Disease Detection in Fruits Using Deep Learning

Raju Hosakoti[1], Soma Pavan Kumar[2], Padmaja Jain[3]

[1]*Student, Dept. of ECE, BNMIT, Bangalore,* [2]*Student, Dept. of ECE, BNMIT, Bangalore,* [3]*Assistant Professor, Dept. of ECE, BNMIT*

[1]raju17ec061@bnmit.in,[2]soma17ec079@bnmit.in,[3]padmaja.jain@bnmit.in

***Abstract*— Crop cultivation plays an essential role in the agricultural field. Presently, the loss of crop is mainly due to infected crops, which reduces the production rate. It is very difficult to monitor the diseases manually. It requires a tremendous amount of work, expertise and excessive processing time. Hence deep learning used for the detection of diseases with more accuracy. This paper aims at providing a cost-effective and real-time solution to detect fruit diseases. CNN is used for feature extraction and classification. Deep learning concepts will help to identify the diseases in fruits (Apple) with more accuracy. Hence this results in predicting the disease in early stage so that the necessary actions to cure them can be taken immediately.**

***Keywords*— Convolutional Neural Network (CNN), Deep learning, Flyspeck, Sooty blotch, Apple Rot**

I. Introduction

The emerging of new technologies such as digital image processing and image analysis technology has many applications in the biological field. In India, around 78% of the farmers are marginal and hence they are poor in resources. Therefore, they are not in a position to use the available resources for increasing productivity. A user-friendly software will help the farmer to some extent to detect whether the fruit is diseased or not. The image is processed using the Image Processing techniques and the disease is detected. The disease is detected by our image processing software that helps the farmers to take some precautions. This proves benefits in monitoring large orchards of fruits, and thus automatically detects the diseases as soon as they appear on fruits.

The largest sector in the Indian economy is agriculture and it is also the largest employer. So, the agricultural industry places a significant role in increasing the Indian economy. Daily products in India are cheaper as compared to other parts of the world for example 1] In Europe 1kg of milk costs € 3.54 (as in Jan. 2021), around 313 rupees, whereas the same quantity of milk costs 46 rupees in India, which is 6.8 times lesser. 2] In Western Europe 1kg of apple costs € 2.5 (as in Jan. 2021), around 220 rupees, whereas the same quantity of apple costs 100rs in India. So, this is the average price variation between Milk, Fruits, Vegetables and many other agricultural products like Rice, Wheat compared to other countries. Hence there is a lot of scope for export, which is a new way for the farmer to earn more money which in term will increase the agricultural income which in turn increase the Indian economy. This is possible only if we produce more fresh fruits. So, this paper focuses on disease detection in fruits which helps in increasing the fresh fruits produced.

Fruits like Apple, Grapes and Mango are called cash crops, hence improvement in productivity of these types of fruits is required. Various diseases may attack apple fruits mainly Flyspeck, Apple Rot, Sooty Blotch etc. So, its early detection is very essential. Here we use the deep learning concepts that make use of various CNN algorithms to improve the accuracy. In deep learning, we will train the images hence that helps for early detection of diseases.

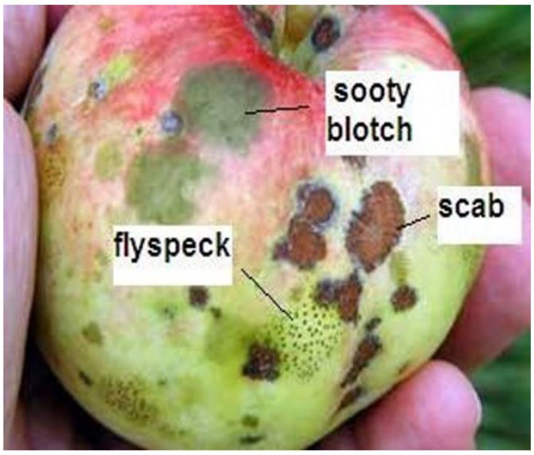


Fig 1. Example of Diseased Apple Fruit

Fig1 shows the different types of diseases that may affect apple. The apple in the above figure is affected by 3 types of diseases they are, 1) Flyspeck 2) Sooty blotch 3) Scab. Flyspeck is a fungal disease with small dots on fruit, Sooty blotch will appear on the surface of the fruit and turns the fruit light black in colour, Scab will turn the fruit dark black.

II. Convolution Neural Network

A Convolutional Neural Network is a Deep Learning algorithm that can take in an input image, assign importance to various aspects in the image and be able to differentiate one from the other. It is a multilayered neural network with a special architecture to detect complex features in data. CNN's have been used in image recognition, powering vision in robots, and for self-driving vehicles. Convolutional Neural Networks will make use of small pre-processing as compared to other image classification algorithms. Hence, CNNs can have the ability to optimization of filters (kernels).

III. Tensor flow

TensorFlow helps build neural network models to classify images. It is a powerful framework that lets you define, customize and tune many types of CNN architectures. It also helps with the flow of data between the various functions and modules used in the python program. The functions and modules are mostly implemented using C program and hence the dataflow between these modules which are responsible for calculating the weights and various other important parameters which are in turn used by other modules and therefore the dataflow between them must be very efficient and fast, this is where TensorFlow plays a very important role.

IV. ImageNet

ImageNet is an open-source repository of images consisting of thousands of classes and millions of images. It is an image database in which millions of images are arranged according to the WordNet hierarchy. Many students and professionals working on Machine Learning projects and this database has played a vital role in many projects as acknowledged by many students and professionals. The data is available for free to researchers for non-commercial use.

V. Convolution

Convolution is a method that is to be performed on the image pixels to crop the outer layer of the input image with a unique value by using a kernel. In the case of convolution, it mainly performs the 'dot product' between the original image and kernel. The kernel, maybe 3*3 or 5*5 image/templet, it is like a filter/sampling filter. The main idea here is to divide the original image into various pixels and the kernel is placed on top of the original image pixels then it will compute the dot product. Similarly, the kernel is moved to the different places and it will compute all dot products then ultimately, we will get one unique image from this dot product called Output image. In the case of kernel movement, we can perform a shift over by one column or shift over by two columns.

VI. Softmax function

Softmax is an 'Activation Function'. The activation function is a non-linear function that decides whether the output from the neuron should be propagated forward or not. If the output from the neuron is high enough then it activates the function and the output propagate forward. These activation functions are very essential because without these activation functions the neural network would not be able to solve non-linear problems. There are so many activation functions that exist they are: 1) Sigmoid 2) Relu 3) Tanh 4) Softmax. Softmax function will convert all score values into "Normalized probability distribution".

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

The above equation represents the mathematical expression for the softmax function. By using this equation ultimately, we will get different valid probabilities i.e., the probability values lie between 0 and 1 and their sum is equal to 1. Once the softmax function is used then it will take only the maximum probability value and propagate forward.

VII. Max Pooling

Pooling means combining the output of a cluster of neurons into a single output. There are mainly two types in pooling 1) Min Pooling 2) Max Pooling. In the case of max pooling, the main idea is to reduce the complexity of the network which will help us to save computational time. In max-pooling, we will use a max-pooling which can be 2*2 in size, and we will use our input image which is divided with different pixel values. Here we will place the max-pooling filter onto the input image. Then ultimately it will take only the maximum value from the input image pixels and then it will place it on to the output pixel. We can perform shift over by 2 here. Hence this max-pooling filter will reduce the size of the input image that leads to reduce the complexity of the network and helps to reduce the computational time.

VIII. Inception Layer Flowchart

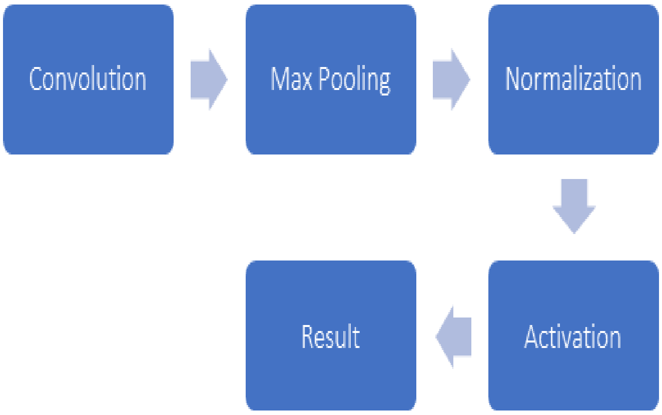


Fig 2. Inception Layer Flowchart

The InceptionV3 model that we used consists of many layers, but, as shown in the figure above, these are the important layers appearing repetitively in the model that we have used:

1. Convolution.
2. Max Pooling.

3. Normalization.
4. Activation.

Convolution is mainly the process of combining two signals to produce a single signal in Signal Processing, but, whereas in Digital Image Processing the image matrix and the kernel are analogous to the signals and the result of the Convolution of the image matrix and the kernel is analogous to the output signal. We have explained the Convolution in detail in Chapter V.

Max Pooling is the second layer used in the model. It is an algorithm used to obtain the pixels with the maximum value. It is used to filter out the pixels with lesser significance in their value to simplify the Image Processing for the upcoming layers.

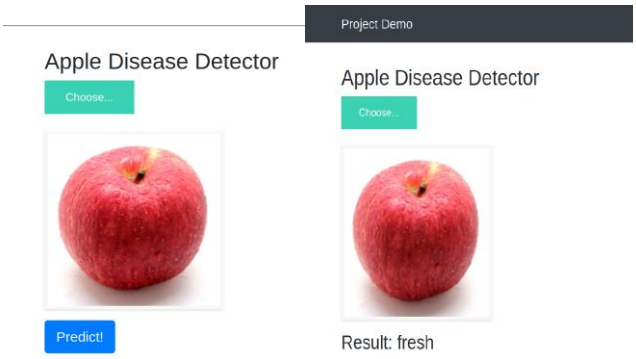
Normalization is the third layer. Normalization is a process that is used in many fields to change the range of values the parameters take. For example, in an image, if the pixels take values ranging from 0 to 255, and normalization is needed to reduce the range of values from 0 to 1 then all the pixel values are divided by 255. Normalization is generally done to simplify the implication that numbers have by reducing the range of values so that they are easily understandable. The Normalization that is done in the above model is Batch Normalization, it makes the Artificial Neural Networks faster and more stable.

The activation layer is the final general layer used in the model. Here the actual original image is divided into various pixels and we will take a kernel of 2*2 in size and place it onto the original image pixel values then the activation function will operate using the equation mentioned in section VI. Then the activation function will decide whether the output from the neuron should propagate forward or not.

There are few other layers that we have used such as Flatten and Dense. 1) Flatten: InceptionV3 model that we are using cannot support the processing of multidimensional arrays or matrices, hence we need to convert these multidimensional arrays or matrices into single-dimensional. Flatten function is used to convert the multidimensional arrays or matrices into a single dimensional array. Hence Flatten function is used for this purpose. 2) Dense: Dense layer is a set of nodes when input comes to the dense layer then it will provide input to all its nodes. The dense layer is used to put data on different dimensions. One node can draw one decision boundary and two nodes can draw two decision boundaries and so on. If the output value is greater than 0.5 then the output is true else output is false. For Example: If the input matrix is: [[1,2], [3,4]] and the kernel matrix is : [[0.5,0.75], [0.25,0.5]] with a bias of 0 with activation as linear. Hence the output matrix is nothing but the matrix multiplication of the two matrices. The output matrix is: [[1, 1.75], [2.5, 4.25]].The value 4.25 is obtained by: 3*0.75 + 4*0.5 = 4.25, which is nothing but normal 2*2 matrix multiplication.

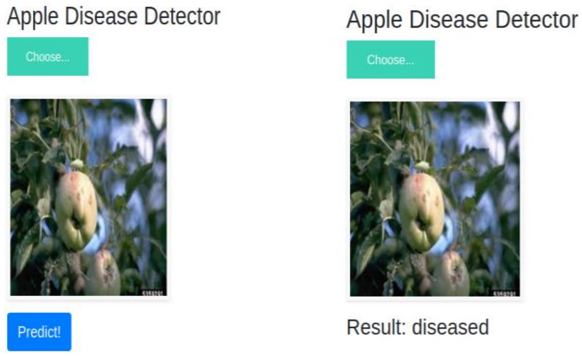
IX. Results and Discussions

We have put the software code in Jupiter Notebook (Python Programming) in Anaconda Navigator IDE. We used a local-host server to upload the images of the apples. The results are as displayed below:



Result: fresh

Fig. 3(a) Image of a fresh apple fruit is uploaded.
Fig. 3(b) Result of the image uploaded is predicted.



Result: diseased

Fig. 4(a) Image of a diseased apple fruit is uploaded.
Fig. 4(b) The result 'diseased' is predicted.

In Deep Learning projects using Python Programming Language, the model is to be trained with trainable images and the training for each image is to be done many times in iterations in order to increase the accuracy of the prediction when test images are used or in real world application of the Deep Learning project. Each training iteration is called an Epoch. More the number of Epochs, more is the accuracy but increasing the number of Epochs substantially also causes the problem of Overfitting. Hence an optimum number of Epochs need to use. I our project after a number of trails and errors we arrived at an optimum number of 30 Epochs.

X. Conclusions

This paper presents deep learning concept used to detect disease in Fruits (Apple). We were able to achieve an Accuracy of 89.70% and the Loss Value is 0.81. This paper presents the innovative methods to detect diseases in fruits and is also a prototype in implementation of the usage of modern technology in agriculture, especially in cash crops. The disease is detected in a very cheap and fast way.

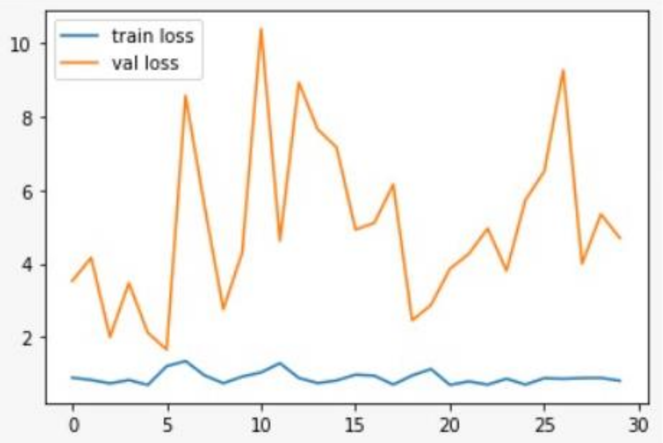


Fig. 5. Training and Validation Loss Graph

The number of Epochs used are 30. Hence the above graph Figure 5, shows the Loss Value for both Training and Validation for 30 Epochs. The graph in Figure 6 shows the Accuracy values for Training and Validation for the same number of Epochs.

Loss Value decides how exactly or incorrectly the model behaves after each iteration. Lower the Loss Value, better a model. As shown above, the loss is calculated on training and validation and it shows how well the model is doing for both the sets. It is not in percentage; it is rather a summation of the errors made for each image in training and validation sets.

Accuracy is depicted in percentage and is used to check how perfect the model is in predicting the results. We obtained an accuracy of 89.70%, which means that out of 100 random fresh apple fruit images, around 89-90 images will be predicted as 'Fresh', while the rest 10-11 images will be predicted as 'Diseased' despite being Fresh Apples. The same applies to Diseased Apple images. This is how Accuracy works.

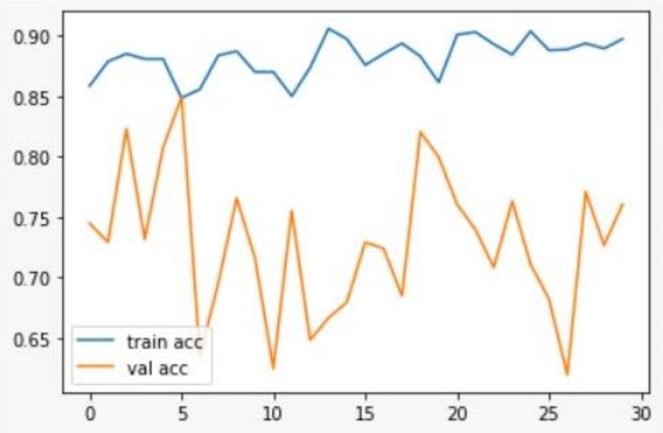


Fig. 6. Training and Validation Accuracy Graph

REFERENCES

[1] Hongjun wang, Qisong Mou, Youjun Yue, Hui Zhao: "Research on Detection Technology of various Fruit Disease Spots Based on Mask R-CNN", Tianjin University of Technology, Tianjin China, 2016.

[2] Shiv Ram Dubey, A.S Jalal: "Detection and Classification of Apple Fruit Diseases Using Complete Local Binary Pattern", December 2012.

[3] Nivedita.R. Kakade, Dnyaneswar.D. Ahire. Ahire: "Real Time Grape Leaf Disease Detection", International Journal of Advance Research and Innovative Ideas in Education, Vol-1 Issue-4, 2015.

[4] Kulkarni Anand H, Ashwin Patil RK: "Applying image processing technique to detect plant diseases". Int J Mod Eng Res, 2012.

[5] Sue Han Lee, Chee Seng Chan, Paul Wilkin, Paolo Remagnino: "Deep-plant: plant identification with Convolutional neural networks", rearXiv, 2015

[6] Halil Durmu, Ece Olcay Güne, Murvet Kirci: "Disease Detection on the Leaves of Tomato Plants by Using Deep Learning", International Journal of Computer Applications, vol. 52, no. 2, 2012.

[7] Sachin D. Khirade and A. B. Patil: "Plant disease detection using image processing," in Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on, IEEE, 2015

[8] S. B. Ullagaddi and S. Viswanadha Raju: "Disease Recognition in Mango Crop Using Modified Rotational Kernel Transform Features". 2017 Internationals.

[9] Jamil Ahmad, Bilal Jan, Haleem Farman, Wakeel Ahmad, and Atta Ullah: "Disease Detection in Plum using Convolutional Neural Network under true field conditions".28 September 2020.