IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.11496

Literature Review on Identifying Plants Diseases and providing supplements - using CNN model

Shailendra Singh[1], Rishab Jain[2], Rishabh Tripathi[3], Riya Goel4, Vanshika Rastogi[5]

Department of Computer Science and Engineering, Inderprastha Engineering College,

AKTU university, Uttar Pradesh, India

Abstract: Plant diseases pose a significant threat to agricultural produce and have disastrous consequences for farmers as the world's population grows. Early detection of plant disease can help ensure food security while also limiting financial losses. Images of diseased plants can aid in disease identification. Convolutional Neural Networks' classification abilities are used to generate consistent results. The simplicity of the created CNN model demonstrates its development and innovation; healthy leaves and backdrop images are consistent with previous CNN models. Using CNN, the model can distinguish between damaged and healthy leaves. Plants are the primary source of food on the planet. Plant infections and illnesses are a major risk, and the most common method of diagnosing plant diseases is to examine the plant body for visible signs and growth

[1]. Various research efforts aim to identify realistic plant protection techniques and assist our farmers as an alternative to the old time-consuming process. Technological advancements have spawned a slew of new ways to supplement old procedures in recent years

[2]. Deep learning approaches are especially effective and powerful in image classification challenges.

INTRODUCTION

Plant diseases are a major cause of agricultural production and economic losses. Correctly identifying disease is a difficult task that necessitates expertise. On the plant leaves, illnesses or their symptoms, such as coloured spots or streaks, are frequently visible. Plant diseases are typically caused by microbes such as fungi, bacteria, and viruses. There is a wide range of signs and symptoms that vary depending on the cause or aetiology of the plant disease. As examples of end-to-end learning, neural networks are an emerging application in a wide range of fields. A neural network's nodes are mathematical functions that take numerical inputs from the incoming edges and produce a numerical output as an outgoing edge. The CNN may have applications in agriculture, such as disease identification and quantification. Typically, diseases are identified by an expert using naked eye observation. This method requires a significant amount of time on large farms or land. The use of a convolutional neural network to recognise and detect plant diseases early will help to improve product quality. To create such a precise image classifier for plant disease diagnosis, we need a large, processed, and verified dataset containing images of both diseased and healthy plants. The Plant Village project has gathered thousands of plant images and made them available for free use. The dataset has already been processed and is available in three formats: coloured, gray scale and segmented.

LITERATURE SURVEY

Deep learning for image-based plant detection" [1] by Prasanna Mohanty et al. proposes a method for detecting disease in plants by training a convolutional neural network. The CNN model has been trained to identify healthy and diseased plants from 14 different species. On test set data, the model had an accuracy of 99.35 percent. When used on images obtained from trusted online sources, the model achieves an accuracy of 31.4 percent; while this is better than a simple model of random selection, a more diverse set of training data can help to increase the accuracy. Other variations of model or neural network training may also yield higher accuracy, paving the way for everyone to easily detect plant disease.

Malvika Ranjan et al. proposed an approach to detect diseases in plants using the captured image of the diseased leaf in the paper "Detection and Classification of Leaf Disease Using Artificial Neural Network." The Artificial Neural Network (ANN) is trained by selecting feature values that distinguish between diseased and healthy samples. The ANN model has an accuracy of 80%.

"Detection of unhealthy regions of plant leaves and classification of plant leaf diseases using texture features," according to the paper [3] According to S. Arivazhagan, the disease identification process consists of four major steps: The colour transformation structure is used first for the input RGB image, and then the green pixels are detected and uninvolved

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⚟ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.11496

using a specific threshold value, which is followed by the segmentation process, and texture statistics are computed to obtain beneficial segments. Finally, classifier is used to classify the disease based on the extracted features.
Kulkarni et al. in the paper ―Applying image processing technique to detect plant diseases" [4], a methodology for early and accurately plant diseases detection, using artificial neural network (ANN) and diverse image processing techniques. As the proposed approach is based on ANN classifier for classification and Gabor filter for feature extraction, it gives better results with a recognition rate of up to 91%.
Emaneul Cortes' paper "Plant disease detection using CNN and GAN" [5] proposes a method for detecting plant disease using Generative Adversarial networks. Background segmentation is used to ensure that features are extracted correctly and that output mapping is correct. Although using Gans to classify diseases in plants shows promise, segmenting based on background did not improve accuracy.

Jyotsna Bankar et al. proposed the use of the inception v3 model in classifying animals of various species in their paper Convolutional Neural Network Based Inception v3 Model for Animal Classification [6]. Inception v3 can be used to classify as well as categorise objects; this capability makes Inception v3 useful in various image classifiers.

PROPOSED METHOD

Plants are susceptible to a wide range of disease-related illnesses and convulsions. There are several reasons for this, which can be distinguished by their impact on plants, disruptions caused by environmental factors such as temperature, humidity, excess or insufficient food, light, and the most common illnesses such as bacterial, viral, and fungal infections. In the proposed system, we use the CNN algorithm to detect illness in plant leaves because it achieves the highest accuracy when the data is good.

A. Dataset

There are 39 different types of plant leaf and background photos in this collection. The collection contains 61,486 photos. Six distinct augmentation approaches were used to increase the size of the data-set. Among the techniques employed are image flipping, Gamma correction, noise injection, PCA colour augmentation, rotation, and scaling. The CNN Model forecasts 39 classes in total.

Plant	Disease Name	No. of Images
Apple	Healthy	2008
	Diseased Scab	2016
	Diseased: Black rot	1987
	Diseased: Cedar apple rust	1760
Corn	Healthy	1859
	Diseased: Cercospora leaf spot	1642
	Diseased: Common rust	1907
	Diseased: Northern Leaf Blight	1908
Grapes	Healthy	1692
	Diseased: Black rot	1888
	Diseased: Esca (Black Measles)	1920
	Diseased: Leaf blight (Isariopsis)	1722
Potato	Healthy	1824
	Diseased: Early blight	1939
	Diseased: Late blight	1939
Tomato	Healthy	1926
	Diseased: Bacterial spot	1702
	Diseased: Early blight	1920
	Diseased: Late blight	1851
	Diseased: Leaf Mold	1882
	Diseased: Septoria leaf spot	1745
	Diseased: Two-spotted spider mite	1741
	Diseased: Target Spot	1827
	Diseased: Yellow Leaf Curl Virus	1961
	Diseased: Tomato mosaic virus	1790

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.11496

Fig:2 DATASET BREAKUP (TABLE I) DATASET IMAGES

A. Performance Evaluation

We run all of our experiments across a variety of train-test set splits to get a sense of how our approaches will perform on previously unseen data and to keep track of whether any of our approaches are overfitting. These splits are as follows: 80–20 (80% of the total dataset used for training, 20% for testing), 60–40 (60% of the total dataset used for training, 40% for testing), and 50–50 (50% of the total dataset used for training, 50% for testing) (20 percent of the whole dataset used for training, and 80 percent for testing).
size = (channels, height , width)
Shape is not calculated automatically in PyTorch; we must manually take care of shape on each layer. In the First Fully Connected Layer, we must specify the output size based on the shape of the convolutional layer. Convolutional Arithmetic is another name for this calculation.

Here is Equation for Convolutional Arithmetic:

Shape:

- Input: $(N, C_{in}, H_{in}, W_{in})$
- Output: $(N, C_{out}, H_{out}, W_{out})$ where

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times \text{padding}[0] - \text{dilation}[0] \times (\text{kernel\_size}[0] - 1) - 1}{\text{stride}[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times \text{padding}[1] - \text{dilation}[1] \times (\text{kernel\_size}[1] - 1) - 1}{\text{stride}[1]} + 1 \right\rfloor$$

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.11496

Research Article

Dilation = 0, for this project.

B.Data Augmentation

• **Image Flipping -** The image can be flipped horizontally or vertically depending on the item in the image. It's a simpler way of improving the performance of our trained DL model. It also allows us to flip images from left to right and up to down.

•**Gamma Correction** - This adjusts the overall brightness of the image. Changing the Gamma Correction amount affects not only the brightness, but also the red-green-blue ratios (RGB).

•**Noise Injection -** Adding noise to a model during training can improve network resilience, resulting in better generalisation and faster learning. (In Python, use Numpy to add noise to a single value.)

- Using the 'noise' variable, generate random noise.
- Increase the noise in Dataset (Dataset = Dataset + noise).
- Separate the Noise Dataset into three parts: -

1) 70% of the time is spent on training.
2) Validation is worth 15% of the total.
3) 15% is designated for testing.

• **PCA Color Augmentation -** Adjust RGB channel intensities to match natural image variations.

• **Rotation** - A source image is rotated by a random number of degrees clockwise or anticlockwise, changing the position of an item in the frame by 0 to 360 degrees.

• **Scaling** - At random, we select a small picture size within a certain dimension range.

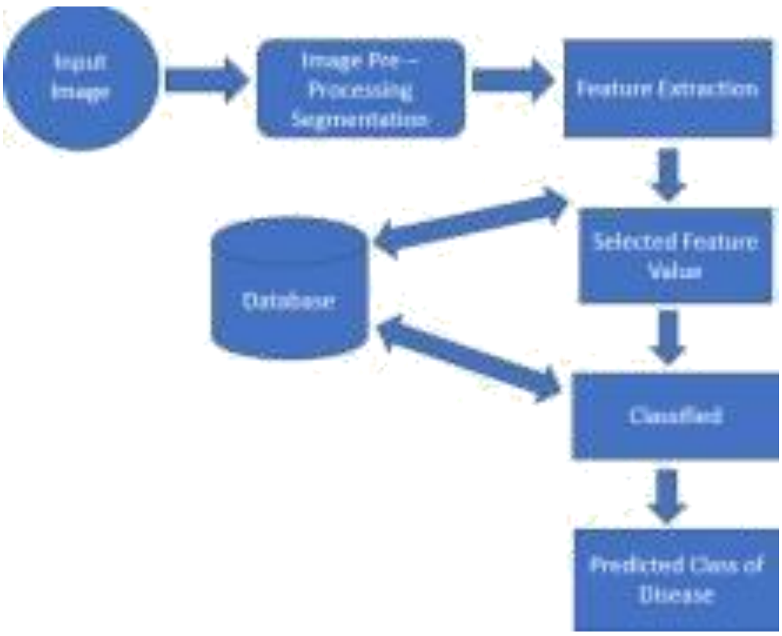
We're utilising Pytorch in our project, and we're using transformations for Data Augmentation, which acts as a filter for all photos.

C. Image Acquisition

This is the method of acquiring photos with a camera by visiting the location or using other readily available resources such as image databases or internet repositories. The images are captured in three colours: red, green, and blue (RGB), for which a colour transformation structure and a device independent colour space transformation are developed.

Image Pre-processing

We should start with indices and then divide the data into train, test, and validation data. There will be 36584 images for training, 15679 images for validation, and the remaining images for training. Then we create an object of Subset Random sampler, which is commonly used to sample our data, and we use this sampler to train and test the data loader.



IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.11496

SYSTEM OVERVIEW

Feature Selection

In all machine learning problems, feature selection is a critical step. We use a convolutional neural network to create models. The size of the fiter of the conv layer and pool layer, as well as the shape of each layer, are specified in this document. In the model, we use ReLU as the activation function to remove nonlinearity, followed by Batch Normalization to normalise the neuron weights, but SoftMax activation is required for the final layer. We have a cross-entropy loss in PyTorch that is a hybrid of SoftMax and category cross-entropy loss. Batch Gradient Descent - batch gd() is where all of the learning occurred.

Classification Algorithm

CNN (convolutional neural network) was used to classify data. The training dataset accounts for 80% of the total, with the remaining 20% accounted for by the testing dataset. The training dataset has 56,236 images and the testing dataset has 14,059 images. The model was trained on 56,236 images, 14,056 of which were left unseen by the model to test its accuracy. A pre-trained VGG16 model achieved approximately 87 percent on Train data, 84 percent on Validation data, and 83 percent on Test data after fine-tuning. VGG-16 is made up of three major parts: convolution, pooling, and fully connected layers.

- Convolution layer- In this layer, filters are applied to images to extract features. The most important parameters are kernel size and stride.
- Pooling layer- Its purpose is to reduce a network's spatial size in order to reduce the number of parameters and computations.
- Fully Connected- These are fully connected connections to previous layers, similar to a simple neural network.

Flask Web App

We'll build a flask web application and deploy it to the cloud after we finish this model.

CONCLUSION

Crop protection is a challenging task in organic farming. This requires a thorough understanding of the crop being grown, as well as potential pests, pathogens, and weeds. In our system, we used images of healthy or diseased plant leaves to train a special deep learning model based on a special architectural convolution network to detect plant diseases and their supplements. The system described above can be upgraded to a real-time video entry system that allows unattended plant care. Another feature that can be added to certain systems is an intelligent system that cures identified illnesses. Plant disease management, according to research, can increase yields by up to 50%.

The goal of this review was to explain how to use DL to detect plant diseases. Numerous visualisation techniques/mappings for recognising disease symptoms were also summarised. Despite significant progress in the last three to four years, the following research gaps remain: -

- In the majority of the studies, the PlantVillage dataset was used to assess the accuracy and performance of the respective DL models/architectures (as described in the preceding sections). Despite the fact that this dataset contains a large number of images of various plant species and diseases, the background is simple/plain. However, for a realistic scenario, the real environment must be considered.

- Hyperspectral/multispectral imaging is a relatively new technology that has been applied in a number of fields of study. As a result, it should be used in conjunction with efficient DL architectures to detect plant diseases before symptoms appear.

- A more efficient method of detecting disease spots in plants should be implemented to save money by avoiding the use of unnecessary fungicide/pesticide/herbicide.

- Because plant disease severity varies over time, DL models should be improved/modified to detect and classify diseases throughout their entire life cycle.

- The DL model/architecture should be efficient in a wide range of lighting conditions, so datasets should not only represent the real environment but also include images captured in various field scenarios.

- A comprehensive study is required to understand the factors influencing plant disease detection, such as dataset classes and size, learning rate, illumination, and so on.

REFERENCES

1. S. D. Khirade and A. B. Patil, "Plant Disease Detection Using Image Processing," 2015 International Conference on Computing Communication Control and Automation, 2015, pp. 768-771, doi:10.1109/ICCUBEA.2015.153.
2. S. C. Madiwalar and M. V. Wyawahare, "Plant disease identification: A comparative study," 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), 2017, pp. 13-18, doi: 10.1109/ICDMAI.2017.8073478.
3. P. Moghadam, D. Ward, E. Goan, S. Jayawardena, P. Sikka and E. Hernandez, "Plant Disease Detection Using Hyperspectral Imaging," 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2017, pp.1-8, doi: 10.1109/DICTA.2017.8227476.
4. S. D.M., Akhilesh, S. A. Kumar, R. M.G. and P. C., "Image based Plant Disease Detection in Pomegranate Plant for Bacterial Blight," 2019 International Conference on Communication and Signal Processing (ICCSP), 2019, pp. 0645-0649, doi: 10.1109/ICCSP.2019.8698007.
5. G. Shrestha, Deepsikha, M. Das and N. Dey, "Plant Disease Detection Using CNN," 2020 IEEE Applie Signal Processing Conference (ASPCON), 2020, pp. 109-113, doi: 10.1109/ASPCON49795.2020.9276722.
6. Mohanty SP, Hughes DP and Salathé M (2016) Using Deep Learning for Image-Based Plant Disease
7. R. M. Haralick, K. Shanmugam and I. Dinstein, "Textural Features for Image Classification," in IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, no. 6, pp. 610-621, Nov. 1973, doi: 10.1109/TSMC.1973.4309314.
8. Breiman,L.Random Forests. Machine Learning 45, 5–32 (2001). https://doi.org/10.1023/A:1010933404324.

