Identifying Infection In Plants Using Inception V3 On Plant Leaves

Jaydutt Joshi, Keith Frank, Shariq Shaikh, Isaac Ayo

I. Abstract

Agriculture plays an important role in large industries, and every person in society is dependent on it. So crop monitoring is a crucial task to know crop health. As we know, agriculture is the backbone of the economy, so productivity and quality are key points in the agriculture sector. To grow the high-quality product and maximize the yield production properly and in time, “Crop Disease Detection" is needed. Disease in crops is natural or very common, but identifying it in a timely manner is essential in the agriculture sector. This paper presents the crop disease analysis using the machine learning algorithm, which uses crop images. Convolutional Neural Network used for crop disease analysis. Images are pre-processed, and essential features are extracted from images then crop diseases are classified using the machine learning classification algorithm. The proposed system's application would be useful to the farmer to detect the disease of the crop just by capturing the photo of the crop and submitting it to the application. Convolutional Neural Networks have a better ability to deal with images and have greater accuracy in more amounts of data.

II. Problem Statement

Design and implement the algorithm which will identify the different types of crop disease using the Convolutional Neural Network algorithm. Automatic identification of plant illnesses is a critical task since it might be useful in monitoring huge fields of crops and, as a result, automatically detect diseases based on symptoms on plant leaves. As a result, using image processing techniques to detect plant disease automatically provides higher accuracy and assistance for disease management visual identification, on the other hand, is less accurate and time consuming. As a result, it will be necessary to create and develop a machine learning method for detecting illness in wheat and sugarcane crops in a timely manner in order to assist farmers in increasing productivity.

With drastic changes in the environment, the number of diseases are also increasing. Modern technologies used for crop cultivation and also various kinds of fertilizers and pesticides are used for increased productivity. But farmers have not known about the actual contents and proper knowledge of the fertilizer. This increased and improper use of this fertilizer and pesticide also causes the various different diseases on the crop.

III. Objectives

1. To collect the data, different images of crops with different diseases are present.
2. To study the best suitable algorithm which deals with image data.
3. To study and analyze different machine learning algorithms for crop disease detection.
4. To investigate and implement algorithms which will identify the crop disease and its name.

5. To detect the disease of different crop images.

6. To evaluate the proposed system

IV. Methodology

A crop disease is defined as anything that prevents a crop from performing to its maximal potential yield production. A crop disease has impacted society and world olden times. Crop diseases have caused enormous economic losses in each one the countries. Detection of diseases in crops refers to the identification of disease present or not. Failures in disease diagnosis show low crop production and low quality, and as a result trade. In this proposal we try to detect the crop disease with more accuracy and a wide number of diseases for two types of crop. This system detects the total 25 different types of most common disease in wheat and sugarcane using the satellite or digital images. For the more accurate result we use image processing for feature extraction, such as color, texture, shape using CNN`s Conv2d, subsampling and ReLU layers. Classification is done using the machine learning`s CNN(convolutional Neural Network) algorithm. CNN machine learning classifier is used for classifying the different kinds of disease of wheat and sugarcane classification.

Training Methodology: We have trained our networks with stochastic gradient utilizing the TensorFlow distributed machine learning system using 50 replicas running each on a NVidia Kepler GPU with batch size 32 for 100 epochs. Our earlier experiments used momentum with a decay of 0.9, while our best models were achieved using RMSProp with decay of 0.9 and $\varrho = 1.0$. We used a learning rate of 0.045, decayed every two epochs using an exponential rate of 0.94. In addition, gradient clipping with threshold 2.0 was found to be useful to stabilize the training. Model evaluations are performed using a running average of the parameters computed over time.

V. Dataset

We have used the CrowdAI Plant Disease Dataset for disease identification. It has 25 classes across five crops/plants

i. Distribution

Plant	No. of Diseases	Images
Apple	4	3,051
Corn	4	3,732
Grape	4	3,942
Potato	3	2,062
Tomato	10	17,860

ii. Samples


Apple Black Rot


Healthy Apple


Tomato Late Blight


Healthy Tomato

Although the above images show a clear contrast between healthy and infected plants, there are a huge amount of images where infected leaves are extremely hard to distinguish from healthy ones.


Tomato Early Blight


Healthy Tomato

The images represent the fact that healthy leaves can look worse as compared to infected ones

VI. Approach

i. Transfer Learning

Transfer Learning for machine Learning is when elements of a pre-trained model are reused in a new machine learning model. Supposing both models are developed to perform similar tasks, then generalised knowledge can be 'transferred' between the two models. This technique in Machine Learning is good in that it reduces the amount of labelled data and also the resources required to train new models. As machine learning evolves, it is becoming a pivotal part of the field as the technique is being used more and more in the development process.

In our day to day world machine learning is becoming a norm, it is applicable and being made use of in several industries, ranging from healthcare, to banking, to customer services, to shopping and to all imaginable sectors. Practical examples involve speech recognition and refining adverts and marketing campaigns to better cater to the needs of clients and consumers, thereby resulting in a better return on investment.

Transfer Learning itself is not a distinct machine learning algorithm, rather it is a method that is implemented while training models. The knowledge obtained from previous training is recycled and reused to help optimize the performance of a new task. The previously trained task and the new task will of course share similarities, this is made possible by allowing for the originally trained model to have a high level of generalisation, making it easy to adapt to the new unseen data.

This technique eliminates the need to start a new task/model from scratch as attempting to train a new machine learning model can be a cumbersome endeavour in that it causes not only a strain on resources but it produces a lot of downtime which could otherwise have been avoided with the use of transfer learning. Downtime for example, in trying to label a new large dataset will require a lot of time.

With Transfer Learning, the model can then be trained on a readily available dataset that had been previously labelled, and it works well as long as it is a similar task.

ii. Why Transfer Learning?

Rather than dedicating hours again to training a new machine learning model, transfer learning allows for us to use a pre-existing dataset and build upon such. This process proceeds by taking the parts that are relevant in an already existing model and applying this to solve a similar new problem. As mentioned above, generalisation is a major part of transfer learning, what this means is that in developing and transferring from the original model, only the knowledge that is fit to be used in another model in differing scenarios and conditions is taken into account.

So what we find is that the models that are to be used in transfer learning are not developed in a way such that they are very closely knit to the training dataset but rather, they tend to be more generalized.

A prominent use of transfer learning is in image classification. Here, a model is trained with labelled data to identify and classify image subjects. This model is then further modified and recycled (transfer learning) to identify other specific subjects within a set of images. The general elements of the model will remain unchanged, conserving resources and reducing time spent on labelling new data.An example of an element that could be retained in the context of image classification would be the part of the model that is responsible for identifying object edges in an image.

In all the benefit of transfer learning are as follow:

- Saves time, reduces the time needed to train a model, eliminating the need to start the building of a new model from scratch
- Saves resources, there is less computational power required and also a reduction in the amount of data necessary to train the new model
- It also helps to eliminate the problem caused by a lack of labelled training data, by taking advantage of the already made model
- Improves the efficiency of machine learning development and deployment for multiple models
- It provides with a more generalised technique/approach to machine problem solving by pivoting against multiple algorithms to tackle new challenges

Transfer Learning has varied applications, however, three prominent fields in which transfer learning is gaining much traction are:

- Natural Language Processing
- Computer Vision
- Neural Networks

The advent of Transfer Learning means that powerful machine learning models are able to be deployed at scale following the adapted scaled development for specific environments and tasks.

iii. Imagenet

ImageNet is an image database organized according to the WordNet hierarchy (the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images.

In the year 2006, AI researcher Fei-Fei Li met with one of the creators of WordNet to discuss his project and subsequently assembled a team of researchers for the ImageNet project. The first time this database was presented was as a poster in 2009, at the Conference on Computer Vision and Pattern Recognition (Florida).

It has been an instrumental project in the progress of computer vision and also in deep learning research. The ImageNet data is freely available, and can be used by researchers.

This dataset is made up of three components, training data, validation data, and the labels for the images. The training data is made up of 1000 categories and 1.2 million images, and it has been packaged in a way that allows for easy download.

The test data and validation data comprises 150,000 pictures, collected from various search engines, such as Flikr and the likes. The categories (1000) contain not only internal nodes but also leaf nodes of ImageNet, however, there is no overlap. A random subset of 50,000 of the images has been released as validation data alongside a list of the categories (1000). The images left have been released without labels and they are used for evaluation purposes.

The computer vision community is increasingly adopting ImageNet in the task of image classification, where it is not used as the end in and of itself but rather a means to an end. It is used in training deep convolutional neural networks to learn good general-purpose features. The practice involves first training a CNN to be able to carry out image classification on ImageNet (pre-training) and then modifying the features obtained for a new target task (fine-tuning). This methodology has become the industry standard for solving a range of computer vision tasks. The use of ImageNet's pre-trained CNN has yielded staggeringly positive results on various classifications datasets, not only that, but also, action recognition, image segmentation, image captioning, object detection, human pose estimation, optical flow and others.

iv. What makes ImageNet better?

By now I'm sure you're wondering and asking 'What makes ImageNet superior to other pre-trained models and why has it become the cornerstone for transfer learning in machine learning and more specifically computer vision?'
In this section we will be looking to answer this question.

ImageNet set out to address two important needs in computer vision research:

- Object categorization: this was recognized to be one of the most important foundational capabilities for both humans and computer vision.
- Large-scale image database: there was a pressing need for more data to allow for generalizable methods to be developed in the field of machine learning.

These two needs converging resulted in developing ImageNet. Ever since ImageNet has been a front-runner in computer vision and has become the industry standard in transfer learning for image training.

A school of thought believes the sheer size of the dataset is what is accountable for how successful the ImageNet pre-trained CNN features are. The size of the dataset (1.2 million labeled images) is said to be the reason for the success, in that it forces generalization on the representation.
Others, however, believe what is responsible for this success is the large number of distinct object classes (1000), which imposes the need for the network to learn a hierarchy of generalizable features.
Finally, the last school of thought is that the representation has been forced to work harder as it has been turned into a fine-grained recognition problem as many of the classes represented have visual similarities (e.g several varying dog breeds).
Academic benchmarks have been hotly pursued in the past decade in the field of computer vision research, none however, have been pursued as intensely as the ImageNet benchmark. Much progress has been realised in network architectures choosing to be measured against the prominent and successful architecture of ImageNet. This has been the case across a broad array of tasks, such as object detection, image segmentation, and perceptual metrics of images.

A body of work titled 'Do Better ImageNet Models Transfer Better?' sought to investigate the implicit hypothesis that's so prevalent in modern computer vision research, the hypothesis that models that perform better on ImageNet necessarily perform better on other vision tasks.
They went ahead to compare 16 classification networks on 12 image classification datasets.
Upon this study they were able to conclude that "ImageNet architectures generalize well across datasets, but that ImageNet features are less general that had been previously suggested."
The superiority and recognition of ImageNet as a standard of transfer

learning is why we have opted to go with ImageNet.

v. Inception V3

The Inception V3 is a deep learning model for image classification that uses Convolutional Neural Networks. The Inception V3 is a more advanced version of the fundamental model Inception V1, which was first released in 2014 as GoogLeNet. It was created by a Google team, as the name implies.

Overfitting of the data occurs when numerous deep layers of convolutions were utilised in a model. To avoid this, the conception V1 model employs the concept of numerous filters of varying sizes on the same level. As a result, instead of having deep layers in our inception models, we have parallel layers, making our model larger rather than deeper.

Multiple Inception modules make up the Inception model.

The Inception V1 model's core module is made up of four parallel layers. 1) 1x1 convolution 2) 2x2 convolution 3) 3x3 convolution 4) 4x4 convolution

Convolution is the technique of altering an image by applying a kernel over the entire image to each pixel and its local neighbours.

Pooling is a technique for reducing the size of a feature map's dimensions. Pooling can take several forms, but the most popular are maximum pooling and average pooling.

Following are the design principles we took into consideration while designing our model:

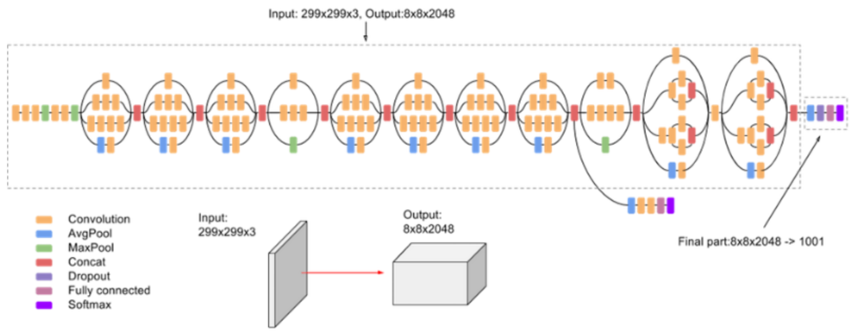
1. Avoids representational bottlenecks, especially early in the network. Feed-forward networks can be represented by an acyclic graph from the input layer(s) to the classifier or regressor. This defines a clear direction for the information flow. For any cut separating the inputs from the outputs, one can access the amount of information passing through the cut. One should avoid bottlenecks with extreme compression. In general the representation size should gently decrease from the inputs to the outputs before reaching the final representation used for the task at hand. Theoretically, information content can not be assessed merely by the dimensionality of the representation as it discards important factors like correlation structure; the dimensionality merely provides a rough estimate of information content.

2. Higher dimensional representations are easier to process locally within a network. Increasing the activations per tile in a convolutional network allows for more disentangled features. The resulting networks will train faster.

3. Spatial aggregation can be done over lower dimensional embeddings without much or any loss in representational power. For example, before performing a more spread out (e.g. 3×3) convolution, one can reduce the dimension of the input representation before the spatial aggregation without expecting serious adverse effects. We hypothesize that the reason for that is that the strong correlation between adjacent units results in much less loss of information during dimension reduction, if the outputs are used in a spatial aggregation context. Given that these signals should be easily compressible, the dimension reduction even promotes faster learning.

4. Balance the width and depth of the network. Optimal performance of the network can be reached by balancing the number of filters per stage and the depth of the network. Increasing both the width and the depth of the network can contribute to higher quality networks. However, the optimal improvement for a constant amount of computation can be reached if both are increased in parallel. The computational budget should therefore be distributed in a balanced way

between the depth and width of the network..



VII. Methodology

i. Creating our own model:

a. <u>Sampling Data</u>

Given that the dataset was highly imbalanced, we performed under/over sampling on the appropriate classes getting the classes weighted within the acceptable 1:2 ratio.

b. <u>Building Model</u>

The model we created had a series of 4 convoluted layers followed by fully connected layers following 'relu' activation. With almost 14 million parameters we expected the model to perform well without overfitting the model.

```
Model: "sequential"
_______________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_9 (Conv2D)           (None, 252, 252, 16)      1216
 conv2d_10 (Conv2D)          (None, 248, 248, 8)       3208
 max_pooling2d_7 (MaxPooling  (None, 124, 124, 8)      0
 2D)
 conv2d_11 (Conv2D)          (None, 120, 120, 32)      6432
 conv2d_12 (Conv2D)          (None, 116, 116, 64)      51264
 max_pooling2d_8 (MaxPooling  (None, 58, 58, 64)       0
 2D)
 flatten_3 (Flatten)         (None, 215296)            0
 dense_6 (Dense)             (None, 64)                13779008
 dropout_2 (Dropout)         (None, 64)                0
 dense_7 (Dense)             (None, 25)                1625
=================================================================
Total params: 13,842,753
Trainable params: 13,842,753
Non-trainable params: 0
```

c. <u>Evaluation</u>

With the accuracy peaking at 82%, this model performed well with certain classes, it failed with most classes with precision and accuracy below 20%. Tomato & its classes still had a higher advantage over most classes due to prevalence and numbers and hence this model was favoring the crop in a large number of cases.

ii. Transfer Learning

As explained previously Inception V3 proved to be the best model for our data. The evaluation below reinforces this claim. Using Tensorflow Hub we were able to acquire the model and train on our dataset with a 20% validation split.

With validation accuracy at 95% and testing accuracy at 96% we were sure of the capabilities of the model.We just had to ensure that the FPs were low on healthy classes since pesticides on healthy plants aren't helpful.

VIII. Evaluation

Class	Precision	Recall	Fscore	Support
Tomato Early Blight	0.692308	0.9	0.782609	10
Corn Maize Northern Leaf Blight	0.833333	1	0.909091	10
Tomato Late Blight	0.888889	0.8	0.842105	10
Tomato Bacterial Spot	0.9	0.9	0.9	10
Tomato Septoria Leaf Spot	0.9	0.9	0.9	10
Apple Black Rot	0.909091	1	0.952381	10
Apple Cedar Rust	1	0.9	0.947368	10
Apple Healthy	1	1	1	10
Apple Scab	1	1	1	10
Corn Maize Cercospora Leaf Spot Gray Leaf Spot	1	0.8	0.888889	10
Corn Maize Common Rust	1	1	1	10
Corn Maize Healthy	1	1	1	10
Grape Black Rot	1	1	1	10
Grape Esca Black Measles	1	1	1	10
Grape Healthy	1	1	1	10
Grape Leaf Blight Isariopsis Leaf Spot	1	1	1	10
Potato Early Blight	1	1	1	10
Potato Healthy	1	1	1	10
Potato Late Blight	1	1	1	10
Tomato Healthy	1	1	1	10
Tomato Leaf Mold	1	1	1	10
Tomato Mosaic Virus	1	0.9	0.947368	10
Tomato Two-Spotted Spider Mite	1	1	1	10
Tomato Target Spot	1	0.9	0.947368	10
Tomato Yellow Leaf Curl Virus	1	1	1	10

IX. References

1. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818-2826, doi: 10.1109/CVPR.2016.308.
2. J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer

Vision and Pattern Recognition, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

3. S. Kornblith, J. Shlens and Q. V. Le, "Do Better ImageNet Models Transfer Better?," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 2656-2666, doi: 10.1109/CVPR.2019.00277.
4. PerceptiLabs. (2021, February 17). Don't Start your Model from Scratch - Use Transfer Learning. PerceptiLabs. Retrieved April 6, 2022, from https://blog.perceptilabs.com/dont-start-your-model-from-scratch-use-transfer-learning/
5. WMIGFTL.(n.d.)People Retrieved April 6, 2022, from http://people.csail.mit.edu/minhuh/papers/analysis/