SYSTEM FOR DETECTING TOMATO PLANT DISEASES USING IMAGE PROCESSING

[1]Suresh D and [2]Dr. Sree Kala T

[1]Research Scholar, Department of Computer Science, School of Computing Sciences,
Vels Institute of Science, Technology and Advanced Studies (VISTAS), Chennai, Tamilnadu
600117 –India Email: dsureshsuse@gmail.com
[2]Associate Professor, Department of Computer Science, School of Computing Sciences,
Vels Institute of Science, Technology and Advanced Studies (VISTAS), Chennai, Tamilnadu
600117- India Email: sreekalatm@gmail.com

ABSTRACT

Plant infections are one of the biggest issues in the agricultural business. Viral, bacterial, fungal, and other causes can all contribute to plant diseases. The majority of farmers are ignorant of these illnesses. Since these illnesses can harm not only the plants themselves but also the farmers and the entire agricultural ecology, it is crucial to identify them as soon as possible. Regarding these practical concerns, the goal of this research was to automatically classify and identify plant diseases, namely those that affect tomato plants. Raspberry Pi is the primary computing device according to the hardware requirements.

The main project process is image processing, which comprises feature extraction, picture ROI adjustment, feature acquisition, and CNN-based classification. Here, the raw input image is modified using the Python programming language and the OPENCV module. Image data is gathered from the trusted web source in order to train the CNN architecture and develop a machine learning model that can identify the types of diseases. As a result, a small number of diseases that typically affect tomato plants are discovered, including Late blight (training 100, test 22), Gray spot (training 96, test 19), and Bacterial Canker (training 90, test 23).

Keywords: Convolution Neural Network (CNN), YOLO, Raspberry-Pi, and image processing.

1. INTRODUCTION

The Indian people depend heavily on agriculture for their livelihood. However, two-thirds of the population is either directly or indirectly dependent on agriculture. The majority of Indian people depend on agriculture for their livelihoods, and in the fiscal year 2015–16, the agricultural sector alone generated around 32.6% of the country's GDP. The average economic growth in 2014–15 was limited to 0.77%. One of the main cash crops grown in Indian is the tomato. Tomatoes are grown all year round on an area of 25.49 hectors, with a harvesting rate of about 13419 KG/Hac [1]. The existing technique relies on visual inspection, which takes time, to identify diseases in these plants.

Image processing has been used in agriculture to sort fresh produce, grade it, find flaws like dark spots, cracks, and bruises on fresh fruits and seeds, etc. Deep Convolution Neural Networks (CNN) have developed recently thanks to advancements in hardware technology, and there are many uses for them. Smart phone-based applications for identifying shapes and diseases in plant leaves have been created, in addition to harder tasks like object recognition and image classification.

2. LITERATURE REVIEW

For the lesion image segmentation and image recognition of alfalfa leaf disease, Qin et al. developed a workable technique. A total of 129 features were originally extracted using the Relief method, and the most crucial features were used to train an SVM model. The outcomes showed that picture recognition of the four illnesses affecting alfalfa leaves could be applied and achieved an average accuracy of 94.74% [2]. The categorization of tomato powdery mildew versus healthy leaves has been done utilizing these techniques employing thermal and stereo images [3].In order to identify apple pathogenic photos, Tan et al. introduced a method based on CNN. To update the CNN parameters, they used a self-adaptive momentum rule. The findings showed that the proposal's recognition accuracy was up to 96.08%, with a rather quick convergence [4].

A unique convolutional neural network-based system for detecting cucumber leaf disease was described. The suggested CNN-based system has an average accuracy of 94.9% when categorizing cucumbers into two groups with prevalent diseases es and a healthy class using the fourfold cross-validation technique.

3. METHODOLOGY

I. System Overview

A general overview of the system is presented in the block diagram of the system as follows:

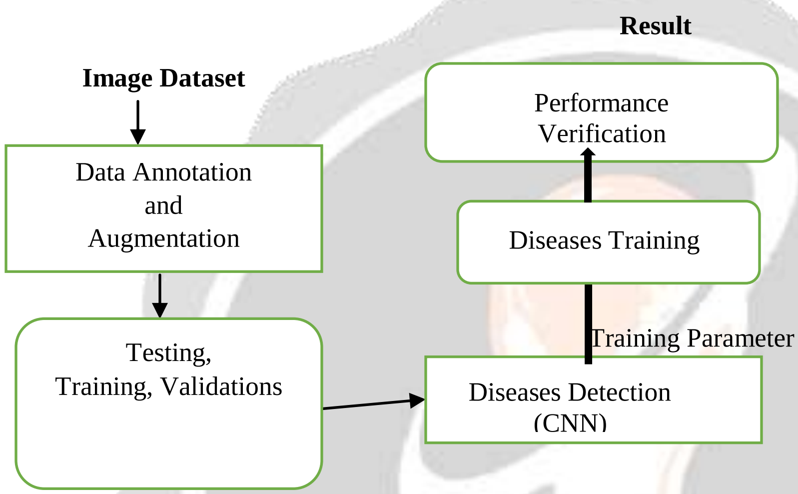

Figure 1: System overview

II Data Collection and Annotation

The collection includes pictures of many tomato plant diseases. A camera gadget was used to take some of the photographs from the farm and some were downloaded from the internet. The image was collected at different time and orientation (e.g. Illumination, different light intensity, placement, different rotation, scales).

Plant parts that are affected (such as the stem, leaves, fruits, etc.).Beginning with the picture dataset, the portions of each image bearing the disease were manually annotated with a bounding box and class. Depending on the level of infection, several diseases may appear identical. As a result, specialists in the field have given us the knowledge necessary to identify the type of disease, which has enabled us to clearly see the categories in the photos and the affected portions of the plant. The class and position of the infected spots in the image are to be identified using this annotation process. The result of this stage is the coordinates of the bounding boxes of various sizes and the class of sickness they correspond to, which will be assessed as the Intersection Over Union (IOU) with the anticipated outcomes in the network during testing.

III Designing Convolution Neural Network (CNN)

The designed network has 24 convolution layers followed by 2 fully connected layers. First, a framework based on the typical YOLO paradigm is constructed. A larger convolution kernel has a higher ability to extract the macro information from the image from the perspective of the convolution kernel, and vice versa. The image's overall size is smaller than a Layer grid. Additionally, additional image information can be considered "noise" that needs to be filtered. The first convolution layer is therefore intended to be 64 kernels of size 7*7*2.

Type	Filter	Size/Stride	Output
Convolutional	32	3*3	224*224
Maxpool		2*2/2	112*112
Convolutional	64	3*3	112*112
Maxpool		2*2/2	56*56
Convolutional	128	3*3	56*56
Convolutional	64	1*1	56*56
Convolutional	128	3*3	56*56
Maxpool		2*2/2	28*28

Table 1 CNN Network Filter

IV Training Model

The input image is taken from the internet and tagged to identify the area's regions. To improve accuracy and reduce false identification, an object with a distinct background, light intensity, orientation, and form and size is gathered. 2000 iterations of the 4 class classification dataset were used to train this model. Gradient descent using an initial learning rate of 0.1, weight decay of 0.0005, momentum of 0.9, and polynomial rate decay with a power of 4. The input image is scaled down from its initial 580*580 resolution to 448*448 over the course of 10 epochs at a 103 learning rate. The classifier obtains top-1 accuracy of 76.5% and top-5 accuracy of 93.3% after training.

Following the removal of the fully connected layers, Classifier is capable of taking photos of various sizes. We simply create four times as many output grid cells and four times as many predictions if the width and height are both doubled. We only need to make sure that the width and height are multiples of 32 because the CNN network down samples the input by 32. Classifier uses images with sizes of 320x320, 352x352, and 608x608 (with a step of 32) during training. The Classifier randomly chooses a different image size to train the model for every 10 batches. As a result, the network is forced to forecast accurately for various input image dimensions and scales. Additionally, we can detect objects in lower resolution photos, but accuracy will suffer. On devices with minimal GPU power, this may be a good speed trade-off. Nearly as good as Fast R-CNN, the 288 288 algorithm performs at better than 90 FPS with mAP. Batch normalization significantly increases convergence while obviating the need for further regularization techniques. We enhance mAP by more than 2% by adding batch normalization to all of the convolutional layers in detection. Additionally, batch normalization aids in model regularization. Batch normalization yields 4%. The final convolution layer is followed by the completely connected layers. Three 3 3 convolutional layers with 1024 filters each are added by the detection method, and then a final 1 1 convolutional layer with 125 output channels is added. Each box forecast contains five parameters. Added by the classifier is a passthrough layer. With an initial learning rate of 103, CNN trains the network for 160 epochs, dividing it by 10 at 60 and 90 epochs. When a region of infected plant parts is included in the annotation file, it is applied to the input image. Each class trains more than 500 photographs. The sort of diseases are represented by class here. Nvidia 1080 Ti was used for training.

V Detection

The detection method creates an SS grid from the input image. Only one object is predicted by each grid cell. There is a predetermined number of border boxes predicted for each grid cell. Five components make up each boundary box: (x, y, w, h) and a box confidence score. The objectness of the box and the accuracy of the boundary box are both reflected in the confidence score. the image's width and height divided by the bounding box's width and height. To the accompanying cell's normalized value, x and y are offsets. Consequently, all of x, y, w, and h are between 0 and 1. There are n conditional class probabilities in each cell. The conditional class probability (one probability per category for each cell) is the likelihood that the identified object belongs to a specific class.

This algorithm's main idea is to construct a CNN network in order to forecast a (7, 7, 30) tensor. The spatial dimension is reduced to 77 with 1024 output channels for each location using a CNN network. 772 border box predictions are made via detection using a linear regression with two completely linked layers (see middle image below). The players with high box scores are kept for the final forecast confidence scores (greater than 0.25) as our final predictions (the right picture). The localization (where an object is placed) and categorization confidence are both measured. The border boxes in the real-world domain are not randomly chosen. Leaf mold and gray spot both have fairly comparable forms and an aspect ratio of roughly 0.41.

The initial training will be more stable if we begin with a variety of predictions that are typical for real-life items because we only need one guess to be accurate. We forecast offsets to each of the aforementioned anchor boxes rather than the boundaries of 5 random boundary boxes. The diversity of the forecasts can be preserved, and each prediction can concentrate on a particular shape if the offset.

4. RESULTS

On a portion of the illnesses dataset, such as tomato plant leaf diseases, the CNN-based classifiers are put to the test. Three tomato plant leaf diseases are included in the dataset: bacterial canker (111 samples), late blight (121 samples), and gray spot (113 samples). The used dataset has 520 photos in 3 categories after including images of a healthy tomato leaf. The dataset is subjected to the first preparation and augmentation. The photographs in the dataset are shrunk to fit within 412412 dimensions, which were chosen because

they are near to the average size of all images and are quite modest. After 10% of the images are removed as the test set, the remaining images are enhanced as the training set by adding horizontally flipped copies of the images to reduce over fitting. After that, a portion of these images is further separated as the validation set, which is then trained on Image-Net beforehand and fine-tuned on the dataset, and the proposed CNN. both architectures use residual learning. First, on the dataset, the pre-trained YOLO models are adjusted to serve as a baseline for comparison. Then, to compare the outcomes, a condensed CNN architecture is suggested and trained both with and without the residual learning framework (residual and plain CNN). All illnesses that might impede the development of tomato plants have been examined. Diseases vary in their characteristics and symptoms, and convolution neural networks (CNNs) are taught by categorizing these visual symptoms.once a training model that can identify every ailment is developed. The trained model performs well on the Pascal voc. Format test, with a mean average precision (MAP) of 0.76. On various image scales and pixel densities, the system can forecast diseases. Size, direction, and light intensity have no bearing on the final product. The accuracy of image detection will be good for high resolution images, nevertheless. The system adjusts the pixel value at this ratio and resizes the input image to 412*412 (width * height).

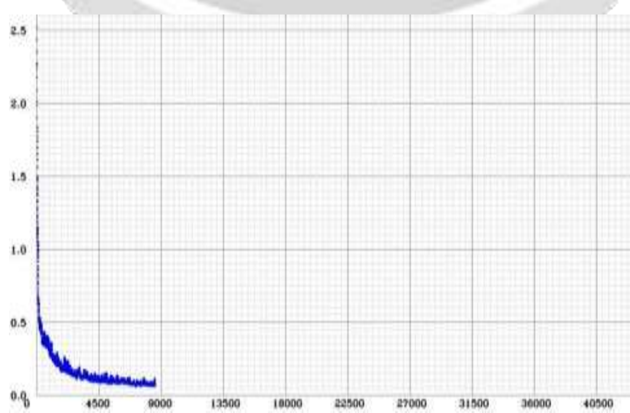

Figure 3: Detection of late blight (accuracy: 95%)

Figure 4: Detection of bacterial Canker (accuracy: 89%)

Figure 5: Detection of bacterial Canker (accuracy: 89%)

5. CONCLUSION

In this manner, by gathering information on different tomato plant illnesses and processing it to train a machine learning model using the CNN architecture, Late blight (training 100, test 21), Gray spot (training 95, test 18), and bacterial canker (training 90, test 21) are the diseases identified. To detect something a model is trained using the YOLO object identification technique built within the Darknet framework. Predict tomato plant illnesses. OPENCV library and the Python computer language are used to edit the raw input image. The Nvidia 1080 GPU is used to train the model. A desktop-based Graphical User Interface (GUI) was designed for this system, which is implemented on the Raspberry Pi.The use of technique-based data annotation and augmentation is also found to produce higher performance. This system's detection capabilities are limited to three groups of illnesses and healthy plants. Data need to be trained on the present model in order to detect different classes of diseases. The algorithm will classify different classes of diseases via transfer learning.

The key difficulty in creating an object detection model using machine learning was gathering a huge number of training photos with various background types, lighting conditions, aspect ratios, and size, shape, and size variations. According to the advice, additional research might be conducted to find all kinds of plant diseases and provide treatments for them. To apply the system in rural and remote areas, this system can be integrated with an IOT server.

6. REFERENCES

[**1**] N. Ghimire, M. Kandel, M. Aryal and D. Bhattarai, "Assessment of tomato consumption and demand in Nepal," *The Journal of Agriculture and Environment*, vol. 18, p. 83, Jun. 2017.

[**2**] F. Qin, D. Liu, B. Sun, L. Ruan, Z. Ma and H. Wang, "Identification of alfalfa leaf diseases using image," 2016.

[3] G. Prince, J.P. Clarkson, N.M. Rajpoot, “Automatic detection of diseased tomato plants using thermal and stereo visible light images,” *PloS one*, Vol. 10 No. 4, 2015, pp. e0123262.

[4] P. Rothe and R. Kshirsagar, "Cotton leaf disease identification using pattern recognition techniques," in International Conference on Pervasive Computing, Pune,India, 2015.

[5] Islam, A. Dinh, K. Wahid and P. Bhowmik, "Detection of potato diseases using image segmentation and multiclass support vector machine," Canadian Conference on Electrical and Computer Engineering, Canada, 30 April–3 May 2017.

[6] K. P. Ferentinos, "Deep Learning models for plant disease detection and diagnosis," 2018.

[7] P. Rothe and R. Kshirsagar, "Cotton leaf disease identification using pattern recognition techniques," in International Conference on Pervasive Computing, Pune,India, 2015.

[8] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk and D. Stefanovic, “Deep neural networks based recognition of plant diseases by leaf image classification,” 2016.

[9] Y. Lu, S. Yi, N. Zeng, Y. Liu and Y. Zhang, “Identification of rice diseases using deep convolutional neural networks,” 2017.

[10] S. Mohanty, D. Hughes and S. Marcel, “Using deep learning for image-based plant disease detection,” 2016.W. Tan, C. Zhao and H. Wu, “CNN intelligent early warning for apple skin lesion image acquired by infrared video sensors,” 2016.