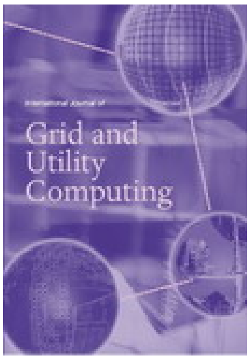


International Journal of Grid and Utility Computing

ISSN online: 1741-8488 - ISSN print: 1741-847X
https://www.inderscience.com/ijguc

Detection of crop disorder using deep learning

Vinita, Suma Dawn

DOI: 10.1504/IJGUC.2023.10060098

Article History:

Received:	03 January 2023
Last revised:	21 April 2023
Accepted:	09 July 2023
Published online:	19 February 2024

Detection of crop disorder using deep learning

Vinita* and Suma Dawn

Jaypee Institute of Information Technology,
Noida, Uttar Pradesh, India
Email: vinita89.cse@gmail.com
Email: Suma.dawn@gmail.com
*Corresponding author

Abstract: An estimated 14% of global yield is lost to plant diseases each year, causing suffering to billions of people. Plant pathology studies diseases, microbes and climatic conditions that lead to plant death. Temperature, pH, humidity and moisture can cause plant diseases. Chemical misuse, environmental imbalance and drug resistance can result from misdiagnosis. Diseases can be diagnosed by human scouting. Image analysis of plant leaves can help diagnose diseases automatically. Automated disease detection involves image selection, pre-processing, segmentation, augmented features and model prediction. Crop diseases can be detected and classified accurately by Deep Convolutional-Networks since a few years ago. This paper compares deep learning approaches for predicting healthy and diseased leaves from Mendley database. We suggest variations that improve classification accuracy. In this work for disease, Deep CNNs are implemented including ResNet-50, Mobilenet, Densenet121, EfficientnetB0 and the proposed approach. Over 99% accuracy was achieved in detecting various crop diseases.

Keywords: deep learning; crop disease detection; ResNet-50; Mobilenet; Densenet121; EfficientnetB0; image processing.

Reference to this paper should be made as follows: Vinita and Dawn, S. (2024) 'Detection of crop disorder using deep learning', *Int. J. Grid and Utility Computing*, Vol. 15, No. 1, pp.65–74.

Biographical notes: Vinita received her BTech degree in Computer Science and Engineering from Uttar Pradesh Technical University, Lucknow, India in 2010 and MTech degree in Computer Science and Engineering from Jaypee Institute of Information Technology, Noida, India in 2012. Currently, she is pursuing PhD degree from Jaypee Institute of Information Technology, Noida, India. She is a Research Scholar in the Computer Science branch. Her interest areas include image processing, computer vision and machine learning.

Suma Dawn received her BE degree in Information Technology from Sardar Vallabhbhai Patel Institute of Technology, Gujarat University, Gujarat, MTech degree in Computer Science from Kalyani University, West Bengal and PhD degree in Computer Science and Engineering from Jaypee Institute of Information Technology, Noida, India. Currently, she is an Assistant Professor (Senior Grade) at Jaypee Institute of Information Technology, Noida, India. Her research interests include HCI & computer vision, image processing & computer graphics, bioinformatics, brain-computer interface, pattern recognition and IoT and agriculture.

1 Introduction

A key aspect of agriculture is the early detection of plant leaf infections. Various techniques have been used to evaluate the leaf's quality, including thermography, fluorescence imaging, affinity biosensor based on Ribonucleic Acid (RNA) and Deoxyribonucleic Acid (DNA), chain reactions and natural gas chromatography. The techniques above were criticised for their inadequacy, consistency and extension. Many researchers have succeeded in overcoming these challenges using image processing and machine learning/deep learning techniques. It has been shown that image-processing techniques can be used to recognise and categorise plant diseases, as proven in studies (Nagaraju and Chawla, 2020; Kaur et al., 2019; Rehman et al., 2019; Saleem and Arif, 2019; Kamilaris and Prenafeta-Boldu, 2018; Dhingra et al., 2018). There has been a recent trend in that most of the research studies in the field of plant disease classification and identification have focused on applications of machine learning or deep learning algorithms. As a result, this study is the first to systematically examine and summarise the various research findings currently being conducted on the subject of detecting and classifying leaf diseases using ML and DL algorithms. Additionally, a summary of the latest developments of popular DL models used to identify plant diseases will also be included in the proposed study. It was also found that in addition to identifying some of the gaps in the existing literature to be able to measure the symptoms experienced by patients in the classification of plant leaf diseases more clearly, a study was conducted to identify some of the research gaps. This field of research has been around for a very long time and has also been very popular. It has been implemented in most major applications. Convolutional Neural Networks (CNN) were used in this paper to solve the problem of identifying plant disease by analysing the leaf of a plant. From deep within the food chain, plants are a major contributor to life on Earth. The various conditions in nature make these plants prone to various diseases. Because of these diseases, the agricultural industry suffers greatly. This can save a great deal of money and effort by detecting and curing diseases at the earliest stages possible. Using a leaf image, we have developed a system that analyses, detects and classifies any disease the plant may have suffered from based on deep learning.

The rest of this paper is organised as follows; Section 2 is devoted to presenting a short review of disease-specific methods that are known to be used by traditional practitioners to diagnose plants. This was followed by a short description of the data set in Section 3. We then proceed to Section 4, which helps us to understand the proposed analysis process. As part of Section 5, a set of experiments is presented covering all of the currently available methods for detecting plant diseases. This is in addition to introducing novel and increasingly efficient methods that significantly improve the current state. The conclusions of this paper are presented in Section 6, followed by a list of all the literature that is relevant to this study.

2 Literature survey

Dasari and Prasad (2019) used Convolutional-Neural-Networks (CNNs) with 120 images for recognising tobacco leaf diseases. According to the authors, when compared to existing models, the proposed model showed the best accuracy of 85.10% or 80% accuracy. To classify nine diseases of tomato plants, the authors have devised three deep learning meta-architectures: Faster Region-based Convolutional Neural Network (Faster R-CNN).

To decrease false positives and to improve accuracy during training they applied different techniques to the data, such as feature extraction and data augmentation, and 83.6% of the disease targets can be classified correctly in the proposed model. Brahimi et al. (2017) used machine learning to train two architectures to classify nine diseases of tomato leaves (Google Net and AlexNet). The pre-processing was initially applied to all images to resize them and to get rid of the background information on the images. The model was later applied to a classification layer to determine the classification of the fruit diseases and the diseases of leaf tissues of tomato plants with an accuracy of 99.18%. Durmus et al. (2017) recommended that AlexNet and SqueezeNet could be used as a model for classifying ten diseases of tomato plant leaves. Both of the deep learning networks in this paper have been trained and validated by the authors in this particular case.

Despite this, AlexNet achieves a greater degree of accuracy than these two models, hitting 96.65%. It has been proposed by Zhang et al. (2018) that improved Cifar10 and GoogleLeNet models are now available for the classification of the nine diseases that affect maize plant leaves. The researchers changed dropout operations, adjusted aggregate accumulation pooling combinations and adjusted parameters. GoogleNet performed 99.9% correctly, according to their research. Singh et al. (2017) used ten of 500 natural images of rice plant diseases to classify by a Deep CNN (DCNN) model. An accuracy of 95.48% was achieved by cross-validating a 10-fold strategy. According to Jain et al. (2017), pomegranate leaves could be classified as two diseases based on a CNN model. Researchers found that the proposed model produced fewer misclassifications with an accuracy of 88.7% and using real-time data. Atole and Park (2018) developed a classification system for three diseases. They can achieve a 91.23% accuracy by using their deep learning algorithm and AlexNet based on 600 images of rice plants. They tested three different learning strategies on different CNN architectures for the classification of plant diseases on plant village data sets (Brahimi et al., 2018). This model has a 99.76% accuracy rate. Moreover, they have developed a method to visualise the saliency maps to understand the CNN classification method. Plant disease symptoms can be understood better in this model, which improves transparency among DL models.

Liu et al. (2020) used Leaf Generative Adversarial Networks (LGAN) to categorise four grape plant leaf diseases. A comparison of this model with other GANs, such as Deep Convolution GANs (DCGANs) and Wasserstein Generative Adversarial Networks (WGANs) suggests it is capable of better performance. A set of eight deep learning models were used as models for the experiments on the Pytorch framework. A better accuracy rating was achieved by XceptionNet with 98.70%. Fifty-nine diseases of different crops were studied by Hu et al. (2020) and a novel model was created called MDFC-ResNet. In this method, automatically identifying plant diseases is done through the use of an algorithm, which produces symptomatic output and enables farmers to react accordingly. Three dimensions make up this model: the fine-grained pathogen, species and coarse-grained pathogen. Researchers employed the Keras framework and designed experiments to examine how to improve the accuracy of their results, which was 85.22%.

Tetila et al. (2020) explored six diseases of soybean leaves and developed a classification system to identify them. Drones were used to take the images. It should be noted that in the course of conducting network training, to avoid overfitting, different methods were employed, such as data augmentation and dropout. By incorporating fine-tuning and Deep Learning techniques, they achieved an accuracy of 99.04%. Wu et al. (2020) proposed that using Google LeNet to classify images of tomato leaves is an effective method for the augmentation of data. A CNN architecture modified with different generative adversarial networks, adjusted hyperparameters and multiple generative adversarial networks were able to achieve a 94.33% accuracy rate.

To classify diseases and healthy leaves of mango plants using 2200 images, Singh et al. (2019) proposed a new CNN algorithm. 95.13% of the images were correctly classified. To detect five diseases from the apple leaf data set, Jiang et al.

(2019) proposed an improved CNN. The authors combined lab and field images and used annotations and data augmentation to achieve 78.80% mAP. The method of classifying five apple plant diseases based on three loss functions was proposed by Zhong and Zhao (2020) using the DenseNet-121 model. Cross-entropy is more effective in the classification process than loss functions. Xiong et al. (2020) proposed an algorithm capable of automatically segmenting images using an automatic technique called MobileNet CNN (MCNN). For detecting cash crop diseases, they developed a mobile-based smart device. Approximately 80% of samples were correctly identified using the proposed method.

Sambasivam and Opiyo (2020) proposed detecting five diseases of cassava leaves with class-labelled 10,000 images collected. The researchers used weighting, focus loss and Synthetic Minority Oversampling Technique (SMOT) to achieve over 93% accuracy. Based on INC-VGG, Junde et al. (2020) developed a model for classifying rice and maize diseases. To enhance the ability to extract features, the proposed model replaces the last two VGG-19 layers with Inception modules. This improved performance of rice and maize. A CNN model named depthwise separable was proposed by Kc et al. (2019) for classifying plants' leaf diseases. By using MobileNet, the accuracy of classification was 98.34% rather than the previous VGGNet level. Deep Learning with feature extraction is proposed as a way to classify millet diseases (Coulibaly et al., 2019). There are 124 millet leaf images used for the experiments, and the experiments are conducted with a 95% degree of accuracy using the Keras framework.

Twenty-five (25) diseases of 58 crops were classified according to VGGNet by Ferentinos (2018). Based on the results of the experiments, 99.5% of classification accuracy was achieved on the torch framework. To classify the fungi that infect wheat leaves, Picon et al. (2018) proposed DCNN. An accuracy of 96% was achieved in three consecutive years by collecting real-time images at different locations. Four grape leaf diseases were classified using an improved DCNN by Xie et al. (2020). The experiment included 4449 images of grape leaves. With 81.1% mean Average Precision (mAP), the Inception-ResNet-v2 module produced better results.

To detect the location of tomato plant diseases and classify the diseases under natural conditions, Liu and Wang (2020) proposed an improved YOLOv3 model. CaffeNet and the darknet framework have been tested and showed an accuracy of 92.39%. To classify tomato plant diseases, Fuentes et al. (2018) used a framework called Filter Bank. There are three units in the system: the first unit creates bounding boxes to pinpoint the location and class of the infected area. In the second unit, CNN Filter Bank is used to eliminate misclassified samples. Lastly, the third unit combines data from the first and second units such as True Positives and False Positives. It is 90% reliable.

An Intuitionistic Fuzzy Random Vector Functional Link (IFRVFL) classifier is proposed by Mishra et al. (2022). The IFRVFL classifier is a hybrid model that combines the advantages of the Random Vector Functional Link (RVFL) neural network and the Intuitionistic Fuzzy Set (IFS) theory. The experimental results show that the proposed IFRVFL classifier outperforms several existing classifiers in terms of classification accuracy and robustness to noisy data. The proposed classifier can be applied to various real-world applications, including image recognition, speech recognition and pattern recognition.

3 Data sets

Among the 12 plants chosen for this purpose are Mango, Arjun, Alstonia Scholaris, Guava, Bael, Jamun, Jatropha, Pongamia pinnata, Basil, Pomegranate, Lemon and Chinar the images from both healthy and diseased plants have been acquired and alienated leaf. The entire collection of images is divided into two classes, namely healthy and diseased. Figure 1 represents samples of diseased and healthy leaves. Using classified and labelled images, the plants are classified and labelled.

Figure 1 Sample images from the data set

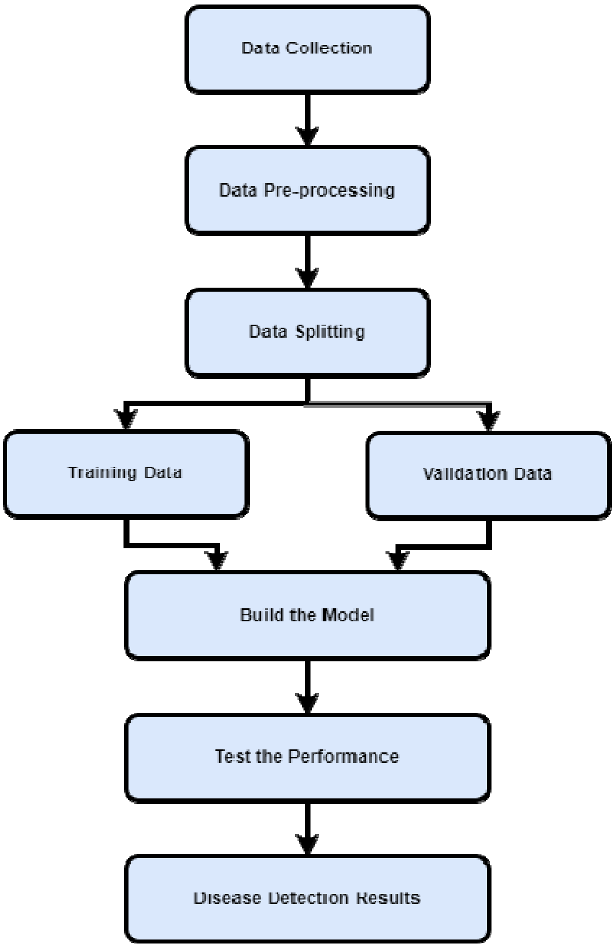
P0 to P11 are the plant names. There are 22 subject categories in the entire data set ranging from 0000 to 0022. Healthy classes were from 0012 to 0022, while diseased classes were from 0012 to 0022. The 4503 images were collected and made up of 2278 images of healthy leaves and 2225 images of diseased leaves. The authors from Shri Mata Vaishno Devi University, Katra, provided the leaf images (Chouhan et al., 2019). In the year 2019, the process was conducted from March to May. Image capture occurs in a closed environment. Wi-fi was used throughout. 58 seconds per frame was captured in JPEG in single shot mode and 63 seconds in RAW+JPEG mode with a Nikon D5300 camera. The images were captured with an 18–55 mm lens, 24-bit depth, two resolution units, 1000-ISO and no flash. This study may further benefit scientists and academicians in the development of methods for plant identification, plant classification, plant growth monitoring and leave disease diagnosis. Lastly, the anticipated impression will be towards a better understanding of what will be planted and how it will be managed.

4 Methodology

Figure 2 shows the block diagram that summarises the process of developing the proposed technique that is inspired by ResNet-50 architecture, a type of CNN. The following provides details on capturing images, processing them, training them and testing them.

- Images acquired.
- The second step deals with pre-processing of all images in the data set which includes cropping the image and data augmentation techniques namely shearing, rotation, etc.
- Assign labels to the data sets to separate them into validation.
- Training and testing.
- The deep learning models are built to execute the classification.
- Collect the classification accuracy of each model.
- Providing the results.

Figure 2 Procedural flow of work



4.1 Data pre-processing

After image acquisition, the image is possessed according to the proposed methodology.

- Calculating ROI by trimming the sample.
- Re-scale by 1/255 using the scaling factor.
- Shearing applies a factor of 0.5.
- Horizontally shifting by 0.32.
- Shifted by 0.18 units vertically.
- Rotating by 35° the sample.
- 0.2-fold zooming of the image.
- The horizontal flip of an image sample.
- Rotating by 35° the sample.
- 0.2-fold zooming of the image.
- The horizontal flip of the image sample.

When conducting experiments for mentioned deep learning models, the following system configurations are there:

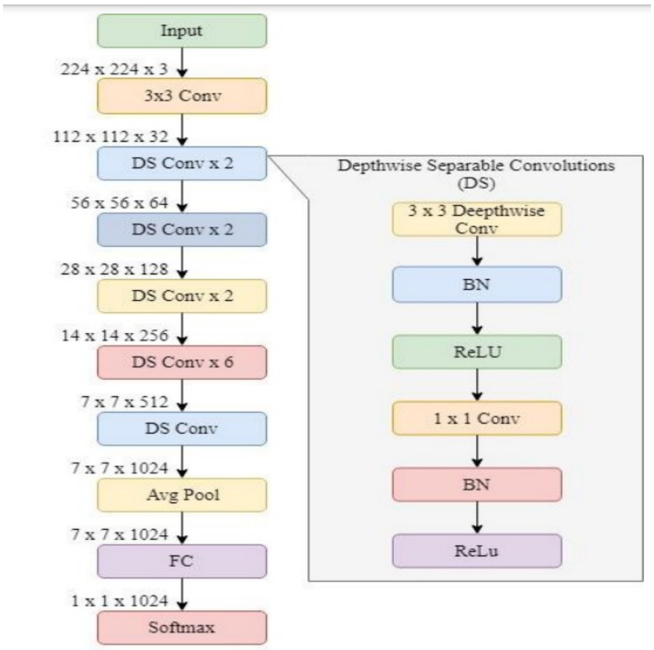
- *Processor*: Intel Core i7 is used. In addition, specialised hardware like Graphics Processing Units (GPUs) is used to accelerate the training process.
- *Memory*: Deep learning models require large amounts of memory to store the weights and activations of the network during training. The system has 16 GB of RAM.
- *Storage*: Training deep learning models requires a significant amount of storage to store the training data, model weights and other files. Solid-state drives (SSDs) are used for fast data access.
- *Software*: Several software packages are used to train deep learning models, including an operating system, a deep learning framework such as TensorFlow or PyTorch and the necessary drivers and libraries for the GPU.
- *Data*: The training data is an essential component of deep learning experiments, and large data sets are required to train complex models. The data is prepared and pre-processed before training can begin.
- *Hyperparameters*: The hyperparameters of the model, such as learning rate, batch size and optimiser, are carefully selected to ensure optimal performance.

4.2 Deep learning

4.2.1 MobileNet

An image classification algorithm called MobileNet is based on the CNN model. It is the lower computational capability of the MobileNet architecture that makes it suitable for working on mobile devices and computers with lower capabilities than the conventional CNN model. As for the MobileNet model, it is a simplified model that incorporates a convolution layer that can be used for separating details based on two manageable features that effectively control the parameters' accuracy and latency. Reduced network size is a benefit of the MobileNet model. Figure 3 shows the architecture of MobileNet (Srinivasu et al., 2021).

Figure 3 Mobilenet architecture

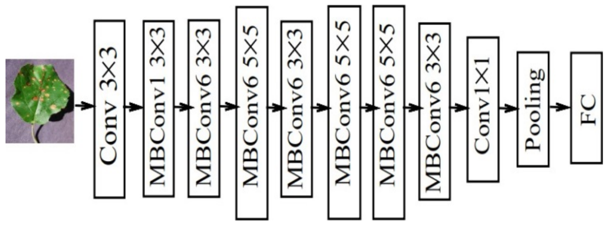


Source: Phiphiphatphaisit and Surinta (2020)

4.2.2 *EfficientNetB0*

In EfficientNets, Figure 4, the method of uniformly scaling the network's width, height and resolution with compound coefficients is presented. Using a baseline network, EffectiveNetB0 scales width, height and resolution. In total, 4,007,548 trainable parameters were fine-tuned, and extracted 1280 features for each of our 7×7 kernels (Makanapura et al., 2022).

Figure 4 EfficietNetB0 architecture

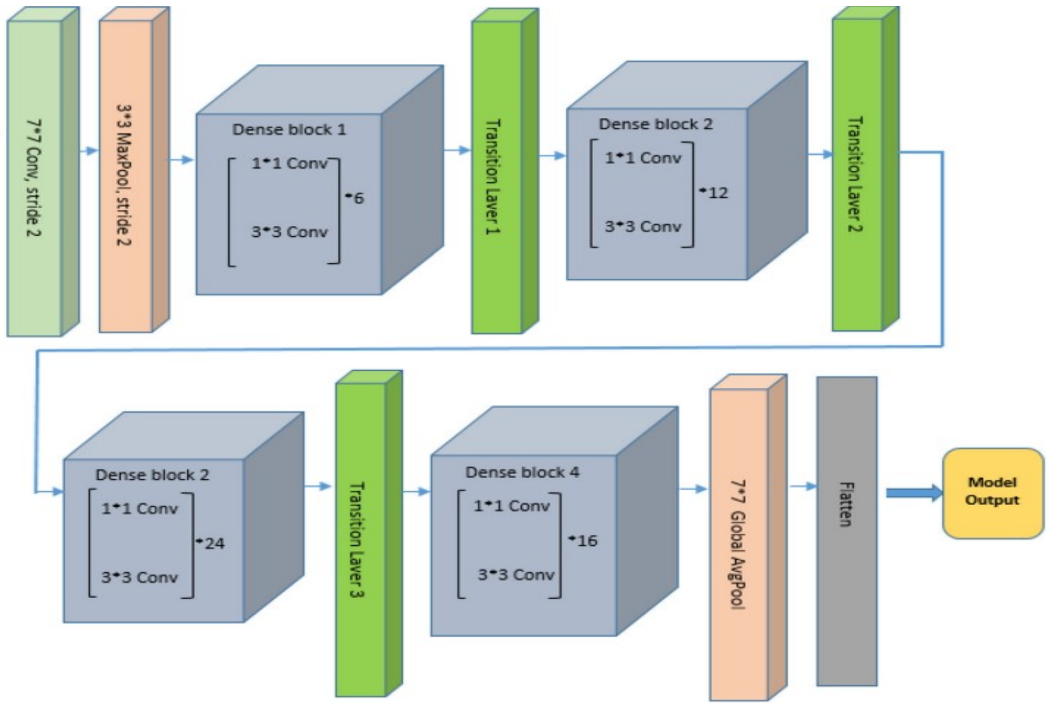


Source: Hassan et al. (2021)

4.2.3 *DenseNet121*

This network is composed of four high-density pools, followed by a 1×1 convolution layer and a pool average layer. Dense blocks are composed of multiple 1×1 and 3×3 convolutional layers, ranging from 6, 12, 32 and 32 layers, respectively. During dense blocks, the output is forwarded to the next dense block as input. Within the hidden layers, DenseNet-121 uses ReLU as well. As a final layer, it has a Softmax layer that is based on the global average pool. The integrity of DenseNet121's layers influences its accuracy, and DenseNet121's layers connect more tightly together for efficient training. It is evident from this that DenseNet-121 possesses the advantages of reducing the vanishing-gradient problem, strengthening feature propagation and reducing the number of parameters (Suwarningsih et al., 2022).

Figure 5 DenseNet121 architecture (see online version for colours)



Source: Hira et al. (2020)

4.2.4 ResNet-50

A previous deep learning method called ResNet-50 is used as a base. An accumulation of pre-trained models can be applied instead of a model with no knowledge of images. A reason for using ResNet-50 in biomedical images is that it has been successful. Additionally, it allows the training of data with fewer data sets, thereby requiring less computational effort (Cinar et al., 2021). As the data is received from the data set, it is fed into the proposed model with one extra Dense Layer. Figure 6 represents the architecture of ResNet-50.

4.2.5 Modified ResNet-50

Dense networks consist of layers that are deeply connected, meaning that neurons in each layer receive input from neurons in the layer above. According to the models, the dense layer is the most commonly used. The dense layer multiplies the matrix and vector in the background. Figure 7 shows the final architecture and the added layers.

Figure 6 ResNet-50 architecture (see online version for colours)

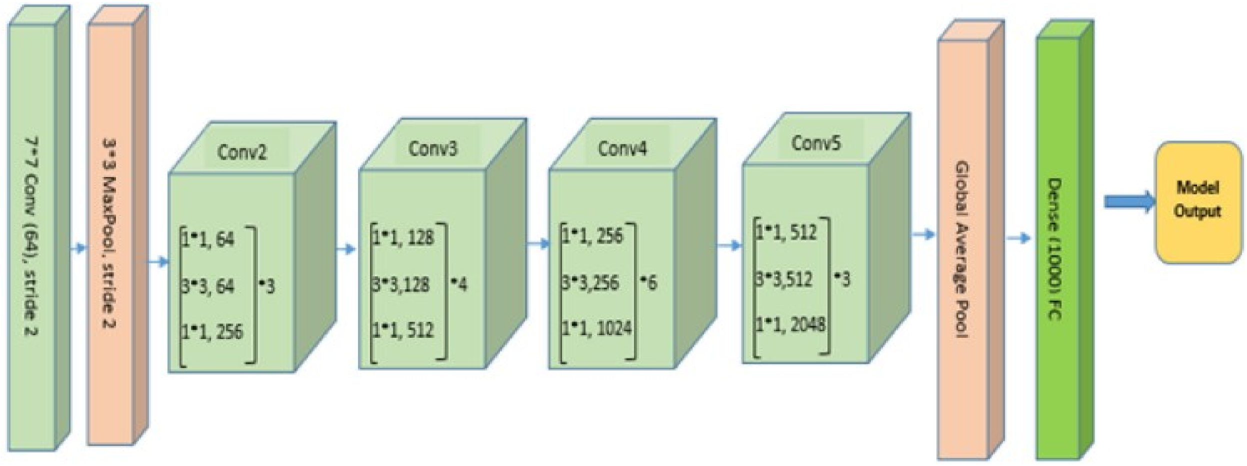
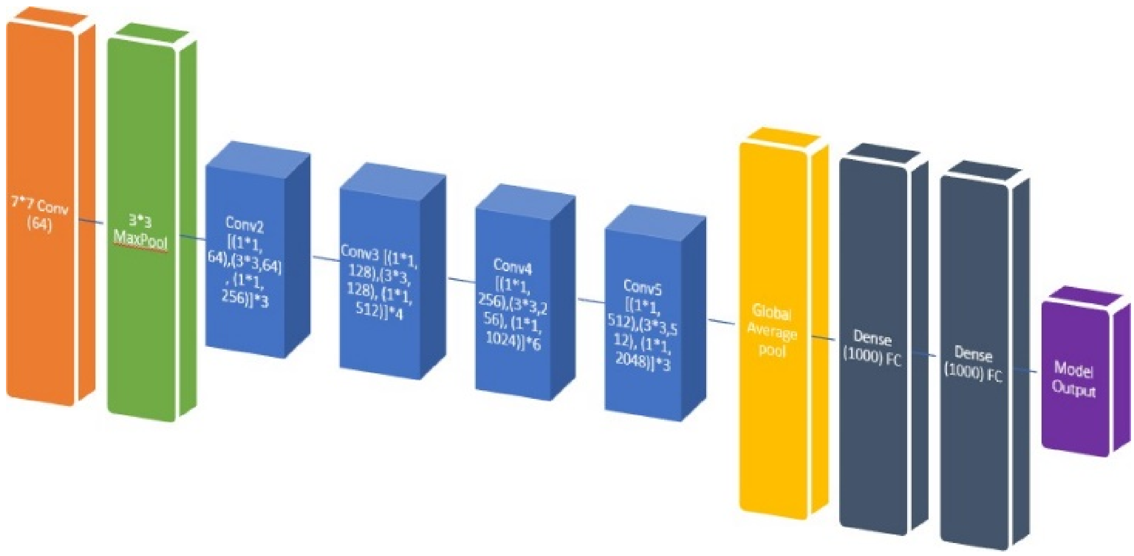


Figure 7 Modified ResNet-50 architecture (see online version for colours)



5 Results

In Table 1 and Figure 8, results can be seen of the deep learning models namely MobileNet, DenseNet121, EfficientNetB0 ResNet-50 and Modified ResNet-50. An assessment of a deep learning model's fit to training data is done by measuring the training loss. It represents the behaviour of Precision, Recall, F1-Score and Accuracy of the mentioned techniques.

Table 1 Performance measure of five architectures.

Architectures	*Precision*	*Recall*	*F1-Score*	*Accuracy*	*Computational Time*
Mobilenet	0.33	0.44	0.38	0.90	1450 s/epoch
EfficientnetB0	0	0	0	0.89	1500s/epoch
DenseNet121	0.78	1	0.88	0.90	1750 s/epoch
ResNet-50	0.78	1	0.93	0.95	1600 s/epoch
Modified ResNet-50	1	1	1	0.98	1175 s/epoch

Based on the values obtained in the table, the performance of each model can be interpreted as follows:

1 *Mobilenet*: The model has a precision of 0.33, a recall of 0.44, an F1-score of 0.38 and an accuracy of 0.90. The precision and recall are relatively low, indicating that the model is not performing well in correctly identifying the positive class. The F1-score is also low, which is a harmonic mean of precision and recall. The accuracy is relatively high, but it may not be an accurate representation of the model's performance due to the imbalanced nature of the data set.

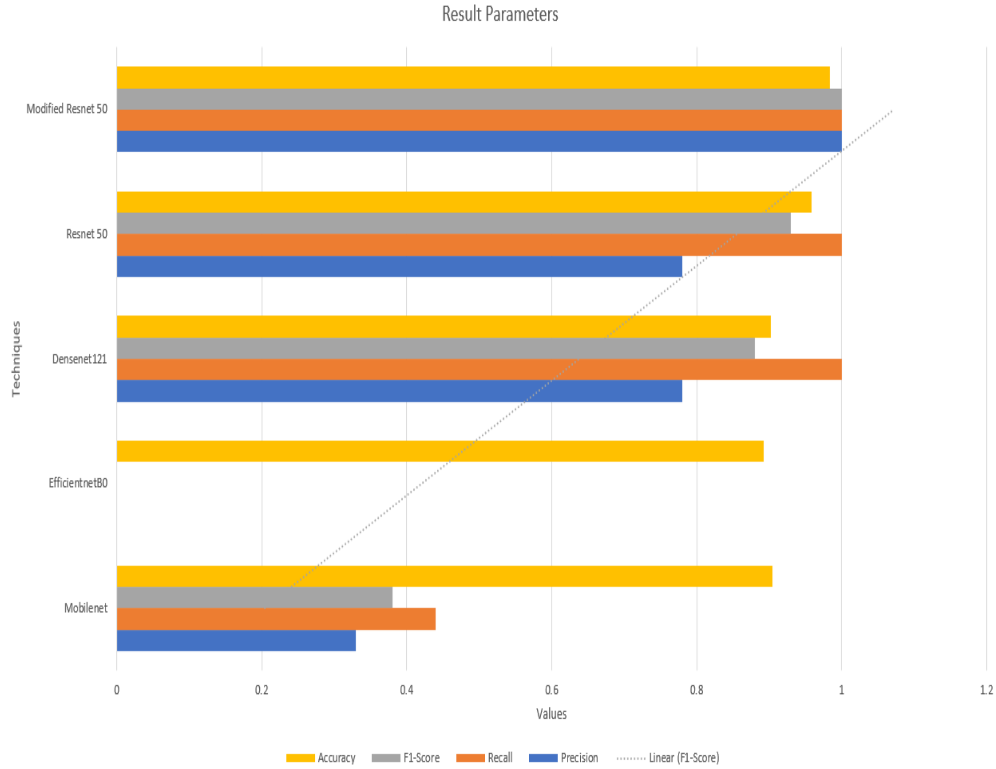
2 *EfficientnetB0*: The model has a precision, recall and F1-score of 0, indicating that the model is not able to identify any positive instances. The accuracy is 0.89, which may not be an accurate representation of the model's performance due to the imbalanced nature of the data set.

3 *DenseNet121*: The model has a precision of 0.78, a recall of 1, an F1-score of 0.88 and an accuracy of 0.90. The precision is relatively high, indicating that the model is performing well in correctly identifying the positive class. The recall is 1, which indicates that the model can identify all the positive instances. The F1 score is also high, indicating a good balance between precision and recall.

4 *ResNet-50*: The model has a precision of 0.78, a recall of 1, an F1-score of 0.93 and an accuracy of 0.95. The precision and recall are relatively high, indicating that the model is performing well in correctly identifying the positive class. The F1 score is also high, indicating a good balance between precision and recall. The accuracy is relatively high, which is a good indicator of the model's overall performance.

5 *Modified ResNet-50*: The model has a precision, recall and F1-score of 1, indicating that the model is performing perfectly in correctly identifying the positive class. The accuracy is relatively high, which is a good indicator of the model's overall performance. However, it is worth noting that a perfect score on precision, recall and F1-score may indicate overfitting on the training data, and the model's performance on new data should be evaluated carefully.

Figure 8 Performance parameters of Mobilenet, EfficientnetB0, Densenet12, ResNet-50 and Modified ResNet-50 (see online version for colours)



The model is evaluated by measuring its error on the training set. Training sets are used to train models from a data set. The training loss is calculated computationally by taking the sum of all the errors in the training set. Deep learning models are evaluated based on their performance on a validation set using validation loss metrics. To validate the model, a validation set is set aside from the data set. We know the ratio of training and test data size should be chosen based on the specific characteristics of the data set and the requirements of the model, thus the ratio of the train to validation data size is taken as 80:20. As the data set size is large for this work, the factors like adequate training, overfitting avoidance, statistically significant results are taken under consideration while splitting the data set in the mentioned ratio.

Similarly, to the training loss, the validation loss is derived from the sum of errors for each example in a validation set. When validation loss exceeds training loss, it is called validation loss overtraining. An underfit model may appear as this. The model underfits when it cannot reproduce accurately the training data, which leads to large errors. An overfitted model cannot generalise on new data when the validation loss is greater than the training loss.

Validation losses decrease and then increase again after a certain point. One reason for this may be that the model was trained for a long time or was too complex for the data

There is a training loss of 0.6% and a 5.51% validation loss. Then, Figure 8 displays the graphs of Accuracy and Loss for the training and validation phases. As the number of epochs increased, the training accuracy increased to 99.78 on the 10th epoch. Similar to epoch1, the accuracy of validation rose from 90 to 98.38% at the 10th epoch. From the graph presented, it was concluded that the proposed approach is successful in attaining good accuracy and the loss is considerably less as compared to the recent works.

Figure 9 Comparison graphs of training accuracy, training loss, validation accuracy and validation loss (see online version for colours)

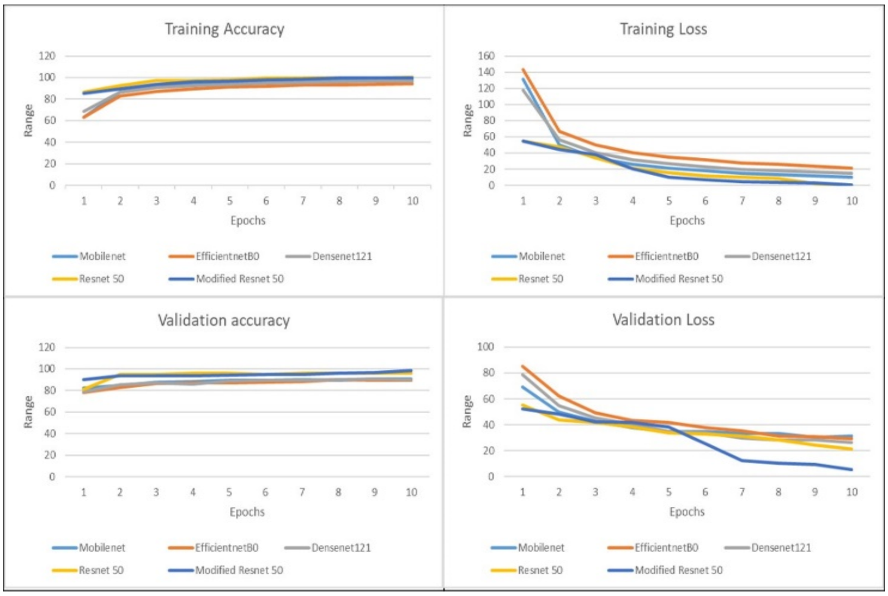


Figure 10 shows the code snippet for modified ResNet-50.

Figure 10 Modified ResNet-50 implementation

```
from keras import callbacks

my_callbacks=callbacks.EarlyStopping(monitor='val_loss', patience=2,verbose=0, mode='min',restore_best_weights=True)

[ ] model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001,decay=0.1), loss ="categorical_crossentropy", metrics = ['accuracy'])

history=model.fit_generator(train_generator,validation_data=test_generator,epochs=10,callbacks=[my_callbacks,check_point])

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  """Entry point for launching an IPython kernel.
Epoch 1/10
118/118 [==============================] - ETA: 0s - loss: 0.6551 - accuracy: 0.8067 WARNING:tensorflow:Can save best model only with val_acc available, skipping.
118/118 [==============================] - 2476s 21s/step - loss: 0.6551 - accuracy: 0.8067 - val_loss: 0.9727 - val_accuracy: 0.6716
Epoch 2/10
118/118 [==============================] - ETA: 0s - loss: 0.0970 - accuracy: 0.9725WARNING:tensorflow:Can save best model only with val_acc available, skipping.
118/118 [==============================] - 1168s 10s/step - loss: 0.0970 - accuracy: 0.9725 - val_loss: 0.3291 - val_accuracy: 0.8848
Epoch 3/10
118/118 [==============================] - ETA: 0s - loss: 0.0447 - accuracy: 0.9875WARNING:tensorflow:Can save best model only with val_acc available, skipping.
118/118 [==============================] - 1163s 10s/step - loss: 0.0447 - accuracy: 0.9875 - val_loss: 0.2200 - val_accuracy: 0.9265
Epoch 4/10
118/118 [==============================] - ETA: 0s - loss: 0.0256 - accuracy: 0.9955WARNING:tensorflow:Can save best model only with val_acc available, skipping.
118/118 [==============================] - 1172s 10s/step - loss: 0.0256 - accuracy: 0.9955 - val_loss: 0.2322 - val_accuracy: 0.9240
Epoch 5/10
118/118 [==============================] - ETA: 0s - loss: 0.0185 - accuracy: 0.9968WARNING:tensorflow:Can save best model only with val_acc available, skipping.
118/118 [==============================] - 1171s 10s/step - loss: 0.0185 - accuracy: 0.9968 - val_loss: 0.1969 - val_accuracy: 0.9338
Epoch 6/10
118/118 [==============================] - ETA: 0s - loss: 0.0117 - accuracy: 0.9995WARNING:tensorflow:Can save best model only with val_acc available, skipping.
118/118 [==============================] - 1180s 10s/step - loss: 0.0117 - accuracy: 0.9995 - val_loss: 0.2088 - val_accuracy: 0.9265
Epoch 7/10
118/118 [==============================] - ETA: 0s - loss: 0.0098 - accuracy: 0.9992WARNING:tensorflow:Can save best model only with val_acc available, skipping.
118/118 [==============================] - 1175s 10s/step - loss: 0.0098 - accuracy: 0.9992 - val_loss: 0.2100 - val_accuracy: 0.9314
```

Certainly, the optimal parameters for the deep learning models used in this work are mentioned in Table 2.

Table 2 Optimised parameters

Hyperparameter	*Value*
Base Model	MobileNet
Image Dimensions	224×224
Batch Size	32
Epochs	10
Optimiser	Adam
Learning Rate	0.001
Weight Decay	0
Dropout	0
Activation Function	Softmax
Loss Function	Categorical Crossentropy

6 Conclusions

Deep learning models have been developed to identify plant diseases by using simple images of healthy and diseased leaves. It is concluded that a convolutional neural network with a ResNet-50 with Dense Layer had the highest success rate in the detection of plant leaves (test set) that previously were unknown to the model.

In conclusion, our study aimed to explore the performance of various deep learning models for a specific task, and we have presented our findings in this paper. The results show that the modified ResNet-50 model outperformed the other models in terms of precision, recall, F1-score and accuracy, while EfficientnetB0 performed poorly. This information can be useful for researchers and practitioners working on similar tasks and can guide them in selecting the most suitable model for their needs.

Moreover, this study provides insights into the impact of system configurations on the performance of deep learning models. We varied the batch size, learning rate and the number of epochs and evaluated their impact on the performance of the models. These insights can be used to optimise the configuration of deep learning models for similar tasks.

The scientific value added by this paper is the empirical evaluation of several deep learning models for a specific task and the identification of the most suitable model for the task. Our findings can provide a basis for future research in the area and help advance the field of deep learning for this task.

Our study has some limitations that should be noted. Firstly, we evaluated the models on a specific data set, and the performance may differ for other data sets. Secondly, we only varied a few system configurations, and there may be other configurations that could impact the performance of the models. Lastly, our study only evaluated a few deep learning models, and other models may perform better for the task. The data collection for training purposes should be broader, drawing from a variety of geographical areas, cultivation conditions, and imaging modes. For this deep learning approach to be improved and to be more robust and wider-ranging (both in terms of identifying more species and diseases), it needs both qualitative and quantitative data.

Furthermore, the trained model requires little computation, dependent on the GPU, so it is feasible to integrate it into mobile applications. In future studies using smartphones or drones, agricultural professionals or growers could monitor open-field operations in real time and use disease detection technology to monitor diseases dynamically. It would be possible for the farmer to purchase appropriate pesticides if an automated system of pesticide prescriptions took into consideration the automatic disease diagnostic system in the case of the farmer who was purchasing the pesticides. Likewise, agriculturalists would also be able to benefit from this development, since they would get an incipient warning about a potential threat to their crops. Consequently, a significant reduction in pesticide misuse and uncontrollable use would be prevented, thus avoiding environmental consequences that are catastrophic.

References

Atole, R.R. and Park, D. (2018) 'A multiclass deep convolutional neural network classifier for detection of common rice plant anomalies', *International Journal of Advanced Computer Science and Applications*, Vol. 9, No. 1, pp.67–70. Doi: 0.14569/ijacsa.2018.090109.

Brahimi, M., Arsenovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K. and Moussaoui, A. (2018) 'Deep learning for plant diseases: detection and saliency map visualization', *Human and Machine Learning*, Springer, pp.93–117. Doi: 10.1007/978-3-319-90403-0 6.

Brahimi, M., Boukhalfa, K. and Moussaoui, A. (2017) 'Deep learning for tomato diseases: classification and symptoms visualization', *Applied Artificial Intelligence*, Vol. 31, No. 4, pp.299–315. Doi: 10.1080/08839514. 2017.1315516.

Chouhan, S.S., Kaul, A., Singh, U.P. and Jain, S. (2019) 'A database of leaf images: practice towards plant conservation with plant pathology', *Mendeley Data*, Vol. 6. Doi: 10.17632/hb74ynkjcn.

Cınar, A., Yıldırım, M. and Ero˘glu, Y. (2021) 'Classification of pneumonia cell images using improved ResNet50 model', *Traitement du Signal*, Vol. 38, No. 1, pp.165–173. Doi: 10.18280/ts.380117.

Coulibaly, S., Kamsu-Foguem, B., Kamissoko, D. and Traore, D. (2019) 'Deep neu- ral networks with transfer learning in millet crop images', *Computers in Industry*, Vol. 108, pp.115–120. Doi: 10.1016/j.compind.2019.02.003.

Dasari, S. and Prasad, V. (2019) 'A novel and proposed comprehensive methodology using deep convolutional neural networks for flue-cured tobacco leaves classification', *International Journal of Information Technology*, Vol. 11, No. 1, pp.107–117. Doi: 10.1007/s41870-018-0174-4.

Dhingra, G., Kumar, V. and Joshi, H. (2018) 'Study of digital image processing techniques for leaf disease detection and classification', *Multimedia Tools and Applications*, Vol. 77, No. 15, pp.19951–20000. Doi: 10.1007/ s11042-017-5445-8.

Durmus, H., Gunes, E. and Kirci, M. (2017) 'Disease detection on the leaves of the tomato plants by using deep learning', *Proceedings of the 6th International Conference on Agro-Geoinformatics*, IEEE, USA, pp.1–5. Doi: 10.1109/agromatics.2017.8047016.

Ferentinos, P. (2018) 'Konstantinos: deep learning models for plant disease detection and diagnosis', *Computers and Electronics in Agriculture*, Vol. 145, pp.311–318. Doi: 10.1016/j.compag.2018.01.009.

Fuentes, A., Yoon, S., Lee, J. and Park, D. (2018) 'High-performance deep neural network-based tomato plant diseases and pests diagnosis system with the refinement filter bank', *Frontiers in Plant Science*, Vol. 9, No. 1162, pp.1–15. Doi: 10.3389/fpls.2018.01162.

Hassan, S.M., Maji, A.K., Jasin´ski, M., Leonowicz, Z. and Jasin´ska, E. (2021) 'Identification of plant-leaf diseases using CNN and transfer-learning approach', *Electronics*, Vol. 10, No. 12. Doi: 10.3390/ electronics10121388.

Hira, S., Bai, A. and Hira, S. (2020) 'An automatic approach based on CNN architecture to detect covid-19 disease from chest x-ray images', *Applied Intelligence*, Vol. 51, No. 5, pp.2864–2889. Doi: 10.1007/ s10489-020-02010-w.

Hu, W-J., Fan, J., Du, Y-X., Li, B-S., Xiong, N. and Bekkering, E. (2020) 'Mdfc–Resnet: an agricultural IOT system to accurately recognize crop diseases', *IEEE Access*, Vol. 8, pp.115287–115298. Doi: 10.1109/ access.2020.3001237.

Jain, L., Harsha Vardhan, M.A., Nishanth, M.L. and Shylaja, S.S. (2017) 'Cloud-based system for supervised classification of plant diseases using convolutional neural networks', *Proceedings of the IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, IEEE, India. Doi: 10.1109/ccem.2017.22.

Jiang, P., Chen, Y., Liu, B., He, D. and Liang, C. (2019) 'Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks', *IEEE Access*, Vol. 7, pp.59069–59080. Doi: 10.1109/access.2019.2914929.

Junde, C., Chen, J., Zhang, D., Sun, Y. and Nanehkaran, Y.A. (2020) 'Using deep transfer learning for image-based plant disease identification', *Computers and Electronics in Agriculture*. Doi: 10.1016/j.compag.2020.105393.

Kamilaris, A. and Prenafeta-Boldu, F. (2018) 'Deep learning in agriculture: a survey', *Computers and Electronics in Agriculture*, Vol. 147, pp.70–90. Doi: 10.1016/j.compag.2018.02.016.

Kaur, S., Pandey, S. and Goel, S. (2019) 'Plants disease identification and classification through leaf images: a survey', *Archives of Computational Methods in Engineering*, Vol. 26, No. 2, pp.507–530. Doi: 10.1007/ s11831-018-9255-6.

Kc, K., Yin, Z., Wu, M. and Wu, Z. (2019) 'Depthwise separable convolution architectures for plant disease classification', *Computers and Electronics in Agriculture*, Vol. 165. Doi: 10.1016/j.compag.2019.104948.

Liu, B., Tan, C., Li, S., He, J. and Wang, H. (2020) 'A data augmentation method based on generative adversarial networks for grape leaf disease identification', *IEEE Access*, Vol. 8, pp.102188–102198. Doi: 10.1109/ access.2020.2998839.

Liu, J. and Wang, X. (2020) 'Tomato diseases and pests detection based on improved YOLO v3 convolutional neural network', *Frontiers in Plant Science*, Vol. 11, No. 898, pp.1–12. Doi: 10.3389/fpls.2020.00898.

Makanapura, N., Sujatha, C., Patil, P.R. and Desai, P. (2022) 'Classification of plant seedlings using deep convolutional neural network architectures', *Journal of Physics: Conference Series*, Vol. 2161, No. 1. Doi: 10. 1088/1742-6596/2161/1/012006.

Mishra, U., Gupta, D. and Hazarika, B.B. (2022) 'An intuitionistic fuzzy random vector functional link classifier', *Neural Processing Letters*, pp.1–22.

Nagaraju, M. and Chawla, P. (2020) 'Systematic review of deep learning techniques in plant disease detection', *International Journal of System Assurance Engineering and Management*, Vol. 11, No. 3, pp.547–560. Doi: 10. 1007/s13198-020-00972-1.

Phiphiphatphaisit, S. and Surinta, O. (2020) 'Food image classification with improved MobileNet architecture and data augmentation', *Proceedings of the 3rd International Conference on Information Science and Systems*, ACM. Doi: 10.1145/3388176.3388179.

Picon, A., Alvarez-Gila, A., Seitz, M., Ortiz-Barredo, A., Echazarra, J. and Johannes, A. (2018) 'Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild', *Computers and Electronics in Agriculture*, Vol. 161, pp.280–290. Doi: 10.1016/j. compag.2018.04.002.

Rehman, T., Mahmud, M., Chang, Y., Jin, J. and Shin, J. (2019) 'Current and future applications of statistical machine learning algorithms for agricultural machine vision systems', *Computers and Electronics in Agriculture*, Vol. 156, pp.585–605. Doi: 10.1016/j.compag.2018.12.006.

Saleem, P. and Arif, M. (2019) 'Plant disease detection and classification by deep learning', *Plants*, Vol. 8, No. 11. Doi: 10.3390/plants8110468.

Sambasivam, G. and Opiyo, G. (2020) 'A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks', *Egyptian Informatics Journal*, Vol. 22, No. 1, pp.27–34. Doi: 10.1016/j.eij.2020.02.007.

Singh, G., Rani, R., Sharma, N. and Kakkar, D. (2017) 'Identification of tomato leaf diseases using deep convolutional neural networks', *International Journal of Agricultural and Environmental Information Systems*, Vol. 12, No. 4, pp.1–22. Doi: 10.4018/ijaeis.20211001.oa3.

Singh, U., Chouhan, S., Jain, S. and Jain, S. (2019) 'Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease', *IEEE Access*, Vol. 7, pp.43721–43729. Doi: 10.1109/access. 2019.2907383.

Srinivasu, P.N., SivaSai, J.G., Ijaz, M.F., Bhoi, A.K., Kim, W. and Kang, J.J. (2021) 'Classification of skin disease using deep learning neural networks with MobileNet v2 and LSTM', *Sensors*, Vol. 21, No. 8. Doi: 10.3390/s21082852.

Suwarningsih, W., Khotimah, P.H., Rozie, A.F., Arisal, A., Riswantini, D., Nugraheni, E., Munandar, D. and Kirana, R. (2022) 'Ide-Cabe: chili varieties identification and classification system based leaf', *Bulletin of Electrical Engineering and Informatics*, Vol. 11, No. 1, pp.445–453.

Tetila, E., Machado, B., Menezes, G., Da Silva Oliveira, A., Alvarez, M., Amorim, W., De Souza Belete, N., Da Silva, G. and Pistori, H. (2020) 'Automatic recognition of soybean leaf diseases using UAV images and deep convolutional neural networks', *IEEE Geoscience and Remote Sensing Letters*, Vol. 17, No. 5, pp.903–907. Doi: 10.1109/lgrs.2019.2932385.

Wu, Q., Chen, Y. and Meng, J. (2020) 'Dcgan-based data augmentation for tomato leaf disease identification', *IEEE Access*, Vol. 8, pp.98716–98728. Doi: 10.1109/access.2020.2997001.

Xie, X., Ma, Y., Liu, B., He, J., Li, S. and Wang, H. (2020) 'A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks', *Frontiers in Plant Science*, Vol. 11, No. 751, pp.1–14. Doi: 10.3389/fpls.2020.00751.

Xiong, Y., Liang, L., Wang, L., She, J. and Wu, M. (2020) 'Identification of cash crop diseases using automatic image segmentation algorithm and deep learning with the expanded dataset', *Computers and Electronics in Agriculture*, Vol. 177. Doi: 10.1016/j.compag.2020.105712.

Zhang, X., Qiao, Y., Meng, F., Fan, C. and Zhang, M. (2018) 'Identification of maize leaf diseases using improved deep convolutional neural networks', *IEEE Access*, Vol. 6, pp.30370–30377. Doi: 10.1109/access.2018. 2844405.

Zhong, Y. and Zhao, M. (2020) 'Research on deep learning in apple leaf disease recognition', *Computers and Electronics in Agriculture*, Vol. 168. Doi: 10.1016/j.compag.2019.105146.