge 2012 (ILSVRC2012). Stanford Vision Lab, 2012. `http://www.image-net.org/challenges/LSVRC/2012/results`.

[18] U. Mokhtar, N. El Bendary, A. E. Hassenian, et al. SVM-based detection of tomato leaves diseases. In D. Filev, J. Jabłkowski, J. Kacprzyk, et al., editors, *Intelligent Systems'2014. Advances in Intelligent Systems and Computing. Proc. 7th IEEE Int. Conf. Intelligent Systems (IS'2014)*, volume 323 of *Advances in Intelligent Systems and Computing*, pages 641–652. Springer, Warsaw, Poland, 24-26 Sep 2015. doi:10.1007/978-3-319-11310-4_55.

[19] T. Muhammad, A. B. Aftab, Md. Ahsan, et al. Transformer-based deep learning model for stock price prediction: A case study on Bangladesh stock market. *International Journal of Computational Intelligence and Applications*, 22(1):2350013, 2023. doi:10.1142/S146902682350013X.

[20] S. Nammi, M. K. Boini, S. D. Lodagala, and R. B. S. Behara. The juice of fresh leaves of Catharanthus roseus Linn. reduces blood glucose in normal and alloxan diabetic rabbits. *BMC Complementary and Alternative Medicine*, 3(1):4, 2003. doi:10.1186/1472-6882-3-4.

[21] X.-X. Niu and C. Y. Suen. A novel hybrid CNN-SVM classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4):1318–1325, 2012. doi:10.1016/j.patcog.2011.09.021.

[22] M. Sandler, A. Howard, M. Zhu, et al. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proc. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognitioneedings (CVPR 2018)*, pages 4510–4520, Salt Lake City, UT, USA, 18-23 Jun 2018. doi:10.1109/CVPR.2018.00474.

[23] F. T. Shohan, A. U. Akash, M. Ibrahim, and M. S. Alam. Crime prediction using machine learning with a novel crime dataset. *arXiv*, 2022. doi:10.48550/arXiv.2211.01551.

[24] S. Sladojevic, M. Arsenovic, A. Anderla, et al. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, 2016:3289801, 2016. doi:10.1155/2016/3289801.

[25] S. Sun, Z. Wang, Z. Zhang, et al. The roles of entomopathogenic fungal infection of viruliferous whiteflies in controlling tomato yellow leaf curl virus. *Biological Control*, 156:104552, 2021. doi:10.1016/j.biocontrol.2021.104552.

[26] C. Szegedy, W. Liu, Y. Jia, et al. Going deeper with convolutions. *arXiv*, 2014. arXiv:1409.4842v1. doi:10.48550/arXiv.1409.4842.

[27] D. Tiwari, M. Ashish, N. Gangwar, et al. Potato leaf diseases detection using deep learning. In *Proc. 2020 4th Int. Conf. Intelligent Computing and Control Systems (ICICCS 2020)*, pages 461–466, Madurai, India, 13-15 May 2020. IEEE. doi:10.1109/ICICCS48265.2020.9121067.

[28] M. Vurro, B. Bonciani, and G. Vannacci. Emerging infectious diseases of crop plants in developing countries: impact on agriculture and socio-economic consequences. *Food Security*, 2(2):113–132, 2010. doi:10.1007/s12571-010-0062-7.

[29] P. E. Waggoner. The aerial dispersal of the pathogens of plant disease. *Philosophical Transactions of the Royal Society B, Biological Sciences*, 302(1111):451–462, 1983. doi:10.1098/rstb.1983.0067.

[30] Y. Zhong and M. Zhao. Research on deep learning in apple leaf disease recognition. *Computers and Electronics in Agriculture*, 168:105146, 2020. doi:10.1016/j.compag.2019.105146.