Grenze International Journal of Engineering and Technology, June Issue

An Investigative Study of Plant Disease Prediction using Machine Learning and Deep Learning Algorithms

Mrs.Disha Sushant Wankhede[1], Vaishali Mishra[2],Atharva Honkhambe[3], Siddhi Deshmukh[4], Akash Singh[5], Shivam Srivastava[6], and Jinshel George[7]
[1,2]Assistant Professor, Vishwakarma Institution of Information Technology/Department of Computer Science, Pune, India
Email: disha.wankhede@viit.ac.in , vaishali.mishra@viit.ac.in
[3-5] Students, Vishwakarma Institution of Information Technology/Department of Computer Science, Pune, India
Email: atharva.21810009@viit.ac.in, siddhi.21810005@viit.ac.in, akash.21810779@viit.ac.in,
[6-7] Students, Vishwakarma Institution of Information Technology/Department of Electronics and Telecommunication, Pune, India
Email: shivam.21810428@viit.ac.in, jinshel.21810425@viit.ac.in

***Abstract*—Plant diseases have an impact on the growth of their particular species, hence early detection is critical. Many Machine Learning (ML) models have been used to identify and classify plant diseases, but with developments in a subset of ML, Deep Learning (DL), this field of research looks to have considerable potential in terms of increasing accuracy. Many developed/modified DL architectures, as well as various visualisation approaches, are used to identify and categorise the signs of plant diseases. Furthermore, numerous performance indicators are utilised to assess these architectures/techniques. This article gives a thorough overview of the DL models that are used to visualise various plant diseases.**

***Index Terms*— Plant diseases detection, Machine learning, Image processing, Agriculture, convolutional neural network.**

I. INTRODUCTION

Agricultural production is vitally important to the economy. This is why disease detection in plants is important in the agricultural area, as illness in plants is fairly common. However, in underdeveloped nations' rural regions, eye inspection is still the predominant method of illness detection. It also needs skilled monitoring on a continual basis. Farmers in rural places may have to drive a long distance to see an expert, which is both time-consuming and costly.

The diagnosis of illness on the plant is a critical step in avoiding a significant loss of harvest and agricultural product amount. The indications can be found on plant components such as stems, leaves, lesions, and fruits. The leaf demonstrates the symptoms by changing, revealing the spots on it. This illness identification is accomplished by manual statement and pathogen detection, which might take extra time and be costly. Pre-processing, Training, and Identification are all required phases in the process. Computers have been utilized for systematization and automation in many agricultural/horticultural applications, including the construction of expert systems (decision support systems) employing computer vision technologies. Colour and texture topographies have been utilized to deal with plant disease trial photographs. Algorithms are used to detect the presence of diseases in leaf photos, the image segmentation approach is applied. Colour and texture feature

Grenze ID: 01.GIJET.8.2.521

extraction has been created, which may be utilized to train support vector machine (SVM) and artificial neural network (ANN) classifiers.Machine learning may be used to classify plant diseases using a variety of feature sets. Traditional and deep-learning (DL)-based features are the most prominent among them. Before efficiently extracting features, pre-processing such as picture enhancement, colour modification, and segmentation [5] is required. Following feature extraction, many classifiers may be utilized. K-nearest neighbour (KNN) [6, 7], support vector machine (SVM) [8, 9], decision tree, random forest (RF) [10], naïve Bayes (NB), logistic regression (LR), rule generation, artificial neural networks (ANNs)[11], and Deep CNN are other prominent classifiers.

II. METHODOLOGY

A. Convolutional Neural Network (CNN)

The main purpose of this study is to create a Convolution Neural Network that can correctly anticipate tomato leaf diseases. Target spot, mosaic virus, yellow leaf curl virus, bacterial spot, early blight, late blight, and Septoria leaf spot are the top seven diseases that harm tomato leaves.

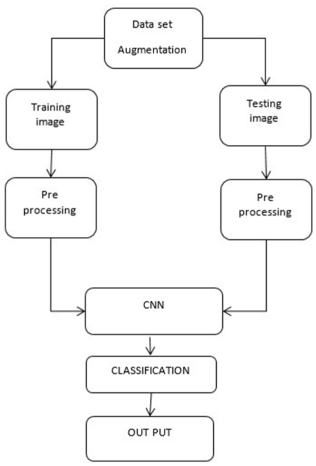
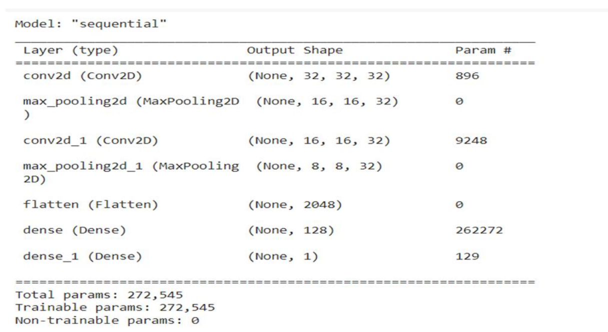


Figure 1. Flowchart of CNN



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 32, 32, 32)        896

 max_pooling2d (MaxPooling2D  (None, 16, 16, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 16, 16, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 8, 8, 32)         0
 2D)

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 128)               262272

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 272,545
Trainable params: 272,545
Non-trainable params: 0
_________________________________________________________________
```

Figure 2. Summary of CNN

There is a total of 272545 trainable parameters. To begin, conv2d contains 896 parameters. Following it, max_pooling2d has 0 parameters, followed by conv2d 1, which has 9246 parameters. There are no trainable parameters in max_pooling2d_1 and flatten layer. Dense and dense_1 layer each have 262272 and 129 trainable parameters.

A. Support Vector Machine (SVM)

Support Vector Machines (SVMs) are supervised learning models that are commonly employed to solve classification challenges. The method draws a line or a hyperplane to divide the data into classes.
SVM creates a decision boundary that splits n-dimensional space into classes, making it easy to insert new data points in the appropriate category in the future. The best suitable boundary is a hyperplane. SVM chooses the hyperplane's extreme points/vectors. Support vectors are extreme instances.
The classifier is trained using photographs of plant leaves to provide a decision boundary for classifying the two groups by detecting extreme examples or support vectors. It will categorize the picture based on these support vectors.

C. K Nearest Neighbour (KNN)

A supervised machine learning technique is K closest neighbour. The K-NN approach assumes that the new case/data and old cases are comparable and places the new case in the most similar category to the existing categories. It stores all available data and categorises new data points according to their similarity. As new data enters, it may easily be sorted into the appropriate category. The KNN method simply saves the dataset throughout the training phase, and when new data is received, it is classified into a category that is quite similar to the new data. Assume there are two categories, A and B, and we have a new data point x1 and want to

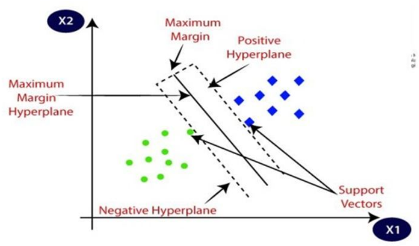


Figure 3. Two different categories that are classified

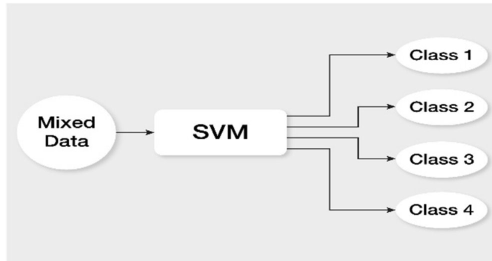


Figure 4. Flowchart of SVM using decision boundary or hyperplane

determine which of these two groups it belongs to. To solve this problem, you'll need to use the K-NN approach. Using K-NN, we may quickly identify a dataset's category or class.

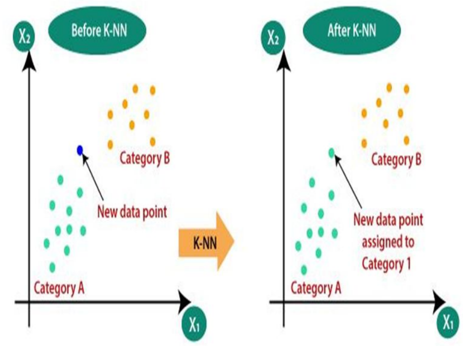


Figure 5. Need of KNN algorithm

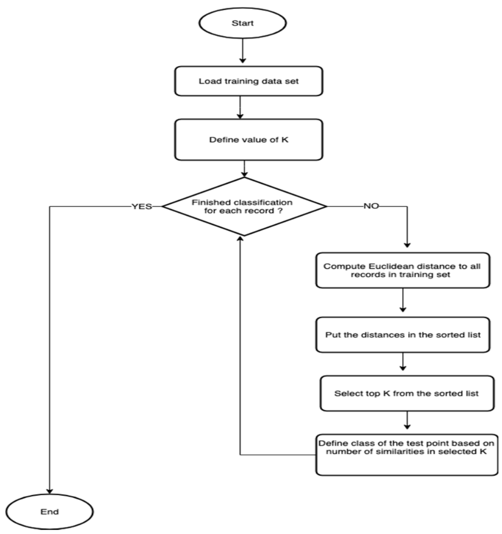


Figure 6. Flowchart of KNN

The KNN model was trained on two separate kinds of potato blight: early and late blight. It discovers the new data's comparable characteristics that lead to the two categories and categorizes the data into one of the two groups based on the most similar features.

This approach computes the Euclidean distance between k neighbours. It counts the number of data points in each category among these k neighbours. The new data points are assigned to the category with the greatest number of neighbours.

D. Random Forest Classifier

Random forest is a popular supervised learning technique for classification and regression tasks. The decision tree is the fundamental unit of random forest classifiers. It creates decision trees based on different samples and uses their majority vote for categorization.

The decision tree is a hierarchical structure constructed from the characteristics (or independent variables) of a data collection. Each internal node in a decision tree represents a 'test' on an attribute (e.g., whether a coin flip comes up heads or tails), each branch reflects the test's conclusion, and each leaf node represents a class label (decision taken after computing all attributes). A leaf is a node that has no offspring.

By examining the feature importance, you may determine which characteristics should be dropped since they do not contribute enough (or nothing at all) to the prediction process. This is significant because, in general, the more characteristics you have, the more likely your model will suffer from overfitting, and vice versa.

Random forest has roughly identical hyperparameters as decision trees and bagging classifiers. Fortunately, there is no need to combine a decision tree with a bagging classifier because the random forest classifier-class may be used instead.Random Forest, in a nutshell, creates many decision trees and blends them to get a more accurate and reliable forecast.

1.Image Acquisition (Dataset): In this classification, an image dataset consisting of photographs of various plants' leaves is utilized, which is accessible as plantvillage dataset on GitHub. It includes plants such as apple, blueberry, cherry, corn, grapes, tomato, potato, and so on, as well as disease labels such as apple black rot, corn common rust, black measles, tomato mosaic virus, and so on. These tagged photos will be used for illness categorization and testing.
2. Data Division: We divided the dataset into two files, Train and Validation. The machine learning model will be trained using the train folder. It is divided into nine subfolders, each of which contains the labels for various illness groups. The validation folder contains validation photos that will be used to evaluate the trained ML model's ability to forecast plant disease and subsequently assess the model's accuracy.
3.Image Preprocessing: Using the label encoder capability of the Sklearn preprocessing toolkit, we converted the labels from the text (folder names) to integers. The pixel values between 0 and 1 were then adjusted to bring the photos to the same scale.
4.Feature Extraction: After organizing the dataset appropriately, we will apply a number of filters to each training image and collect the answer in a pandas data frame in this step. We employed two filters or feature extractors for this step: pixel values and Gabor filters. The best aspect of every image is its pixel value. Because a bright image tells us something, but a dark image tells us something else.
A series of Gabor filters with varying frequencies and orientations may be effective for extracting valuable characteristics from a picture. Only theta and sigma are modified in Gabor filters for this categorization. By varying lambda, gamma, and ksize, we can generate an endless number of filters. Following that, we rearranged the data to create a vector for categorization.

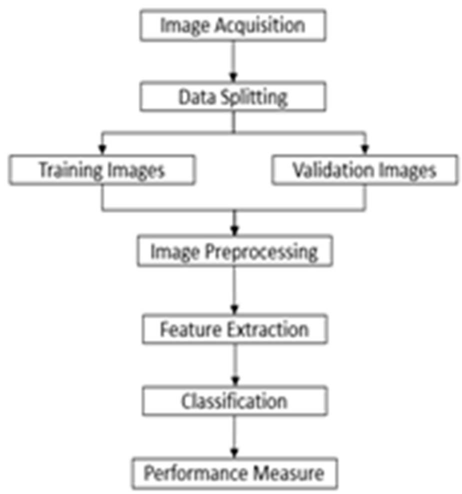


Figure 7. Block diagram of the process of classification

III. RESULTS AND DISCUSSIONS

A. Convolutional Neural Network (CNN)

To train the proposed models, an available dataset of binary tomato plant disease species was employed. To test the models' performance in CNN, other factors such as batch size and number of epochs were introduced. We employed a GPU runtime type with a 50 percent memory allocation. The implemented models had a disease-classification accuracy rate of 99.8 percent at the 15th epoch, which was greater than that of usual handcrafted-feature-based tech.
Conv2D and Dense are used to build a convolutional neural network in five phases.
We utilized the activation function relu to develop CNN, and using relu helps to limit the computation required to operate the neural network from growing exponentially. As the size of the CNN rises, the computational cost of adding more ReLUs grows linearly.
We have 99.87 percent accuracy, 99.89 percent val_accuracy, 0.6 percent loss, and 0.64 percent val_loss after the 15th epoch.Around 4689 images of two kinds of tomato leaf diseases were used in the analysis. The model was created with Keras, a Python-based neural network API.Each block consists of three layers: a convolutional layer, an activation layer, and a maximum pooling layer. This design features four blocks, followed by

```python
# Part 2 - Building the CNN
import tensorflow as tf
# Initialising the CNN
cnn = tf.keras.models.Sequential()

# Step 1 - Convolution
cnn.add(tf.keras.layers.Conv2D(filters=32,padding="same",kernel_size=3, activation='relu', strides=2, input_shape=[64, 64, 3]))

# Step 2 - Pooling
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))

# Adding a second convolutional layer
cnn.add(tf.keras.layers.Conv2D(filters=32,padding='same',kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))

# Step 3 - Flattening
cnn.add(tf.keras.layers.Flatten())

# Step 4 - Full Connection
cnn.add(tf.keras.layers.Dense(units=128, activation='relu'))

# Step 5 - Output Layer
#cnn.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
## For Binary Classification
cnn.add(Dense(1, kernel_regularizer=tf.keras.regularizers.l2(0.01),activation
         ='linear'))
```

Figure 8(a). Building the convolutional neural network with the five steps with Conv2D and Dense

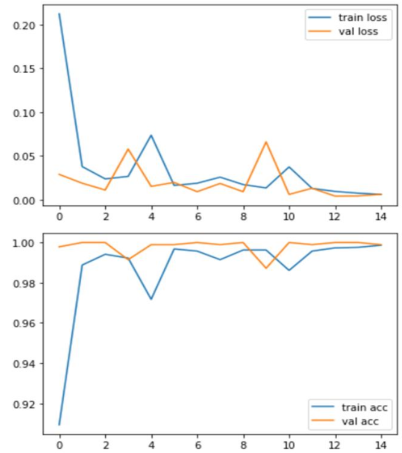


Figure 8(b). Training and testing accuracy with CNN

completely connected layers with SoftMax activation. Convolutional and pooling layers are used to extract information, whereas fully connected layers are used to categorise and forecast. The network's activation layer introduces nonlinearity.

```python
[15] import numpy as np
     from tensorflow.keras.preprocessing import image
     test_image = image.load_img('/content/drive/MyDrive/archive/New Plant Diseases Dataset/valid/Tomato___Tomato_Yellow_Leaf_Curl_Virus/01933767
     test_image = image.img_to_array(test_image)
     test_image=test_image/255
     test_image = np.expand_dims(test_image, axis = 0)
     result = cnn.predict(test_image)

[16] result

     array([[-0.22492985]], dtype=float32)

     if result[0]<0:
       print("Image classified is Tomato_Yellow_Leaf_Curl_Virus")
     else:
       print("Image Classified is Tomato_mosaic_virus")

     Image classified is Tomato_Yellow_Leaf_Curl_Virus
```

Figure 9. The image classification report

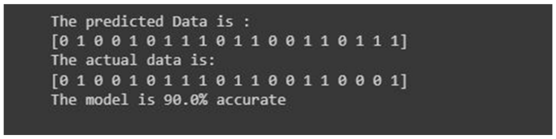
This is the only example of the outcome. If the floating value is more than zero, the picture is regarded severely contaminated. However, if the floating value is smaller, the picture is regarded to be less infected by that particular virus. Agriculture is one of the most important industry in India, employing the majority of the people. As a result, identifying diseases in these plants is critical for economic growth. Tomatoes are one of the most significant crops that are mass-produced. As a result, using binary classification, this technique tries to detect and forecast two distinct illnesses in the tomato crop. Although we only tested the built model on the customized plant village dataset, we believe it may be used to handle other classification problems. Future studies will explore how well the constructed model can detect and classify other datasets, as well as plant diseases.

B. Support Vector Machine (SVM)

The SVM model was trained using 50 photos of potato early blight and 50 images of late blight. Twenty percent of the photos were chosen to put the model to the test. The model's accuracy is determined to be 90%.

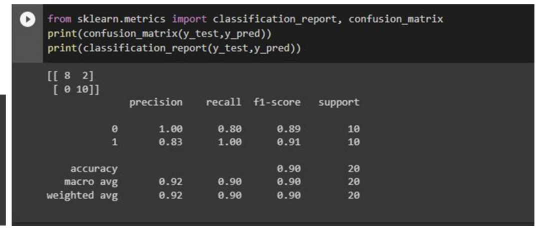
C. K Nearest Neighbour (KNN)

The KNN model was also trained using 50 late blight photos and 50 early blight images. The model's accuracy was about 60% when 20 percent of the photos were used as testing data.



```
The predicted Data is :
[0 1 0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 1 1 1]
The actual data is:
[0 1 0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 0 0 1]
The model is 90.0% accurate
```

Figure 10(a). The SVM model accuracy report



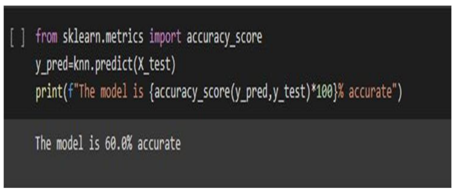
```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

[[ 8  2]
 [ 0 10]]
              precision    recall  f1-score   support

           0       1.00      0.80      0.89        10
           1       0.83      1.00      0.91        10

    accuracy                           0.90        20
   macro avg       0.92      0.90      0.90        20
weighted avg       0.92      0.90      0.90        20
```

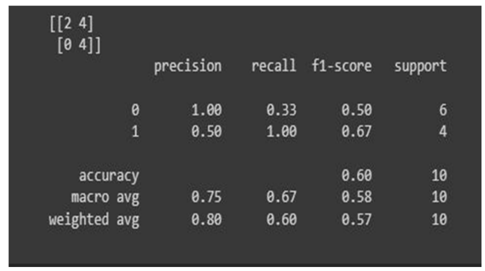
Figure 10(b). The SVM classification report



```
from sklearn.metrics import accuracy_score
y_pred=knn.predict(X_test)
print(f"The model is {accuracy_score(y_pred,y_test)*100}% accurate")

The model is 60.0% accurate
```

Figure 11(a). The KNN model accuracy report



```
[[2 4]
 [0 4]]
              precision    recall  f1-score   support

           0       1.00      0.33      0.50         6
           1       0.50      1.00      0.67         4

    accuracy                           0.60        10
   macro avg       0.75      0.67      0.58        10
weighted avg       0.80      0.60      0.57        10
```

Figure 11(b). The KNN classifier classification report

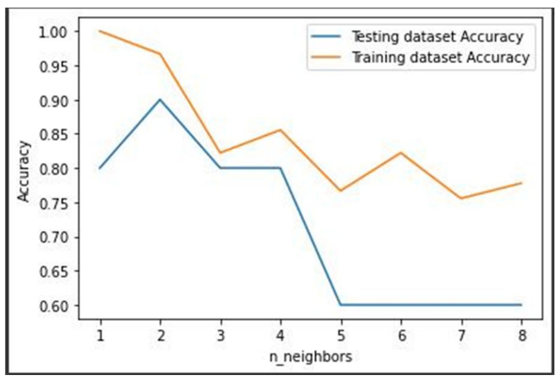


Figure 11(c). Training and testing accuracy with n_neighbours

D. Random Forest Classifier

In this experiment, the picture size was reduced from 128 to 128 by the original size. The reason for this is because all of the photographs have a unique size since the images in the data set were all different sizes. Following that, certain codes are added to the model and details are trained on it, and the model forecasts image illness appropriately. The categorization is 65 percent accurate.

The model is described in further depth in the categorization report linked above. It assesses the accuracy of the model's predictions. Precision is a classifier's ability to avoid labelling a negative occurrence as positive. A classifier's recall is its capacity to discover all positive occurrences.

In F1, 1.0 is the highest score and 0.0 is the lowest score based on a weighted harmonic average of accuracy and recall. Compared to the random forest and SVM models, the KNN model scores in the middle at 0.60 on average, but lower than the random forest.

IV. CONCLUSIONS

Agriculture is one of the most significant industry in India, employing the majority of the people. As a result, identifying diseases in these plants is critical for economic growth. In this study, we assessed some of the major problems and shortcomings of previous studies that examined the automated detection of agricultural illnesses using various algorithms. The results demonstrate that the four models' performance varies greatly. As a result, depending on the accuracy of the comparing findings, the CNN and SVM significantly outperform the RFC and KNN. Our suggested network, which employs a neural convolutional network (CNN), SVM, RFC, and KNN, has the potential to be extremely effective for identifying and forecasting numerous plant leaf diseases. In the

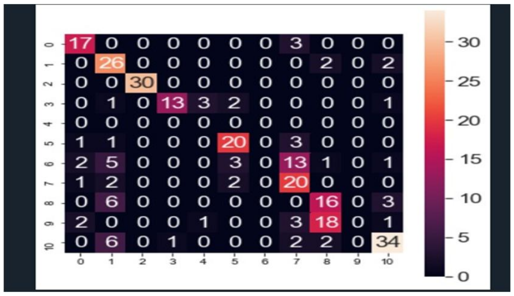


Figure 12. Confusion matrix

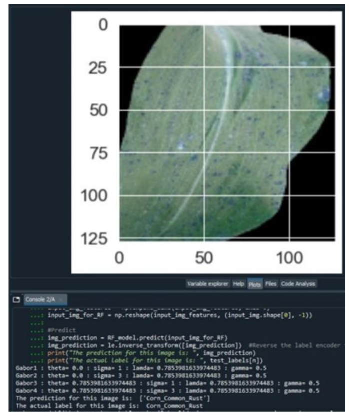


Figure 13(a). Image prediction report for Corn Leaf

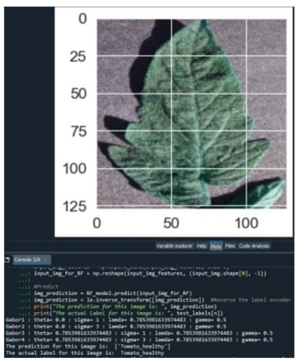


Figure 13(b). Image prediction report for Tomato leaf

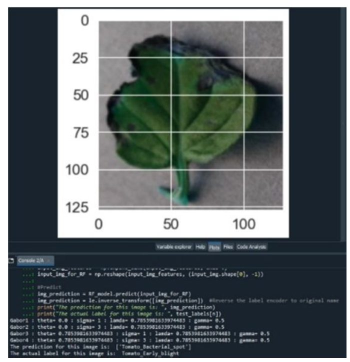


Figure 13(c). Image prediction report for Tomato leaf

E. Compariosn

ALGORITHM	PREDICTION ACCURACY (%)	F1-SCORE ACCURACY	SUPPORT
CNN	99.87%	-	-
KNN	60%	0.60	10
SVM	90%	0.90	20
RFC	65%	-	-

future, we want to assess the constructed model for the detection and classification of other datasets, as well as plant leaf disease.

ACKNOWLEDGMENT

The knowledge behind this study was obtained from a variety of research publications centered on detection of plant diseases. and numerous strategies connected to it. This work would not have been possible without the contributions of anonymous reviewers. We would also want to express our appreciation to our institute, Vishwakarma Institute of Information Technology, Pune, for providing us with the chance to do research on this paper.

REFERENCES

[1] Camargo, A., and J. S. Smith. "An image-processing based algorithm to automatically identify plant disease visual symptoms." Biosystems engineering 102.1 (2009): 9-21.

[2] Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanehkaran, Y.A. Using deep transfer learning for image-based plant disease identification. Comput. Electron. Agric. 2020, 173, 105393.

[3] Bai, X.; Cao, Z.; Zhao, L.; Zhang, J.; Lv, C.; Li, C.; Xie, J. Rice heading stage automatic observation by multi-classifier cascade based rice spike detection method. Agric. For. Meteorol. 2018, 259, 260–270.
[4] Ramcharan, A.; Baranowski, K.; McCloskey, P.; Ahmed, B.; Legg, J.; Hughes, D.P. Deep Learning for Image-Based Cassava Disease Detection. Front. Plant Sci. 2017, 8, 1852.
[5] Camargo, A.; Smith, J. An image-processing based algorithm to automatically identify plant disease visual symptoms. Biosyst. Eng. 2009, 102, 9–21.
[6] Singh, J.; Kaur, H. Plant disease detection based on region-based segmentation and KNN classifier. In Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018, Palladam, India, 16–17 May 2018; pp. 1667–1675.
[7] Camargo, A.; Smith, J. Image pattern classification for the identification of disease causing agents in plants. Comput. Electron. Agric. 2009, 66, 121–125.
[8] Türkoğlu, Muammer, and Davut Hanbay. "Plant disease and pest detection using deep learning-based features." Turkish Journal of Electrical Engineering & Computer Sciences 27.3 (2019): 1636-1651.
[9] Fuentes, Alvaro, et al. "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition." Sensors 17.9 (2017): 2022.
[10] Loey, Mohamed, Ahmed ElSawy, and Mohamed Afify. "Deep learning in plant diseases detection for agricultural crops: a survey." International Journal of Service Science, Management, Engineering, and Technology (IJSSMET) 11.2 (2020): 41-58.
[11] Wankhede D.S. (2021) Analysis and Prediction of Soil Nutrients pH,N,P,K for Crop Using Machine Learning Classifier: A Review. In: Raj J.S. (eds) International Conference on Mobile Computing and Sustainable Informatics. ICMCSI 2020. EAI/Springer Innovations in Communication and Computing. Springer, Cham. https://doi.org/10.1007/978-3-030-49795-8_10.