An Experimental Study for Crop Disease Detection using CNN

// add NAMES

[1, 2, 3] *Department of Computer Science and Engineering, SRM University Amravati, Andhra Pradesh, India.*

Abstract

With the development of advanced technologies in today's world, there has been equal increase in the amount of pollution which in turn affects the life of plants. It is important to protect plant life in order to save environment and move forward towards a sustainable living. The work presented in this project briefs about identifying different diseases from the tomato plant leaves. The dataset chosen incudes 4585 diseased images of the tomato leaves classified under 10 different diseases. The CNN model presented in this study helps to identify the type of disease present on the tomato leaves. The accuracy is found to be 87.74%.

Keywords— Convolutional Neural Networks (CNN), Keras, Tensorflow, Deep Learning, Data augmentation, Fine tuning.

I. INTRODUCTION

Current advances have enabled human culture to create sufficient food to satisfy the need of in excess of 7 billion individuals. In any case, food shortage is consistently there. Plant diseases are not just a danger to food security at a worldwide scale, however, can likewise have deplorable ramifications for smallholder ranchers whose vocations rely upon healthy crops [1].

Plant diseases radically diminish the quality and amount of crops which annihilates the expected advantages of farming. Diminishing yield misfortune to diseases will typically allow farmers to produce food all the more productively, profiting the agrarian industry and the climate. Plant diseases are amazingly critical to be recognized and appropriately restored at an underlying stage else, the uncured disease may influence the whole ranch what's more, ruin it. At present, ranchers need to painstakingly examine each and each crop intermittently to recognize diseases, which is an very testing and tedious assignment (outstandingly at the point when the quantity of crops is colossal). In the event that we can mechanize the interaction, this will yield huge advantages for the rancher. Consequently, our fundamental goal is to diminish the human exertion and increment the accuracy to recognize the disease and dispose of the conventional technique for visiting the ranch for recognizing diseases by building up an mechanized framework for the equivalent [2].

India is a developed country and about 70% of the populace relies upon farming. Ranchers have an enormous scope of variety for choosing different appropriate yields and tracking down the reasonable pesticides for the plant. Infection on plant prompts a critical decrease in both the quality and amount of farming items. The investigations of plant infection allude to the investigations of outwardly detectable examples on the plants. Observing of wellbeing and infection on plant assumes a significant part of ineffective development of yields in the ranch. In the early days, the checking and investigation of plant illnesses were done physically by the mastery individual around there. This requires a huge measure of work and furthermore requires inordinate processing time.

As of late, with the effective use of profound learning model addressed by the convolutional neural network (CNN) in numerous fields of PC vision (CV, PC vision), for instance, traffic discovery [3], clinical Image Recognition [4], Scenario text identification [5], appearance acknowledgement [6], face recognition [7], and so forth A few plant sicknesses and nuisances identification techniques dependent on profound learning are applied in genuine farming practice, and some homegrown and unfamiliar organizations have built up an assortment of profound Learning-based plant infections and bugs location WeChat applet and photograph acknowledgement APP programming. In this manner, plant infections and irritations discovery technique dependent on profound learning have significant scholarly exploration esteem, yet additionally has an exceptionally wide market application prospect.

This paper is worried about the advancement of a computerized framework that can naturally, recognize disease utilizing leaf image classification. So, a basic CNN model is presented in this work that would take tomato leaf images of 4585 in number as input. The model is trained and tested in 10 different tomato leaves with 20 epochs. The main purpose of this paper is to provide a solution to identify plant diseases by implementing CNN procedures on tomato leaves. Hence, we present an experimental performance analysis using CNN. In short, the key idea of our proposed work in this paper is first we perform the Data augmentation and then fine tuning. The training and testing of our model are performed by taking different combinations of training and testing datasets.

Main/key contributions of the proposed work in this paper are described as follows.

1. To pre-process the data and implement data augmentation on images.
2. Training and testing the model using CNN with different layers.
3. Comparing the learning performance of the model with loss and accuracy parameters, comparing scores obtained by the algorithm, and computational time requirement.

The rest of the paper is organized as follows. Section II includes the background and related work. Section III describes the proposed crop disease prediction using CNN. Section IV explains test setup and results and analysis. At the end section, V highlights the conclusion and future works of the proposed crop disease detection

II. RELATED WORK

G. Madhulatha et al. [8] clarified in their work that Plant diseases can cause a decrease in the horticultural item quality and creation. This is extremely essential to discover the plant diseases at an early stage for worldwide wellbeing and prosperity. Programmed plant disease location is turning into a unmistakable examination area. It gives benefits in checking the huge crop fields and helps in recognizing the side effects of the disease when they are found on the leaves. In this paper, the basically, center around discovering the plant diseases and which will lessen the crop misfortune and thus builds the creation productivity. Our proposed work distinguishes the indications of plant diseases at the extremely beginning stage and groups plant disease based on the side effects utilizing a Deep Learning (DL) procedure. The proposed approach perceives the diseases utilizing a profound CNN, with the best accuracy of 96.50%. This accuracy rate approves the model presentation to early warning or warming device.

Shima Ramesh et al. [9] briefed that, Crop diseases are an important danger to food security, anyway their speedy distinctive verification stays inconvenient in various pieces of the world in view of the non-participation of the significant establishment. Rise of exact methods in the field of leaf-based image characterization have shown amazing outcomes. This paper utilizes Random Forest in distinguishing among healthy and diseased leaf from the data sets made. Our proposed paper incorporates different periods of execution specifically dataset creation, highlight extraction, training the classifier and grouping. The made datasets of diseased and healthy leaves are by and large trained under Random Forest to arrange the diseased and healthy images. For separating highlights of a image they utilized Histogram of an Oriented Inclination (HOG). Generally speaking, utilizing AI to train the enormous data sets accessible freely gives us a reasonable method to distinguish the disease present in plants in a monster scope.

Sharath DM et al. [10] clarified about the Majority populace of the world relies upon agribusiness as their essential occupation to procure their pay. In the event that any issues happen in that essential area, it will influence the work of the populace unfavorably. Consequently, it is important to keep up the appropriate equilibrium in the horticulture area by keeping something similar from antagonistic impacts like the dry season, plant diseases and so forth. In the farming area, particularly agriculture yield more pay to the ranchers than different crops. These crops are inclined to numerous diseases effectively and manual recognition of the disease in the crop is a lot of troublesome in the early stage. To keep away from blunders due to manual discovery of diseases, Machine learning techniques are utilized. Image classification is finished by catching the contaminated district of the image. The contaminated image is accommodated upgrade followed by a image division. at that point, the fragmented image is given as a contribution for the grouping utilizing the convolutional neural network. Further developed an android application.

Garima Shrestha et al. [11] gave experiences into an outline of the plant disease location utilizing various calculations. A CNN based strategy for plant disease location has been proposed here. Reenactment study and investigation is done on example images regarding time intricacy and the space of the tainted area. It is finished by a image classification strategy. An aggregate of 15 cases have been taken care of to the model, out of which 12 cases are of diseased plant leaves, in particular, Bell Paper Bacterial Spot, Potato Early Blight, Potato Late Blight, Tomato Target Spot, Tomato Mosaic Virus, Tomato Yellow Leaf Curl Virus, Tomato Bacterial Spot, Tomato Early Blight, Tomato Late Blight, Tomato Leaf Mold, Tomato Septoria Leaf Spot and Tomato Spider Mites and 3 instances of healthy leaves specifically, Bell Paper Healthy, Potato Healthy and Tomato Healthy. The test accuracy is gotten as 88.80%. Diverse execution grids are inferred for something similar.

Writer name	**Proposed Methodology**	**Limitations**
G. Madhulatha et al. [8]	**CNN-96.50%**	**Classifies limited number of diseases**
Shima Ramesh et al. [9]	**Random forest and HOG-70%**	**Small dataset of 160 images and less accuracy.**
Sharath DM et al. [10]	**Image classification and CNN**	**Restricted to single plant leaves**
Garima Shrestha et al. [11]	**CNN- 88.80%**	**Less image labels used and less accuracy**

Table 1: Summary of existing works

III. PROPOSED WORK

A basic CNN model with Data acquisition and fine tuning has been implemented in this project.

1. Convolution Neural Network-

The degrees of progress in Computer Vision with Deep Learning has been constructed and finished with time, in a general sense more than one explicit computation — a Convolutional Neural Network. A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning computation that can take in a data image, consign importance (learnable burdens and inclinations) to various points/objects in the image and have the choice to isolate one from the other.

The pre-taking care required in a ConvNet is a great deal lower when diverged from other request estimations. While in unrefined methods channels are hand-planned, with enough preparation, ConvNets can get comfortable with these channels/characteristics. The design of a ConvNet is similar to that of the accessibility illustration of Neurons in the Human Brain and was awakened by the relationship of the Visual Cortex. Particular neurons respond to enhancements simply in a limited area of the visual field known as the Receptive Field. A combination of such fields covers to cover the entire visual district [12].

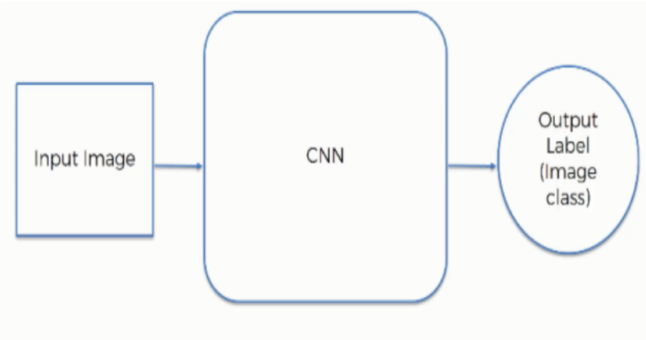


Fig 1: Elements interaction in a CNN model

There are different models of CNN's accessible which have been key in building calculations which power and will control AI all in all within a reasonable time frame. Some of them have been recorded beneath:

- LeNet
- AlexNet
- VGGNet
- GoogLeNet
- ResNet
- ZFNet

Each layer of the convolutional neural network can either be:

- Convolutional layer -CONV- followed with an activation function
- Pooling layer -POOL- as detailed above
- Fully connected layer -FC- a layer which is basically similar to one from a feedforward neural network

Stage 1: CNN Layer

At the convolutional layer, we apply convolutional products, using many filters this time, on the input followed by an activation function ψ.

More preciously, at the $\mathbf{l^{th}}$ **layer**, we denote:

- `Input` : $a^{[l-1]}$ with size $(n_H^{[l-1]}, n_W^{[l-1]}, n_C^{[l-1]})$, $a^{[0]}$ being the image in the input
- `Padding` : $p^{[l]}$, `stride` : $s^{[l]}$
- `Number of filters` : $n_C^{[l]}$ where each $K^{(n)}$ has the dimension: $(f^{[l]}, f^{[l]}, n_C^{[l-1]})$
- `Bias` of the n^{th} convolution: $b_n^{[l]}$
- `Activation function` : $\psi^{[l]}$
- `Output` : $a^{[l]}$ with size $(n_H^{[l]}, n_W^{[l]}, n_C^{[l]})$

And we have:

$\forall n \in [1, 2, ..., n_C^{[l]}]$:

$$conv(a^{[l-1]}, K^{(n)})_{x,y} = \psi^{[l]}\left(\sum_{i=1}^{n_H^{[l-1]}} \sum_{j=1}^{n_W^{[l-1]}} \sum_{k=1}^{n_C^{[l-1]}} K_{i,j,k}^{(n)} a_{x+i-1,y+j-1,k}^{[l-1]} + b_n^{[l]}\right)$$
$$dim(conv(a^{[l-1]}, K^{(n)})) = (n_H^{[l]}, n_W^{[l]})$$

Thus:

$$a^{[l]} = [\psi^{[l]}(conv(a^{[l-1]}, K^{(1)})), \psi^{[l]}(conv(a^{[l-1]}, K^{(2)})), ..., \psi^{[l]}(conv(a^{[l-1]}, K^{(n_C^{[l]})}))]$$
$$dim(a^{[l]}) = (n_H^{[l]}, n_W^{[l]}, n_C^{[l]})$$

With:

$$n_{H/W}^{[l]} = \left\lfloor \frac{n_{H/W}^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor ; s > 0$$
$$= n_{H/W}^{[l-1]} + 2p^{[l]} - f^{[l]}; s = 0$$
$$n_C^{[l]} = number\ of\ filters$$

The `learned parameters` at the l^{th} layer are:

- **Filters** with $(f^{[l]} \times f^{[l]} \times n_C^{[l-1]}) \times n_C^{[l]}$ parameters
- **Bias** with $(1 \times 1 \times 1) \times n_C^{[l]}$ parameters (broadcasting)

Fig 2: CNN layer equations

Stage 2: ReLU Layer

The second piece of this progression will include the Rectified Linear Unit or ReLU. We will cover ReLU layers and investigate how linearity functions with regards to Convolutional Neural Networks.

Stage 3: Pooling

In this part, we'll cover pooling and will see precisely how it by and large functions. Our nexus here, nonetheless, will be a particular kind of pooling; max pooling. We'll cover different methodologies, however, including mean (or aggregate) pooling. This part will end with an exhibit made utilizing a visual intuitive device that will figure the entire idea.

In general, considering the j^{th} node of the i^{th} layer we have the following equations:

$$z_j^{[i]} = \sum_{l=1}^{n_{i-1}} w_{j,l}^{[i]} a_l^{[i-1]} + b_j^{[i]}$$
$$\rightarrow a_j^{[i]} = \psi^{[i]}(z_j^{[i]})$$

The input $a^{[i-1]}$ might be the result of a convolution or a pooling layer with the dimensions $(n_H^{[i-1]}, n_W^{[i-1]}, n_C^{[i-1]})$.

In order to be able to plug it into the fully connected layer we `flatten` the tensor to a `1D vector` having the dimension: $(n_H^{[i-1]} \times n_W^{[i-1]} \times n_C^{[i-1]}, 1)$, thus:

$$n_{i-1} = n_H^{[i-1]} \times n_W^{[i-1]} \times n_C^{[i-1]}$$

The `learned parameters` at the l^{th} layer are:

- **Weights** $w_{j,l}$ with $n_{l-1} \times n_l$ parameters
- **Bias** with n_l parameters

$$a_{x,y,z}^{[l]} = pool(a^{[l-1]})_{x,y,z} = \phi^{[l]}((a_{x+i-1,y+j-1,z}^{[l-1]})_{(i,j)\in[1,2,...,f^{[l]}]^2})$$
$$dim(a^{[l]}) = (n_H^{[l]}, n_W^{[l]}, n_C^{[l]})$$

With

$$n_{H/W}^{[l]} = \left\lfloor \frac{n_{H/W}^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor ; s > 0$$
$$= n_{H/W}^{[l-1]} + 2p^{[l]} - f^{[l]}; s = 0$$
$$n_C^{[l]} = n_C^{[l-1]}$$

Fig 3: Pooling layer equations

Stage 4: Flattening

This will be a short breakdown of the smoothing association and how we move from pooled to evened out layers when working with Convolutional Neural Networks.

Stage 5: Full Connection

In this part, all that we covered all through the fragment will be solidified. It clarifies a more full image of how Convolutional Neural Networks work and how the "neurons" that are finally conveyed get comfortable with the plan of images [13].

In general, considering the j^{th} node of the i^{th} layer we have the following equations:

$$z_j^{[i]} = \sum_{l=1}^{n_{i-1}} w_{j,l}^{[i]} a_l^{[i-1]} + b_j^{[i]}$$
$$\rightarrow a_j^{[i]} = \psi^{[i]}\left(z_j^{[i]}\right)$$

The input $a^{[i-1]}$ might be the result of a convolution or a pooling layer with the dimensions $\left(n_H^{[i-1]}, n_W^{[i-1]}, n_C^{[i-1]}\right)$.

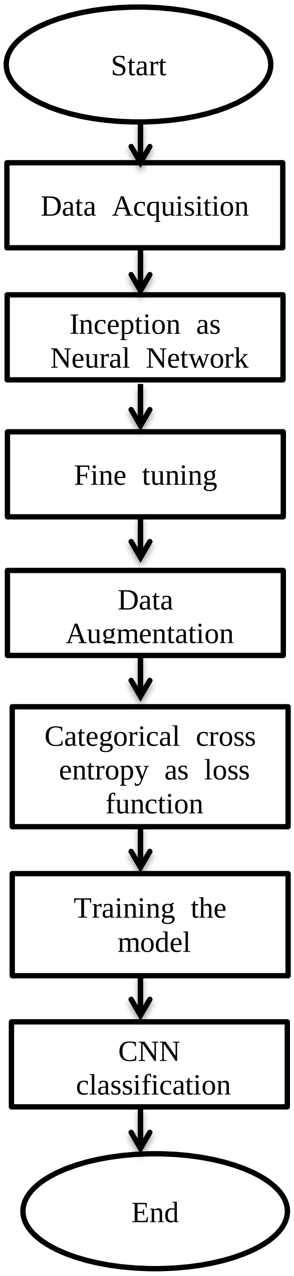
In order to be able to plug it into the fully connected layer we `flatten` the tensor to a `1D vector` having the dimension: $\left(n_H^{[i-1]} \times n_W^{[i-1]} \times n_C^{[i-1]}, 1\right)$, thus:

$$n_{i-1} = n_H^{[i-1]} \times n_W^{[i-1]} \times n_C^{[i-1]}$$

The `learned parameters` at the l^{th} layer are:

- **Weights** $w_{j,l}$ with $n_{l-1} \times n_l$ parameters
- **Bias** with n_l parameters

Fig 4: Fully connected layer equations



Description of flowchart-

1. Data acquisition

Data acquisition is the way toward inspecting signals that action genuine states of being and changing over the subsequent examples into computerized numeric qualities that can be controlled by a PC.

2. Inception Networks-

When planning a convolutional neural organization, we frequently need to pick the kind of layer: CONV, POOL or FC. The commencement layer does them all. The consequence of the relative multitude of activities is then linked in a solitary square which will be the contribution of the following layer as follows

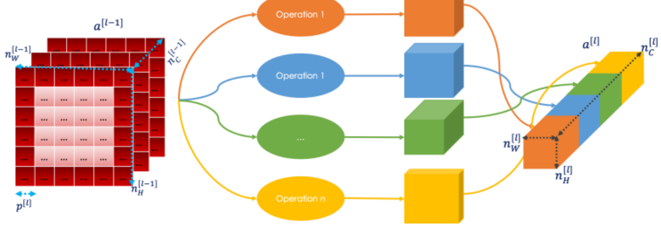


Fig 7: Inception networks block diagram

3. Fine Tuning

Unfreeze a few of the top layers of a frozen model base and Unfreeze a couple of the top layers of a frozen model base and together train both the recently added classifier layers and the last layers of the base model. This permits us to "fine-tune" the higher-request include portrayals in the base model to make them more applicable for the particular assignment.

4. Data Augmentation

Image data augmentation is a method that can be utilized to falsely grow the size of a preparation dataset by making adjusted renditions of images in the dataset. Data augmentation is a procedure to falsely make new preparing data from existing preparing data. This is finished by applying space explicit procedures to models from the preparation data that make new and diverse preparing models. Image data augmentation is maybe the most notable kind of data augmentation and includes making changed adaptations of images in the preparation dataset that have a place with a similar class as the first image. Changed incorporate a scope of tasks from the field of image control, like movements, flips, zooms, and considerably more. The expectation is to extend the preparation dataset with new, conceivable models. This implies, varieties of the preparation set images that are probably going to be seen by the model. Present day profound learning calculations, for example, the convolutional neural organization, or CNN, can learn highlights that are invariant to their area in the image. All things considered, augmentation can additionally help in this change invariant way to deal with learning and can help the model in learning highlights that are likewise invariant to changes. Image data augmentation is ordinarily simply applied to the preparation dataset, and not to the approval or test dataset. This is unique in relation to data arrangement, for example, image resizing and pixel scaling; they should be performed reliably across all datasets that associate with the model [14].

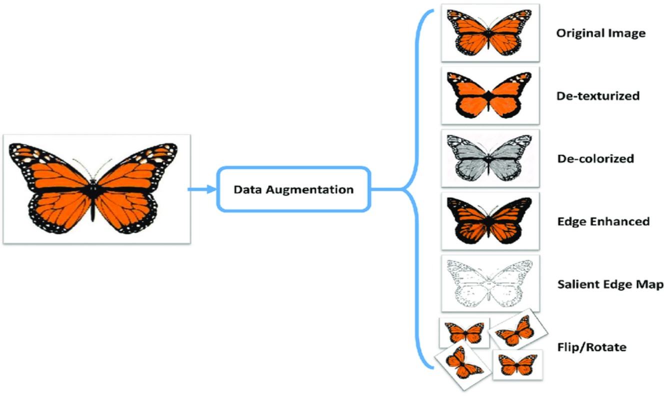


Fig 8: Data augmentation

5. Categorical cross entropy as loss function

Categorical crossentropy is a loss functon that is utilized in multi-class order errands. These are errands where a model can just have a place with one out of numerous potential classes, and the model should choose which one. Officially, it is intended to measure the contrast between two likelihood circulations..

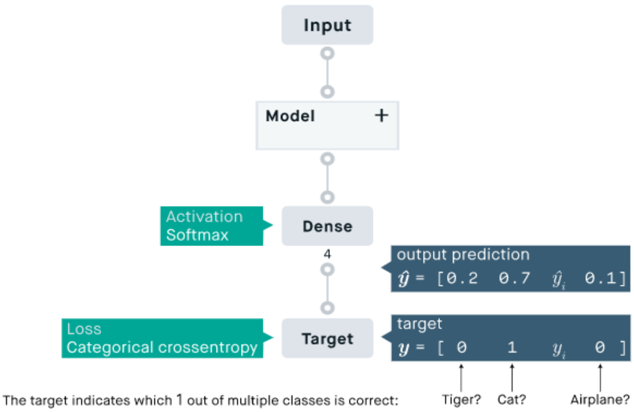


Fig 9: Categorical cross entropy as loss function flow

6. Training the model

Training a model basically implies picking up (deciding) great qualities for every one of the loads and the predisposition from marked models. In regulated learning, an AI calculation fabricates a model by inspecting numerous models and endeavouring to track down a model that limits misfortune; this interaction is called exact risk reduction or minimising failure.

7. CNN Classification

The basic CNN classifier is explained in the image below.

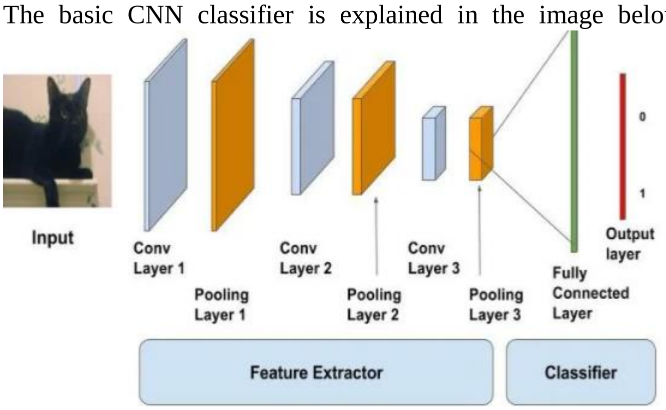


Fig 10: CNN Classifier

I. EXPERIMENTAL SETUP AND RESULTS ANALYSIS

To check the performance of the CNN model built for crop disease detection, we performed an extensive experiment. The experimental setup and results and analysis sections are described as follows:

(a) Experimental Setup: We conducted our experiment on Intel core i5 processor with 4 GB RAM size. Further, we used Jupiter Notebook with Python 3. 7.4. The key motivation to use python language for the implementation of the CNN model.

	Medium	High-end
CPU	4 Physical Cores	12 Physical Cores
Memory	4 GB	24 GB
Disk	4 disks x 1TB = 4 TB	12 disks x 3TB = 36 TB
Network	1 GB Ethernet	10 GB Ethernet or InfiniBand

Fig 11: Hardware setup

(b) Results and analysis:

This particular section provides experimental outcomes of the proposed method. First, the details of the dataset are provided.

Data set Description-

The dataset consists of 4585 images of diseased tomato leaves. The names of the diseased tomato leaves and its classification are mentioned below. The dataset contains diseased images of tomato leaves divided into training and validation sets.

Classification of Tomato leaves is considered as-

- Tomato___Bacterial_spot': 0,
- Tomato___Early_blight': 1,
- Tomato___Late_blight': 2,
- Tomato___Leaf_Mold': 3,
- Tomato___Septoria_leaf_spot': 4,
- Tomato___Spider_mites
- Two-spotted_spider_mite': 5,
- Tomato___Target_Spot': 6,
- Tomato___Tomato_Yellow_Leaf_Curl_Virus': 7,
- Tomato___Tomato_mosaic_virus': 8,
- Tomato___healthy': 9

Epochs used-20

- Training set-The pictures present in this set are utilized to prepare the model
- Validation set-The pictures present in the validation set are utilized to assess the model exhibition. The example of data used to give a fair-minded assessment of a model fit on the training dataset while tuning model hyper boundaries. The assessment turns out to be

more one-sided as the ability on the validation dataset is fused into the model design.

- Training accuracy-The accuracy of a model on models it was built on.
- Validation accuracy-The accuracy you compute on the data set you don't use for training, however, you use (during the training interaction) for approving (or "testing") the speculation capacity of your model.
- Training loss is the mistake on the training set of data
- Validation loss is the mistake subsequent to running the validation set of data through the prepared organization [15].

Meaning of the Terms:

- True Positive (TP) : Observation is positive, and is anticipated to be positive.
- False Negative (FN) : Observation is positive, yet is anticipated negatively.
- True Negative (TN) : Observation is negative, and is anticipated to be negative.
- FalsePositive (FP) : Observation is negative, however is anticipated positive.In order to evaluate the accuracy of the proposed model, we calculate using the given formulae-

Accuracy is the ability to determine the correctness or closeness of personality categorization. The formula for accuracy can be given by

Accuracy = (TP+TN)/(TP+TN+FP+FN)

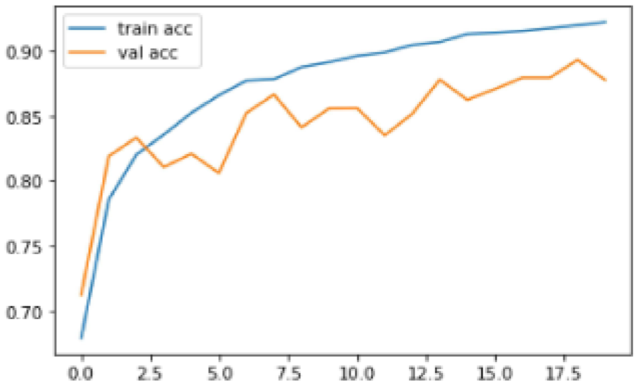


Fig 12: Graphs for Training Vs Validation accuracy

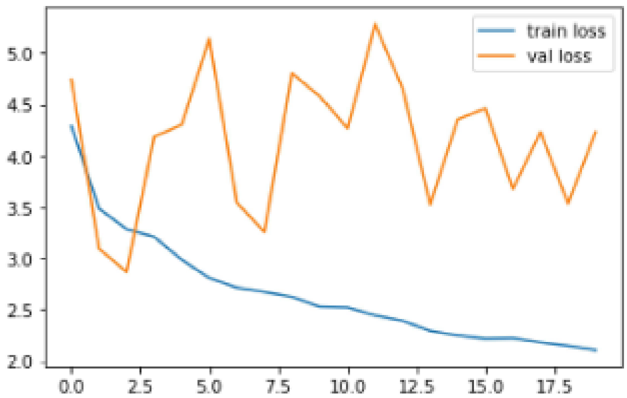


Fig 13: Graph 7: Training Vs Validation loss

	LOSS	ACCURACY
TRAINING	2.11%	92.1%
VALIDATION	4.22%	87.74%

Table 2: Training and Validation loss and accuracy generated for 20 Epochs

V. CONCLUSION AND FUTURE WORKS

In this research work, we presented a basic CNN model for leaf disease detection. Results were reliable with the hypothetical strength also, constraints of each approach. The CNN model presented in this study helps to identify the type of disease present on the tomato leaves. The accuracy is found to be 87.74%. The accuracy can be increased by increasing the epochs. This work can be extended by developing a web application that takes leaf images as input and predict the disease accurately.

REFERENCE:

[1] Mohanty Sharada P., Hughes David P., Salathé Marcel. Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science, 2016.

[2] Manpreet Sandhu, Pratik Hadawale, Saumaan Momin, Prof. Ajitkumar Khachane. Plant Disease Detection using ML and UAV. International Research Journal of Engineering and Technology (IRJET), 2020.

[3] Yang D, Li S, Peng Z, Wang P, Wang J, Yang H. MF-CNN: traffic flow prediction using convolutional neural network and multi-features fusion. IEICE Trans Inf Syst. 2019;102(8):1526–36.

[4] Sundararajan SK, Sankaragomathi B, Priya DS. Deep belief cnn feature representation based content based image retrieval for medical images. J Med Syst. 2019;43(6):1–9.

[5] Melnyk P, You Z, Li K. A high-performance CNN method for offline handwritten chinese character recognition and visualization. Soft Comput. 2019;24:7977–87.

[6] Li J, Mi Y, Li G, Ju Z. CNN-based facial expression recognition from annotated rgb-d images for human–robot interaction. Int J Humanoid Robot. 2019;16(04):504–5.

[7] Kumar S, Singh SK. Occluded thermal face recognition using bag of CNN(BoCNN). IEEE Signal Process Lett. 2020;27:975–9.

[8] G. Madhulatha, O. Ramadevi. Recognition of Plant Diseases using Convolutional Neural Network. Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)

[9] sShima RameshNiveditha M, Pooja R, Prasad Bhat N, Shashank N. Mr. Ramachandra Hebbar, P V Vinod. Plant Disease Detection Using Machine Learning. International Conference on Design Innovations for 3Cs Compute Communicate Control.

[10] Sharath D M, S Arun Kumar, Roham M G, Akhilesh. Disease Detection in Plants using Convolutional Neural Network. Third International Conference on Smart Systems and Inventive Technology (ICSSIT 2020).

[11] G. Shrestha, Deepsikha, M. Das and N. Dey, "Plant Disease Detection Using CNN," 2020 IEEE Applied Signal Processing Conference (ASPCON), Kolkata, India, 2020.

[12] The Ultimate Guide to Convolutional Neural Networks (CNN). Retrieved from https://www.superdatascience.com/blogs/the-ultimate-guide-to-convolutional-neural-networks-cnn

[13] How to Configure Image Data Augmentation in Keras by Jason Brownlee.Retrieved from https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/

[14] Validation accuracy vs Testing accuracy.Retrieved from https://stats.stackexchange.com/questions/401696/validation-accuracy-vs-testing-accuracy#:~:text=In%20other%20words%2C%20the%20test,or%20for%20%22early%20stopping%22.

[15] What is Train loss, Valid loss, and Train/Val mean in NNs. Retrieved from https://stackoverflow.com/questions/36963054/what-is-train-loss-valid-loss-and-train-val-mean-in-nns