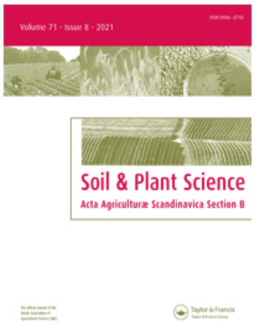
Volume 71 · Issue 8 · 2021
Soil & Plant Science
Acta Agriculturæ Scandinavica Section B

Taylor & Francis
Taylor & Francis Group

Acta Agriculturae Scandinavica, Section B — Soil & Plant Science

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/sagb20

Automatic recognition of tomato leaf disease using fast enhanced learning with image processing

Thanjai Vadivel & R. Suguna

To cite this article: Thanjai Vadivel & R. Suguna (2021): Automatic recognition of tomato leaf disease using fast enhanced learning with image processing, Acta Agriculturae Scandinavica, Section B — Soil & Plant Science, DOI: 10.1080/09064710.2021.1976266

To link to this article: https://doi.org/10.1080/09064710.2021.1976266

Published online: 25 Oct 2021.

ACTA AGRICULTURAE SCANDINAVICA, SECTION B — SOIL & PLANT SCIENCE
https://doi.org/10.1080/09064710.2021.1976266

ENVISAGE COMPUTER MODELLING AND STATISTICS FOR AGRICULTURE

OPEN ACCESS

Automatic recognition of tomato leaf disease using fast enhanced learning with image processing

Thanjai Vadivel and R. Suguna

Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India

ABSTRACT

The changes in weather have beneficial and harmful effects on crop yields. There will be a loss of yield because of the diseases in crops. With the growing population, the fundamental want of food is growing. That is why agriculture gains a prominent position all around the world. It eventually ends up by a massive defeat for the farmers and the financial boom of India. The article's primary goalis to bring together farmers and cutting-edge technologies to minimise diseases in plant leaves. To enforce the idea, 'Tomato' is selected in which leaf sicknesses are expected and identified by the Artificial Intelligence algorithms, CNN (Convolution Neural Network) with pc technological know-how. Tomato is a mere consumable vegetable in India. In this investigation, seven types of tomato leaf disorders were sensed, including one wholesome elegance. The farmers are able to check the symptoms with the shapes of images of the tomato leaves with those expecting diseases. Its comparison of various classification and filters/methods with different techniques, such as K-Means classifier, SVM (Support Vector), RBF(Radial Basis Function) Kernel, Optimised MLP(Multilayer perceptron), NN classifier, BPNN (back-propagation neural network) and CNN Classifier. The classification accuracy of the existing method after experiment is RBF – 89%, k-means – 85.3%, SVM – 88.8%, Optimised MLP – 91.4%, NN – 97, BPNN – 85.5%, CNN – 94.4%**. The proposed architecture can achieve the desired accuracy of 99.4%.**

ARTICLE HISTORY
Received 15 April 2021
Accepted 31 August 2021

KEYWORDS
Tomato leaves; RBF Kernel; Fast Enhanced Learning Method; image processing; image classification; CNN; K-means clustering; NN Classifier; SVM

Introduction

Advanced cultivation strategies to enhance the yield of vegetation are spreading in India. However, as farmers age, the farmer turnover costs will boom, so there may be a difficulty that the strategies of professional farmers can be misplaced before the new farmers can inherit them. Therefore, researchers have been conducting studies on the spread of superior cultivation strategies. Such research work has tried to breed tacit expertise of professional farmers, primarily based on environmental and photo information (Sudgen et al. 2021). Water-strain cultivation which is executed to deliberately lessen the quantity of irrigation, is based mainly on the experience and instinct of professional farmers and is called sophisticated cultivation technique. Global water pressure due to drought is identified because it is the riskiest strain to flora. Occasional greens are most exceptionally well known in the appraisal of the veggies developed extensively. Tomato is undoubtedly one of them. It is considered a consumer-friendly device so it can assist the vegetable farmers particularly the 'Tomato' cultivator to reduce repression by means organiser of nine classes that are made specifically: 'Bacterial spot', 'Healthy', 'Mosaic', 'Septoria spot ', 'Yellow curl', 'Septoria spot disease prediction', 'Septoria Spot' etc., of notice its leaf infections and increasing the give way by increasing the opportunities for numerous vegetable disease studies and specialised marketplace (Argüeso et al. 2020). The principal reason is to remedy the sickness detection troubles the 'Tomato' growers going through these days in their cultivable ground, particularly in India. That is why tomatoes leaf sickness prediction, which could be very important, has been selected. These studies try to eliminate the damaging face results of chemical compounds and insect killers with an image processing system (Sudgen et al. 2021).

CONTACT Thanjai Vadivel thanjaivadivel@veltech.edu.in Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India
This paper was prompted by the desire to create an automatic leaf extraction system that grows with the evolving technology of machine vision.

Contribution

Tomato is the tremendous stockpile of sustenance among the 15 most extreme flexible mainstream vegetables, which might be burned through severally. Tomato is the most excellent vegetable classification in the world contributing over 16 per cent%. Tomatoes have been named the sixth most abundant vegetable in the world, according to FAO, based on yearly production measurements. Tomato is a seasonal vegetable in India cultivated in a frigid climate. There's an incredible interest in tomatoes even in slow times of the year in our country (Arcidiacono 2020). A moderate contrast is found in the quality, price, demand, call for and rate in the tomato market all-round the year. Now and then, numerous ailments of tomatoes are noticeable because of creepy-crawly assaults, horrible climate, soil, polluted water, and harmful pesticides. Much sooner than tomatoes can be developed. Tomato cultivators get devastated because of the assault of bugs and disorders. Tomatoes leaf disorders are awful as well.

Motivation and justification

This paper is sure to find the leaf disease of tomatoes using refreshed innovation 'image processing'. Presently, not all those plants have been attacked by some disease. Farmers and ranchers could anticipate days of confusion and affirmation in the future. Agrarian agents may not have sufficient accessibility to offer virtual assistance to provincial ranchers in finding out which afflictions they were experiencing. Ranchers can examine insect killers, seeds and still disorders by telephone calls to the closest available agribusiness specialists (Nazki et al. 2020). Image preparing enabled overcoming that challenge easily by clicking images of the prejudiced plant and seeking information using Figure 1 in identifying the infection. The gadget ranchers can get some answers concerning which infections attack they might be confronting. For this, the gadget has to be pre-instructed with vital records. On this exploration, the five well-known tomato leaf diseases will be identified: Bacterial Spot, past due scourge, Septoria Leaf Spot, Tomato mosaic, Yellow bended along fortifying tomato leaf location. CNN is the most important for making image behaviour a satisfaction. CNN can be completed on any variety of class inconvenience like plant problem recognition. Our proposed methods are used in this model to forecast the tomato class diseases.

Outline of the proposed work

Figure 1 focuses on detecting diseases from the leaves of the tomato plants. 'Related Works' section focuses on the different transformations used to classify the tomato leaves disease. 'Algorithm for the Fast Enhanced Learning Method' section deals with the Enhancement of CNN Classification Techniques Implementation. 'Proposed Experimental Design' section focuses on the proposed methodology to represent the convolution network and its analysis for the performance in various techniques and our method using CNN.

They were later segmenting the dataset into disease and non-disease images. The images will be given to feature extraction there. We undergo processes texture, shape and colour extraction. Data will be classified using the proposed convolution network models. They were finally comparing the work with the existing algorithm with the proposed method.

In this exploration, 10,000 images have been selected to work with disease and non-disease of tomato plant leaf from plant village dataset (https://www.kaggle.com/abdallahalidev/plantvillage-dataset). Figure 2 contains sample tomato leaf diseases such as 'Bacterial spot', 'Healthy', 'Mosaic', 'Septoria spot ', 'Yellow curl', 'Septoria spot disease prediction', 'Septoria Spot' (https://www.kaggle.com/abdallahalidev/plantvillage-dataset)

Approximately 1000 legitimate information have been selected, which can be utilised for this examination from the dataset. It has isolated the dataset to pre-test and train envelope with 20% and 80%, respectively (Xie et al. 2020).

Infections have been separately named for convenience. The images have been resized (128 × 128) pixels off the images used for quicker calculations (Singh et al. 2021).

Related works

Convolution Neural Network (CNN) is perhaps the most remarkable out-of-the-case directed AI calculations. It is difficult to begin any exploration without an essential and sufficient dataset. Dataset is the centre piece of any exploration. To assure the most remarkable exactness and widespread acceptance of any investigation, it is necessary to gather as much information as possible. For this 'tomato leaves disease recognition', in excess of 10000 images have been gathered from as of late reaped yield and around 1000 image have been gathered from plant village dataset (https://www.kaggle.com/thanjaivadivelm/tomato). The Inception V3 network is fine-tuned using shallow SVM classifier to train a generic plant leaf feature extraction CNN (Argüeso et al. 2020).

'Convolutional neural organisations (CNN) can be utilised for the production of a computational model that

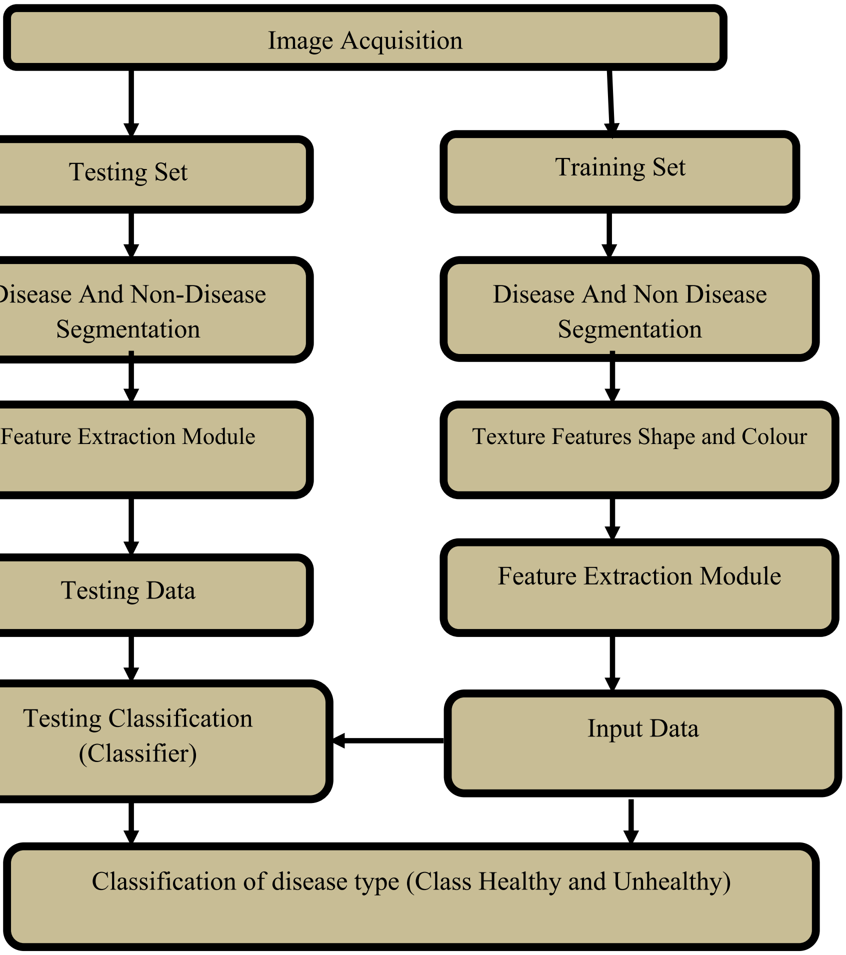
Image Acquisition
Testing Set
Training Set
Disease And Non-Disease Segmentation
Disease And Non Disease Segmentation
Feature Extraction Module
Texture Features Shape and Colour
Testing Data
Feature Extraction Module
Testing Classification (Classifier)
Input Data
Classification of disease type (Class Healthy and Unhealthy)

Figure 1. Classification of diseases of tomato leaves.

chips away at the unstructured picture information sources and converts them to relating grouping yield names. They have a place to classify multi-layer neural organisations, which can be prepared to get familiar with the necessary highlights for arrangement purposes. They require less pre-preparationg than the conventional methodologies and perform programmed include extraction to give better execution. CNN canform a computational model that chips away at the unstructured picture sources of info and convert them

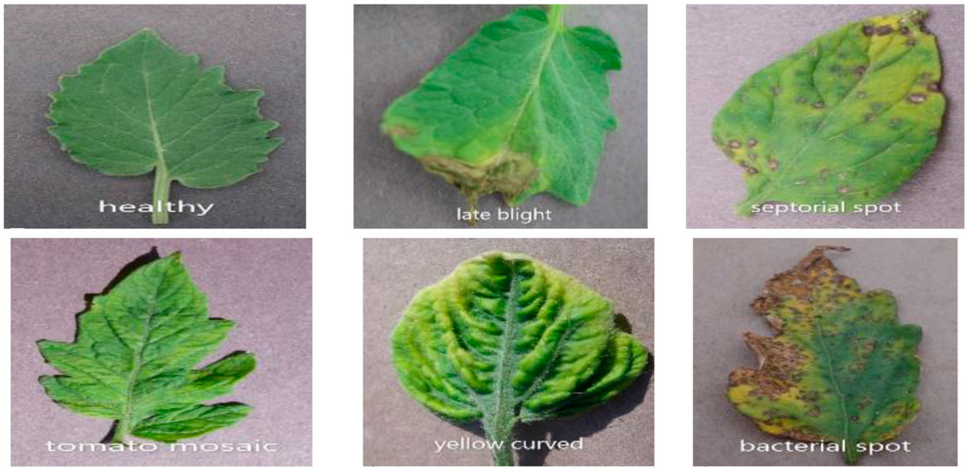

Figure 2. Sample tomato leaf dataset image.

to comparing grouping yield names. They have a place to classify multi-layer neural organisations, which can be prepared to become familiar with the necessary highlights for arrangement purposes. They require less prepreparation than customary methodologies and perform programmes including extraction of tomato leaf infection location (Pham et al. 2020).'

A CNN model accepts images as information. The exactness of picture handling is straightforwardly affected by those extricated highlights. The CNN model comprises layers, for example, Convolution, ReLU Layer, Pooling, Fully Connected, Flatten and Normalisation. Utilising CNN, we could look at the images piece by piece. Each piece is known as an element or channel. From the info picture, CNN utilises the weighted network and concentrates on the particular highlights without losing the data about its spatial game plan.

Figure 3 has input (128, 128, 3), channel size 64, part size (5 × 5), cushioning 'Equal', steps (1 × 1). It follows initiation ReLU '(1)', channel size 64, portion size (5 × 5), cushioning 'Same', steps (1 × 1) trailed by maxpool size (3 × 3), steps (3 × 3). Each conv layer follows a similar design, which is utilised for enacting the previous layers in each cluster model.

Input layer

The first input layer considered all the resized augmented data on this layer.

Convolutional layers

Convolutional layers are used for computng all tasks related to all layers, and it is also called the primary layer in CNN. It performs the task related to input convolution and sends the output of these layers to the next convolutional layer. The process continues to learn all the low-level features up to the high-level features and learn all the in-depth specific features.

Max-pooling layers

The max-pooling layers are presented between the convolutional layers, which are responsible for the reduction of pixel and computational reduction. The pooling layer works in almost every pixel to reduce the computational cost. It just works like filters that are applied to the image, which is responsible for noise reduction. It is also applied to the feature map. The pooling layer is smaller than the feature map. The pooling layer has 2×2 strides of 2 pixels, which means the reduction of the pooling layer will be the a factor of 2 of the feature map. In our methodology, the max-pooling layer is added between convolutional layers to determine the max value for the feature map.

Softmax layer

Softmax layer converts the given input from the previous layer into probabilities. Numerous multi-layer neural systems end in a penultimate layer that yields genuine esteemed scores that are not helpfully scaled and might be hard to work. Here the softmax is extremely helpful because it changes over the scores to standardised probabilities

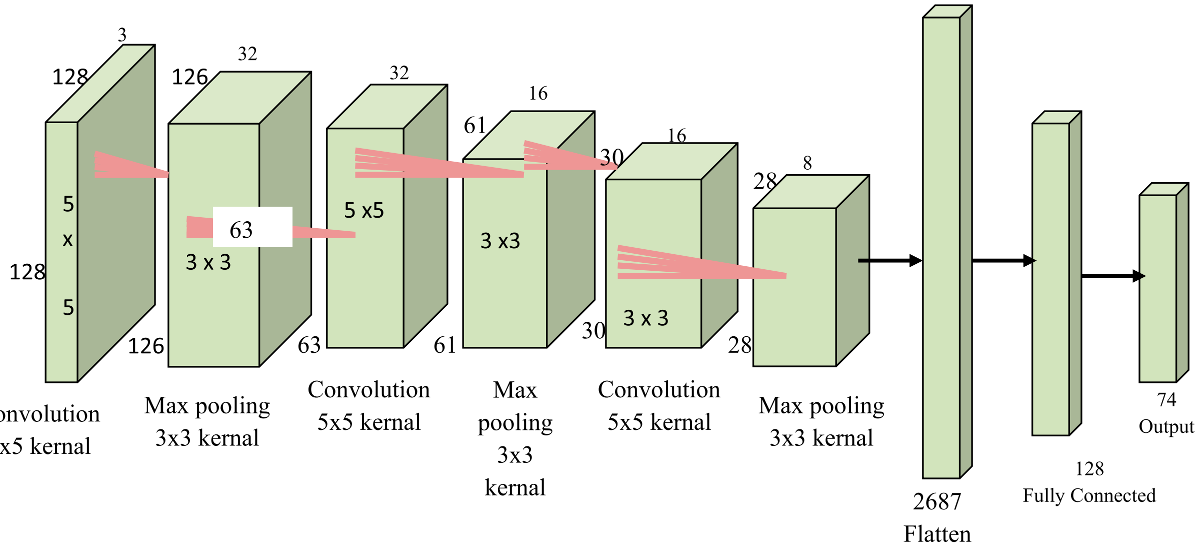

Figure 3. Representation of the convolution network.

Fully connected layer

The feature map was separated and diminished in light of the convolution and max pool layers in the information pictures. The final output depends on the classes obtained by the completely associated layer. The Resnet, the first layer from the given image was segmented with the resolution of 236 × 236 × 3 with the size of 12 × 12 for 110 parts, in which 236 × 236 is the width and height and 3 is the depth, which RGB individually. The element map obtained from the convolutional layer is passed to the following layer which further diminished the element guide and went to the following convolutional layer. Finally, it arrives at the completely associated layer. The fourth and fifth layers have more kernels. In the fully connected layer, the number of neurons is more than 4000. It is interconnected with each layer in the network and it also had an activation function in all previous layers.

Training and testing

In this training and testing part, the layers, which represent the network, trained the images. The activation function plays a vital role in the training of the data; the accurate classification of cancer is done only in the new layer obtained in the network. We take 70% of the trained model in the dataset with training parameters, such as epochs, learning rate and batches. We prepare the model with the clusters of 1–100 with 1000 epochs.

A thick layer of 512 units was used for the Flatten layer the ReLU, initiation restricted half of the dropout. Its last yield layer utilise 5 units of SoftMax '(2)' and Sigmoid '(3)'.

$$\sigma\,(Z) = \frac{\sigma^{zj}}{\sum_{k=1}^{k} e^{zk}} \quad for\ j = 1\ ..\ k \tag{1}$$

$$\phi(z) = \frac{1}{1 + e^{z}} \tag{2}$$

$$v_t = (1 - \beta_2)\sum_{i=1}^{t} \beta_2^{t-1}.g_i^2 \tag{3}$$

In the test section, 103 photos in Bacterial Location, 90 photos in Late Blight, 116 photos in Septorial Infected Plants, 128 photos in Tomato Mosaic with 142, 120 photos in Yellow Curved and solid separately were considered separately. In the preparation of the information section, 378 photos in Bacterial Location, 339 photos in Late Blight, around 446 photos in Septorial Leaf Disease, 328 photos in Tomato Mosaic and 562 in Yellow Curved and 397 photos for the sound class were considered (https://www.kaggle.com/thanjaivadivelm/tomato).

In short, its design has an input image size of 224 × 224. Channel size is 5×5 and padding is done to keep the arrangement of transitional yields the same. It has 13 convolution layers and 3 dense layers. Actuation work utilised is PDP9 layers. Two penultimate layers have 5096 hidden hubs each and the last layer has 1000 yield hubs equivalent to the number of classes (Kumar et al. 2020, July). The VGG16 pre-prepared loads were stacked and a yield layer of 16 measurements, relating to 74 classes of tomato, was added. It was found that the move picking up utilising VGG16 was not hopeful in this model drove dataset of tomato leaves from Plant Village dataset organisers. In Table 1, the exactness of 99% was obtained when the code for 1000 epochs was prepared.

The V3 is a 42-layer learning network with not many parameters. The deduction in boundaries is finished with the assistance of authorising convolutions. For instance, a 5 × 5 channel convolution should be possible by two 3 × 3 channel convolutions. The boundary in this cycle decreases from 5 × 5 = 25–3 × 3 + 3 × 3 = 18. In this way, it acquires 38% reduced number of boundaries. With fewer boundaries, the models will fit less and, in this manner, increase the exactness. The Inception V3 was used in the pre-trained model to perform move learning on the 16-class tomato disease. The precision was under VGG16 and came to 99.5% when it was prepared for 1000 leaves. The model functioned admirably on a more noteworthy number of classes as 42-layer design makes it overfit for a modest number of classes, when the qualification highlights are not clear or huge.

Algorithm for the Fast Enhanced Learning method

"**Step 1:** Set Initial value of λ(step size)

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cts}}\sum_i L_{cts}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{4}$$

λ is 10, N_{cts} is 256, and N_{reg} is 2400. With this location, the two divisions of the region proposal network (RPN) for object proposal generation and achieve further speed-up $L_{reg} = L_{loc}$. (Input, N_{cts}, and desired data, L_{cts}), (p_i

Table 1. The test after-effect of every classification (adjusting).

Category	Total test number	Correct number	Accuracy (%)
Healthy	100	99	99
Tomato_bacterial_spot	100	99	99
Tomato_early_blight	100	100	100
Tomato_late_blight	100	98	98
Tomato_leaf_mold	100	100	100
Tomato_Septoria_leaf_spot	100	100	100
Tomato_two_spotted_spider_mite	100	99	99
Tomato_target_spot	100	100	100
Tomato_mosaic_virus	100	100	100
Yellow_leaf_curl_ virus	100	99	99
Total	1000	99.5	99.5

and t_i) – threshold error.

$$L_{loc}(t^u, \nu) = \Sigma_{ie(k,y,w,h)} Smooth\ Li(ti^u - \nu i) \tag{5}$$

$$smooth_{L_1}(x) = \left\{ \begin{array}{ll} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & otherwise \end{array} \right\} \tag{6}$$

Step 2: All in all, the deficiency of Faster CNN is separated into two enormous squares. The main square is the deficiency of preparing RPN and the subsequent square is the deficiency of the classifier in preparing quick CNN. The total misfortune capacity of quicker CNN is portrayed as follows:

$$L_{final} = L(\{p_i\}, \{t_i\}) + L(p, u, t^u, v) \tag{7}$$

With the Mask CNN, a new branch is added for predicting segmentation masks on each Region of Interest (RoI) in real-time, pixel by pixel, on each Region of Interest (RoI). When it comes to pixel-to-pixel alignment between network inputs and outputs, Faster CNN is not intended for it. The contrast between Mask CNN and Faster CNN is the red piece in Figure 4. It is visible that the overall organisation structure is the equivalent. There is a 'head' part after RoI Align in Mask CNN Return on Investment Alignment (ROI Align), which does not digitalise the cell boundaries (top right) and makes every target cell the same size (bottom right). It also employs interpolation to improve the accuracy of the feature map values within the cell.

Step 3: The principal work is to grow the yield measurement of RoI Align, which is more exact when foreseeing Mask. In the instructional meeting of Mask Branch, rather than utilising FPN-style Softmax misfortune, K Mask expectation diagrams are yielded and the normally paired cross-entropy misfortune preparation is utilised obviously in preparing Mask branch. Among the K components map yield, just one of the element maps relating to the ground truth classification adds to Mask misfortune. The preparation misfortune capacity of Mask CNN can be portrayed as

$$L_{final} = L(\{p_i\}, \{t_i\}) + (L_{cls} + L_{box} + L_{mask}) \tag{8}$$

Step 4: 'The info in Figure 4 is gone through a pile of convolution layers. The contribution to the convolution is a fixed size 224 × 224 RGB picture which is pre-processed to take away the mean RGB esteem registered on the preparation set. The channels have a 3*3 open field, and the convolution step is 1 pixel; the spatial cushioning is 1 pixel to protect the spatial goal after convolution. Spatial pooling is completed by five max-pooling layers over a 2 × 2 pixel window. A heap of convolution layers is trailed by three Fully-Connected (FC) layers: the first two have 4096 channels each, the third

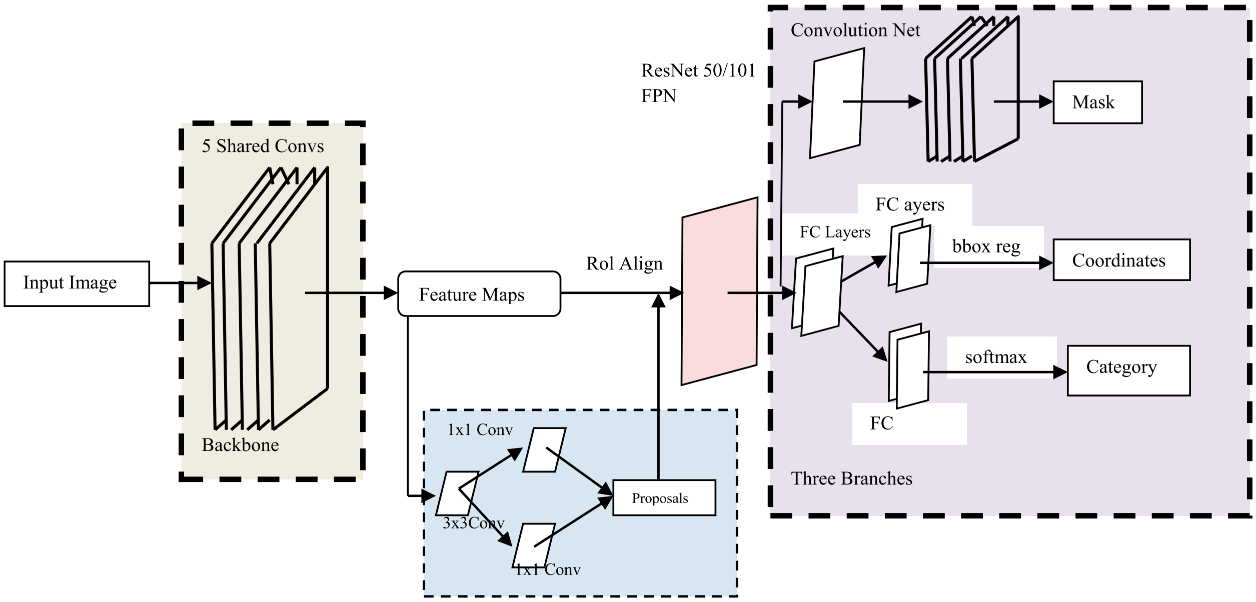

Figure 4. CNN classification network architecture.

Table 2. Experimental design platform.

Operating system	Windows 10
Core	GPU NVIDIA GEFORCE GTX
Language	Python
Libraries	OS, Tensorflow, Keras, OpenCv, Image Datagenerator, Matplot, numpy.
Dataset	Plant village dataset
Platform	Kaggle (https://www.kaggle.com/thanjaivadivelm)

performs 1000-way ILSVRC grouping and accordingly contains 1000 channels (one for each class). All shrouded layers are furnished with the correction (ReLU) non-linearity. In this plan, the yields of the penultimate completely associated layer are separated as the component descriptors. These descriptors of preparing images and their comparing marks are utilised to prepare CNN classifier.

Step 5: The number of features is evaluated by the minimum value of its optimisation problem. This is because when the loss function reaches a different quality, iteration is stopped.

Step 6: Learning inaccuracy quantified by all layers.

Step 7: These processes are repeated several times once the defect value is obtained.

$$t_i = \frac{1}{n}\sum_{i=1}^{n} \max(0.1 - y_i(\omega^* x_i + b)) + \lambda\omega^2 \tag{9}$$

where x_i is the i^{th} sample in the training dataset (t_i) and y_i point to the class label of the point x_i.

Proposed experimental design

In Table 2, all the implementations were executed in python under Windows 10, using the GPU NVIDIA GEFORCE GTX. Then, some of the experiments were executed using Kaggle website (https://www.kaggle.com/thanjaivadivelm/plant-disease-ml-classifer). The dataset contains 10,000 images at a resolution of 256 × 256. Before feeding into the neural network, we scaled the values to 0–1 range. To train the model we have used adam optimiser, categorical_crossentropy used for loss function. The loss function measures the accuracy during the training. To calculate metrics, we used both accuracy and error.

The accuracy of different network structure is presented in Table 3. At first, the batch size and 4992 iterations were combined; the initial learning rate was 0.001 and dropped by a factor of 0.1 every 2496 iterations. To get more convincing conclusions, ResNet(16, 9984) was also used to execute the experiments, while that of the test set corrupts to 89%. The preparation cycle costs 4.5 h to merge, and the testing time is 0.12s/picture. The test is the after-effect of every classification and the disarray framework.

Using Figure 4, CNN Classification Network Architecture, we trained a model on images of plant leaves for classifying both crop disease and the presence and identity of disease on images that the model had not seen before. Within the PlantVillage dataset of 54,306 images containing 38 classes of 1 crop species and 10 diseases (includes healthy plant images), this goal has been achieved, as demonstrated by the top accuracy of 99.4%. Thus, without any feature engineering, the model correctly classifies crop and disease from 10 possible classes in 497 out of 500 images.

Result and discussion

The plant infection is only a piece of the leaf that is influenced by the sickness-causing specialists such as microbes, organisms and infections. There are different infections such as bacterial spot, late curse, tomato Mosaic, found in tomato leaves. Every tomato leaf sickness is depicted in Figure 5(a)–(c).

Table 3. Comparative analysis of leaf disease detection using different classifiers.

S. No.	Types of leaves	Feature extraction	Classification	Accuracy
1	Grape leaf	LBP	SVM	95.2%
2	Plant leaves image from Jardan	Color co-occurrence method	NN	92%
3	Rice leaf	Zone-wise feature extraction	MDC and k-NN	87.4%
4	Citrus leaf	GLCM	SVM	93%
5	Sugarcane leaf	GLCM	SVM	92%
6	Cucumber leaf	Morphological feature extraction	Minimum Distance classification	93%
7	Herbal plant leaves	Morphological feature Extraction	ANN	97.3%
8	Brinjal leaves	GLCM	ANN	Not Specified
9	Paper Mulberry, Mono Maple leaf	Haralick Texture with Gabor filter, shape, colour feature extraction	FRVM	97.6%
10	Plant leaves	CCM	NN	92%
11	Plant leaves	GLCM	SVM	95.6%
12	Plant leaves	Texture feature extraction	MDC	Not specified
13	Rice leaf	Color feature extraction	ANN	92.1%
14	Rice Leaf	Zooming Algorithm	Self-Organising	Not specified
15	Rose, Lemon, Beans, Banana Leaf	Colour Co-occurrence method	MDC < SVM	94.8%
16	**Proposed work:** Tomato leaf	GLCM, Colour feature extraction with VGG16-based models	SVM < CNN	99.5%

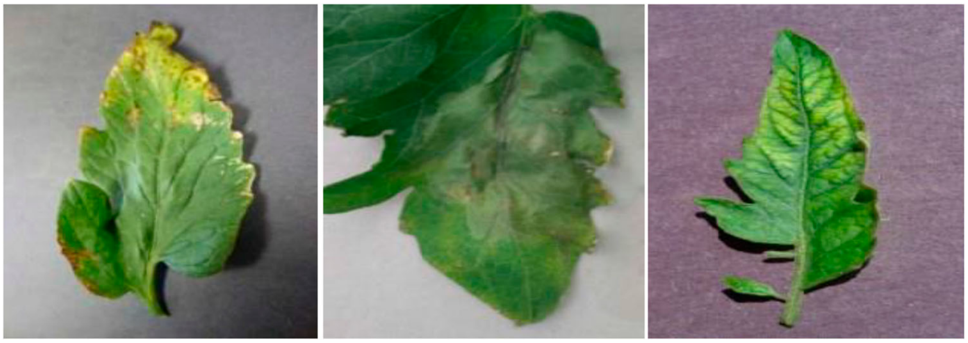

Figure 5. (a) Bacterial spot; (b) late blight; (c) tomato mosaic.

Highlight Extraction is utilised for extricating surface highlights from the sick parts of the images of the leaves. GLCM is broad technique for surface extraction from images. In this network, the sickness manifestations are maintained by getting the component estimations of disparity, relationship, homogeneity, contrast, Angular Second Moment (ASM). Differentiation quantifies pixel's power contrast and neighbour pixel's force of a picture. Homogeneity addresses the component's closeness. ASM addresses the image's routineness and consistency. Connection quantifies the direct power reliance of dim degrees of adjoining pixels in the image, and Difference is a proportion of the distance between the sets of pixels in the area of interest. Because of the qualities determined from the disease-influenced zones, the highlights of the picture for additional investigation were separated.

Irregular backwood classifier with randomised choice trees is applied here to prepare the framework to accomplish the directed AI. Table 3 represents a various comparative analysis of leaf disease detection with different Features extraction and classifier. For instance, in feature extraction, we examined texture feature extraction, colour feature extraction, Zooming Algorithm, Zoon wise feature extraction etc. In the classification method we examined SVM, ANN, KNN, etc.,(Jogekar and Tiwari 2020, July). The performance analysis of the above result is depicted in Table 4.

Every choice tree yields a decision in favour of the objective variable forecast. Both mathematical and unmitigated information are dealt with, and over fitting of the informational preparation index is maintained at a strategic distance by the irregular backwood classifier (Kumara 2020). It is utilised for both arrangement and relapse issues. Arbitrary Forest chooses the forecast that secures the most extreme vote. It is a troupe packing calculation, thus accomplishing a common forecast blunder. It runs proficiently in any event when the dataset is tremendous and the quantity of highlights is likewise generally high.

The spine organisation of YOLOv3-small is a 7-layer standard convolution structure instead of a Darknet arrangement. YOLOv3-minuscule is equivalent to the start to finish object discovery strategy. The info layer is a picture, and afterwards after ten convolutions and 6 sub-examining activities, the yield includes maps having a size. At the same time, the element map after the fifth down testing is unsampled and convolved to acquire a size of and it is exposed to standard convolution twice to get a size of. The yield includes guides of the two scales contain the expectation data of items. The quantity of FLOPs of YOLOv3-little is just 5.56 bn, and the model size is just 33.7MB. It can run on installed or cell phones. Nonetheless, its spine network has just 7 layers, so it can't separate more significant level semantic highlights, and its accuracy is low (Kuznetsova et al. 2020).

The focal minutes are invariant to scale, interpretation and turn. It is adequate to the condition of shape coordinating. Minutes are a gathering of seven numbers decided utilising focal minutes, which are invariant to picture changes. The initial 6 minutes end up being invariant to scale, interpretation, reflection and revolution. The indication of the seventh second will change with picture reflection. The conditions for 7 minutes are as follows':

$$h_0 = \eta_{20} + \eta_{02} \tag{10}$$

$$h_1 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2 \tag{11}$$

$$h_2 = (\eta_{30} + 3\eta_{12})^2 + (3\eta_{21} + \eta_{02})^2 \tag{12}$$

$$h_3 = (\eta_{30} + 3\eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{13}$$

Table 4. Performance analysis of different classifier techniques.

Prediction	Classification type/size	Filter/methods	**Techniques**	Precision	Recall	F1-Score	Support	Mean	PSNR(**dB**)	Flatten	Classification Accuracy(%)
K-Means classifier											
BP	Conv5 × 5 kernal	128/ YOLOv3, YOLOv3- tiny	**K-Means classifier**	0.875	0.828	0.823	445	84.5546	26.556	2287	81.7
HEALTHY				0.913	0.824	0.811	345	76.2361	28.234	2367	84.9
MV				0.893	0.782	0.801	412	75.801	27.367	2102	85.8
LM				0.905	0.867	0.856	389	74.901	28.896	2276	84.2
YV				0.891	0.879	0.892	423	79.923	21.423	2105	82.3
SM				0.937	0.891	0.915	467	80.263	21.564	2118	82.9
TS				0.941	0.896	0.934	512	81.321	21.786	2189	84.9
Over all				0.975	0.928	0–923	545	84.5546	23.556	2187	85.3
SVM classification											
BP	MaxPool 3 x3 kernal	126/ YOLOv3, YOLOv3-tiny	**SVM**	0.915	0.858	0.863	345	74.546	26.556	2387	85.7
Healthy				0.983	0.834	0.881	245	76.361	25.234	1989	81.9
MV				0.913	0.722	0.821	212	75.801	23.367	2202	86.8
LM				0.895	0.817	0.896	389	74.901	21.896	2196	80.2
Yellow curl				0.811	0.889	0.992	313	79.923	21.423	2305	86.3
SM				0.977	0.861	0.985	387	80.263	24.564	2218	85.9
TS				0.987	0.898	0.994	199	80.321	22.786	2189	87.9
Overall				0.967	0.978	0.915	234	76.765	26.876	2143	89
RBF Kernel											
BP	Conv5 × 5 kernal	63/ YOLOv3, YOLOv3-tiny	**RBF Kernel**	0.895	0.722	0.893	395	76.446	23.358	2487	84.7
Healthy				0.899	0.917	0.829	385	74.261	21.284	2289	82.9
MV				0.923	0.762	0.891	312	75.901	24.367	2802	83.8
LM				0.875	0. 834	0.896	319	74.201	22.496	2496	89.2
Yellow curl				0.871	0.798	0.992	393	78.223	22.523	2655	85.3
SM				0.917	0.891	0.995	327	80.363	23.164	2768	84.9
TS				0.937	0.918	0.945	299	80.121	21.686	2849	87.9
Overall				0.924	0.968	0.956	431	79.234	21.453	2345	88.8
Optimised MLP											
BP	MaxPool 3 x3 kernal	61/ YOLOv3, YOLOv3-tiny	**Optimised MLP**	0.815	0.882	0.929	325	83.446	22.658	2587	87.7
Healthy				0. 895	0.867	0.879	285	78.261	24.474	2389	90.9
MV				0.819	0.822	0.931	312	89.901	21.257	2702	85.8
LM				0.893	0. 814	0.876	399	78.201	21.386	2196	89.8
Yellow curl				0.921	0.888	0.822	273	75.223	24.413	2955	84.7
SM				0.887	0.836	0.915	277	78.363	24.454	2368	85.9
TS				0.927	0.918	0.978	374	81.321	23.636	2149	90.2
Overall				0.942	0.967	0.928	345	88.761	25.865	2281	91.4
NN classifier											
BP	Conv5 × 5 kernal	30/ YOLOv3, YOLOv3-tiny	**NN classifier**	0.825	0.832	0.969	225	73.446	21.658	2483	91.7
Healthy				0. 935	0.998	0.905	315	85.261	21.574	2269	95.9
MV				0.869	0.852	0.932	292	71.901	22.957	2652	92.8
LM				0.893	0. 844	0.916	369	78.201	23.786	2276	89.6
Yellow curl				0.821	0.985	0.892	373	75.223	24.513	2165	87.7
SM				0.847	0.876	0.915	297	69.363	24.854	2108	88.9
TS				0.977	0.928	0.958	294	75.321	23.936	2059	94.8
Overall				0.951	0.975	0.952	213	78.677	23.567	2312	97.3
BPNN											
BP	MaxPool 3 x3 kernal	28/ YOLOv3, YOLOv3-tiny	**BPNN**	0.925	0.932	0.949	355	83.446	24.658	2382	81.7
Healthy				0. 975	0.878	0.975	415	89.261	26.974	2469	83.9
MV				0.929	0.952	0.952	392	84.901	25.957	2452	82.8
LM				0.913	0. 944	0.946	403	88.201	24.836	2376	84.6
Yellow curl				0.898	0.985	0.899	383	90.223	24.893	2285	81.7

SM				0.893	0.896	0.935	407	89.363	23.954	2388	79.9
TS				0.987	0.948	0.972	394	89.521	24.986	2459	83.8
Overall				0.961	0.972	0.984	432	91.345	27.976	2498	85.5
CNN classifier											
BP	Conv 5 × 5 kernal	256/ YOLOv3, YOLOv3-tiny	**CNN classifier**	0.935	0.922	0.919	375	93.446	26.358	2482	80.7
Healthy				0. 970	0.8978	0.905	378	96.261	27.474	2269	81.9
MV				0.959	0.922	0.902	397	94.901	24.657	2152	83.8
LM				0.923	0. 974	0.916	396	89.201	27.536	2476	82.6
Yellow curl				0.918	0.988	0.909	383	92.223	26.693	2285	82.7
SM				0.923	0.899	0.835	402	95.363	25.554	2358	86.9
TS				0.917	0.958	0.872	396	93.521	24.386	2499	85.8
Overall				0.975	0.981	0.934	412	98.145	28.787	2532	89.2
Our method											
BP	Conv 5 × 5 kernel	1024/ Fast Enhanced Learning Method		0.955	0.932	0.919	495	96.646	27.658	2495	90.8
Healthy				0. 985	0.984	0.979	618	100.961	29.674	2619	98.9
MV				0.949	0.945	0.962	467	97.591	27.857	2462	95.3
LM				0.932	0. 934	0.956	516	95.341	26.936	2476	95.5
Yellow curl				0.938	0.968	0.979	599	98.323	28.493	2595	96.4
SM				0.953	0.919	0.899	512	99.463	26.854	2528	97.2
TS				0.967	0.978	0.898	536	99.521	25.986	2539	95.5
Overall				0.995	0.995	0.988	0.689	191.453	30.321	2687	99.5

$$h_4 = (\eta_{30} + 3\eta_{12})(\eta_{30} + 3\eta_{12})^2(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 + (3\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{30})^2] \quad (14)$$

$$h_5 = (\eta_{20} + \eta_{02})(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{02})^2 + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (15)$$

$$h_6 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (16)$$

It has two stages: GLCM calculation and figuring of surface highlights dependent on GLCM. GLCM is a square lattice having a measurement N_g, where N_g compares to the quantity of dark levels (Farhan and Kamil 2020, November; Anggraini et al. 2020).

Fourteen factual terms are resolved utilising co-event framework to characterise the surface, which are as follows:

$$\text{Angular Second Moment} = \sum_i \sum_j p(i,j)^2 \quad (17)$$

$$\text{Contrast} = \sum_{n=0}^{N_g-1} n2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \right\}, |i - j| = n \quad (18)$$

$$\text{Correlation} = \frac{\sum_i \sum_j (ij)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (19)$$

$$\text{Variance} = \sum_i \sum_j (i - \mu)^2 p(i,j) \quad (20)$$

$$\text{Inverse Difference Method} = \sum_i \sum_j \frac{1}{1 + (i-j)^2} p(i,j) \quad (21)$$

$$\text{Sum Average} = \sum_{i=2}^{2N_g} i p_{x+y}(i) \quad (22)$$

$$\text{Sum Variance} = \sum_{i=2}^{2N_g} (i - f_s)^{2p_{x+y}}(i) \quad (23)$$

$$\text{Sum Entropy} = \sum_{i=2}^{2N_g} p_{x+y}(i) \log\{p_{x+y}(i)\} = f_s \quad (24)$$

$$\text{Entropy} = \sum_i \sum_j p(i,j) \log(p(i,j)) \quad (25)$$

$$\text{Difference Variance} = \sum_{i=0}^{N_g-1} i^2 p_{x-y}(i) \quad (26)$$

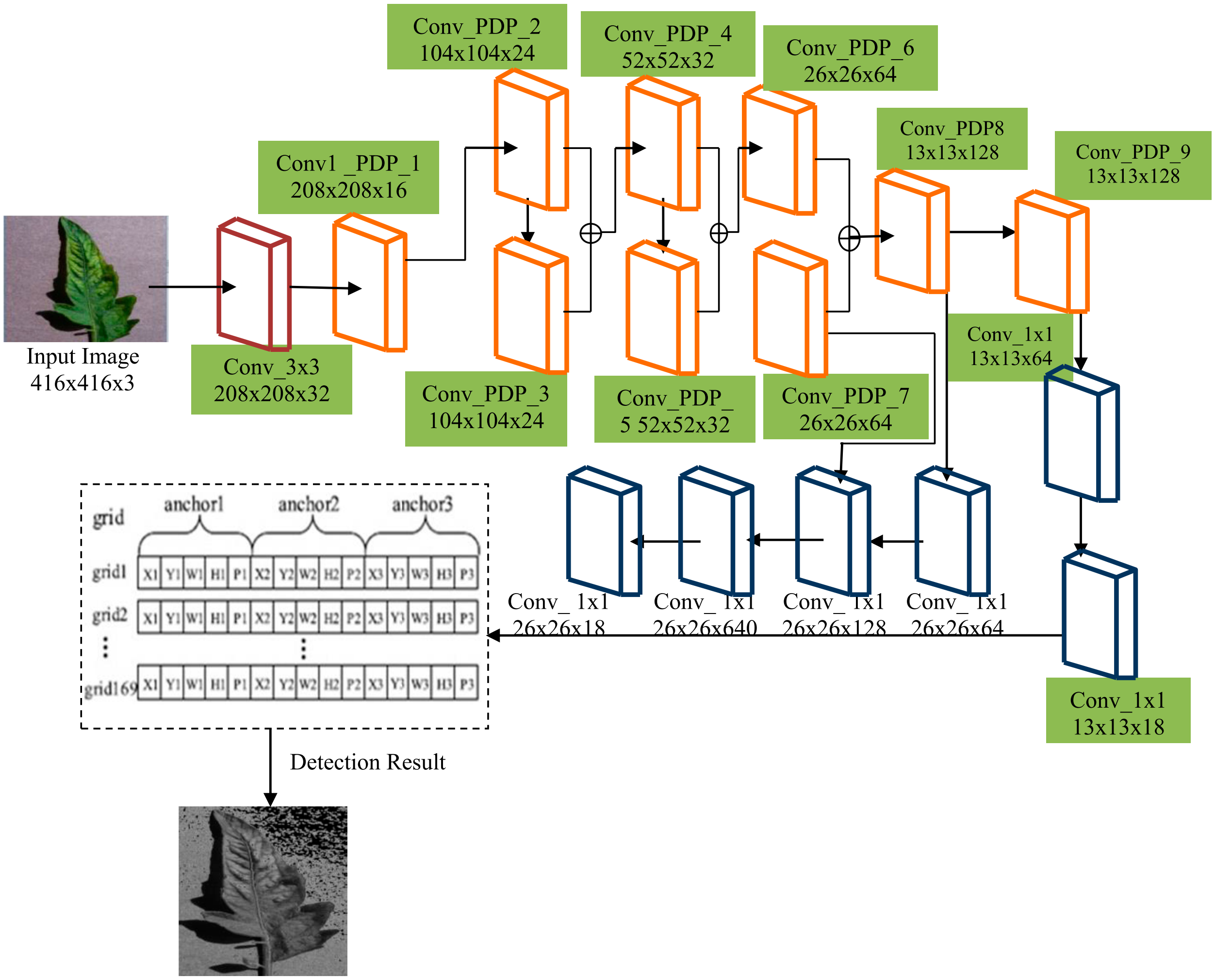

Figure 6. Evaluation of classification of tomato leaves and finding the disease.

$$\text{Difference Entropy} = \sum_{i=0}^{N_g-1} p_{x-y}(i)\log\{p_{x-y}(i)\} \quad (27)$$

$$\text{Info.Measure of Correlation 1} = \frac{HXY - HXY1}{\max\{HX, HY\}} \quad (28)$$

$$\text{Info.Measure of Correlation 2} = (1 - \exp[-2(\mathrm{HXY2} - \mathrm{HXY})])^{1/2} \quad (29)$$

P_{ij} = Element i,j of the normalised symmetrical GLCM

N = Number of grey levels in the image as specified by the number of levels under quantisation on the GLCM texture page of the Variable Properties dialog box.

μ = the GLCM mean

σ = the variance of the intensities of all reference pixels in the relationships that contributed to the GLCM, calculated as

C = The Correlation feature

$\mathrm{sgn}(x)$ = Sign of a real number

$$\text{Precision} = \mathrm{tp}\ /(\mathrm{tp} + \mathrm{fp}) \quad (30)$$

where as tp – true-positive and fp – false-positive

$$\text{Recall} = \mathrm{tp}\ /(\mathrm{tp} + \mathrm{fn}) \quad (31)$$

where fn is false-negative and tn is true-negative.

Support - number of images with that target label

$$\text{Accuracy} = \mathrm{tp} + \mathrm{tn}\ /\ \mathrm{tp} + \mathrm{tn} + \mathrm{fp} + \mathrm{fn} \quad (32)$$

$$\text{PSNR} = 10\log10(\mathrm{pval2/MSE}) \quad (33)$$

where pval is a pixel value of the image.

$$MSE = \sqrt{\frac{\sum_{i=1}^{n}(Y' - Y)^2}{n}} \quad (34)$$

The mean squared error lies between two images Y' and Y.

Table 5. Overall comparison analysis for the proposed method.

Overall classification test accuracy (%)								
Prediction of leaf	K-Means classifier	SVM	RBF Kernel	Optimised MLP	NN classifier	BPNN	CNN classifier	Our methods
BP	81.7	85.7	84.7	87.7	91.7	81.7	80.7	90.8
Healthy	84.9	81.9	82.9	90.9	95.9	83.9	81.9	98.9
MV	85.8	86.9	83.8	85.8	92.8	82.8	83.8	95.3
Septoria spot	84.2	80.2	89.2	89.8	89.6	84.6	82.6	95.5
Yellow curl	82.3	86.3	85.3	84.7	87.7	81.7	82.1	96.4
SM	82.9	85.9	84.9	85.9	88.9	79.9	86.9	97.2
TS	84.9	87.9	87.9	90.2	94.8	83.8	85.8	95.5
Overall	85.3	89	88.8	91.4	97.3	85.5	89.2	99.5

The significant advance in the proposed procedure is the preparation cycle. Figure 6 shows the flowchart of preparing measures. The images are resized to a goal of 208 × 208 × 32. This is because to keep up consistency in extricated highlights. The subsequent stage in the wake of resizing the picture includes extraction. Hu minutes, haralick, shading histogram and LBP highlights are separated, linked and put away in a number configuration. After all the highlights are separated from the class envelope, the class organiser is increased by 1 till the highlights are removed from all the five envelopes (Lohithashva et al. 2020).

The arrangement exactness of the proposed strategy is contrasted with other cutting-edge methods. Table 5 shows the examination table of strategies utilised and characterisation exactness of various techniques. It also discussed other leaf disease detections using ML algorithms. Labels used in the performance analysis ['Bacterial_spot[BP]', 'Early_blight[EB]', 'Late_blight[LB]', 'Leaf_Mold[LM]', 'Septoria_leaf_spot[SLS]', 'Spider_mitesTwo-spotted_spider_mite[SM]', 'Target_Spot[TS]', 'Tomato_Yellow_Leaf_Curl_Virus[YV]', 'Tomato_mosaic_virus]MV]', 'healthy']. The techniques used in this paper are K-means, SVM, RBF Kernel, Optimised MLP, etc. Table 5 represents the summary of the Performance Analysis of different classifier Techniques with the proposed method. The parameters used to measure are precision, Recall, F1 Score, Support, Mean, PSNR, Flatten.

Here dataset has been separated into two categories: 'training' and 'validation'. For training, ten classes are made. They are 'Bacterial spot', 'Healthy', 'Mosaic', 'Septoria spot ', 'Yellow curl', etc., Ten thousand images were taken for training each class of 1000 images. For validating, 1000 images were taken, each type has 100 images. After validating, some of the images are tested using random images available from the internet source. The images are collected from the source (https://extension.umn.edu/diseases/late-blight; https://extension.umn.edu/diseases/early-blight-tomato).

Conclusion

This paper mainly considers the programmed location of tomato disease depending on the leaf. The identification models are prepared to identify the tomato sicknesses and bugs using learning innovation, which accomplishes a standard characterisation accuracy of 99.49%. Nonetheless, the general choice depends on relative top indentation test images (i.e. straightforward foundation, object-focused, positive close-up shooting). Future examination will zero in on the confound calculations to distinguish tomato vermin and sicknesses, depending on inferior quality leaf images. A CNN-based model is used to recognise the infection in the tomato crop. In the proposed CNN-based engineering, there are three convolution and top pooling layers with differing numbers of channels in each layer. For the analysis, the tomato leaf information from the Plant Village dataset is obtained. In the dataset, there are 16 infection classes, and the classes have solid images. Information growth procedures are applied to adjust the images inside the class. Also, the standard testing exactness of the model is 99.5%. As future work, attempts to change the model with more significant of images using another harvest will be worked. Besides, studies to improve a similar model on the same dataset as testing exactness are also considered.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

M. Thanjai Vadivel received his B.E. degree in Computer Science and Engineering from Anna University, Chennai, in 2008. He received his Masters degree in IT (Networking) from VIT University, Vellore, Chennai, India, in 2012. He is a research scholar and assistant professor in the Department of Computer Science and Engineering at Veltech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India. His areas of interest are network security, image processing, IoT, wireless sensor networks, cloud security and Data Mining.

R. Suguna completed B.E. in Computer Engineering at Thiagarajar College of Engineering, Madurai, and M.Tech in Computer Science and Engineering at IIT Madras in 1989 and 2004, respectively. She received a doctorate from Anna University in 2011. Her research interests include Image Processing, Data Mining and Machine Learning. She has 27 years of experience in teaching and held various positions in institutions. She has organised and chaired many national/international conferences and published papers in reputed journals. She is an active member of CSI and IEEE.

ORCID

Thanjai Vadivel http://orcid.org/0000-0002-8754-1947

References

Anggraini RA, Wati FF, Shidiq MJF, Suryadi A, Fatah H, Kholifah D. 2020. Identification of herbal plant based on leaf image using glcm feature and k-means. J Tech Nusa Mandiri. 17(1):71–78.

Arcidiacono C. 2020. Webinars cycle < sustainable and environmentally safe plant health care in a technologically advanced agriculture > to celebrate IYPH 2020 proclaimed by FAO WEBINAR data sharing and other open [Doctoral dissertation, Sainsbury Laboratory].

Argüeso D, Picon A, Irusta U, Medela A, San-Emeterio MG, Bereciartua A, Alvarez-Gila A. 2020. Few-shot learning approach for plant disease classification using images taken in the field. Comput Electron Agric. 175:105542.

Farhan AH, Kamil MY. 2020, November. Texture analysis of breast cancer via LBP, HOG, and GLCM techniques. In IOP conference series: materials science and engineering (Vol. 928, No. 7, p. 072098). IOP Publishing.

https://extension.umn.edu/diseases/early-blight-tomato.

https://extension.umn.edu/diseases/late-blight.

https://www.kaggle.com/abdallahalidev/plantvillage-dataset.

https://www.kaggle.com/thanjaivadivelm/tomato.

Jogekar R, Tiwari N. 2020, July. Summary of leaf-based plant disease detection systems: A compilation of systematic study findings to classify the leaf disease classification schemes. In 2020 fourth world conference on smart trends in systems, security and sustainability (WorldS4) (pp. 745–750). IEEE.

Kumar A, Razi R, Singh A, Das H. 2020, July. Res-VGG: A novel model for plant disease detection by fusing VGG16 and ResNet models. In international conference on machine learning, image processing, network security and data sciences (pp. 383–400). Springer, Singapore.

Kumara S. 2020. Identification of nitrogen status in *Brassica juncea* l. Using color moment glcm and backpropagation neural network.

Kuznetsova A, Maleva T, Soloviev V. 2020. Using YOLOv3 algorithm with pre-and post-processing for apple detection in fruit-harvesting robot. Agronomy. 10(7):1016.

Lohithashva BH, Aradhya VM, Guru DS. 2020. Violent video event detection based on integrated LBP and GLCM texture features. Rev D'Intell Artif. 34(2):179–187.

Nazki H, Yoon S, Fuentes A, Park DS. 2020. Unsupervised image translation using adversarial networks for improved plant disease recognition. Comput Electron Agric. 168:105117.

Pham TN, Van Tran L, Dao SVT. 2020. Early disease classification of mango leaves using feed-forward neural network and hybrid metaheuristic feature selection. IEEE Access. 8:189960–189973.

Singh RK, Singh A, Zander KK, Mathew S, Kumar A. 2021. Measuring successful processes of knowledge co-production for managing climate change and associated environmental stressors: adaptation policies and practices to support Indian farmers. J Environ Manag. 282:1–14. [111679]. https://doi.org/10.1016/j.jenvman.2020.111679.

Sudgen F, Agarwal B, Leder S, Saikia P, Raut M, Kumar A, Ray D. 2021. Experiments in farmers' collectives in eastern India and Nepal: process, benefits, and challenges. J Agrarian Change. 21(1):90–121.

Xie X, Ma Y, Liu B, He J, Li S, Wang H. 2020. A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. Front Plant Sci. 11:751.