SnapCrop: Android Application for Detecting Crop Diseases

Kushal P
2014B4A70624G
BITS Pilani, Goa Campus

Nikhil Ramesh
2014B5A70598G
BITS Pilani, Goa Campus

***Abstract*—SnapCrop is a mobile application developed as a part of the project requirements of the course Software Development for Portable Devices. The application uses a trained deep convolutional neural network to identify diseases in plants.**

***Keywords*—android application, convolutional neural networks, neural networks, image processing, SnapCrop,Plant diseases.**

I. Introduction

SnapCrop is an android based mobile application that enables users to identify crop diseases. The application requires the user to use the mobiles camera to click a picture of the leaf of a selected crop which it will use to identify the disease. The actual diagnosis of the disease is done on a remote server (AWS).
This application is mainly intended for farmers whose entire income depends on crop yield. Certain crop diseases are notorious for the disastrous effects they have on the yield and if the farmer is able to identify these diseases early on, he/she can take the necessary action to prevent its spread.

II. Flow of Control

Users are first directed to the home screen of the application, a screen shot of which is shown in Figure 2. Here, the user must select the type of crop, i.e., potato or tomato. Currently SnapCrop can be used to identify diseases in four plants, namely: potato, tomato, grape and corn. Once the user has selected the crop the application opens up the camera using which the user can click a picture of the leaf of the suspected crop. This image data along with the name of the selected crop is put into a JSON object which is sent to an AWS instance where a flask server is running. This script then passes the image onto a deep convolutional neural network trained on images of that particular crop to identify the disease. Once the disease has been identified, the diagnosis is returned along with some suggested cures to the application, a screen shot of which is shown in Figure 3. The entire flow is summarised in a sequence diagram shown in Figure 1.

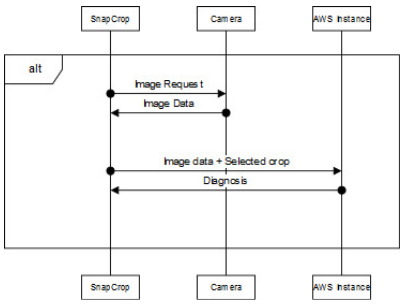


Fig 1

III. User End: Android

The entire GUI of the application has been designed in Android Studio. The application uses 4 main activities, namely: The home screen, camera activity, loading screen and an activity for displaying the final diagnosis along with cures.

Fig 2

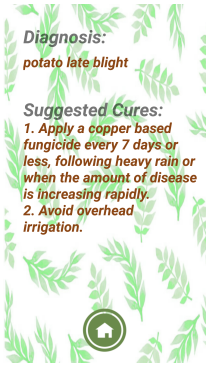


Fig 3

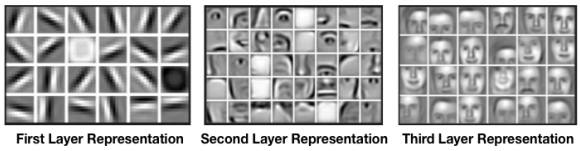
IV. Server Side

A. Convolutional Neural Networks

Image recognition involves not only the current pixels, but also the relative position of individual pixels. CNNs architecture is built on capturing different spacial features that are crucial in learning. Single pixel is merely a color and independently mean nothing, but more of them together form patterns and whole images. That is because spatial relationship of pixels is very important, especially in case of pictures. Single pixel is merely a color and independently mean nothing, but more of them together form patterns and whole images. That is because spatial relationship of pixels is very important, especially in case of pictures.

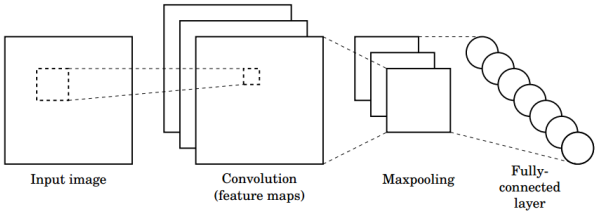
Lets say we want to detect human face from image. Simple neural network would assign every pixel to one neuron in input layer. But what does that mean? It mean that we do not keep spatial information of pixels. We split the image into individual neurons and then feed the network with them.

But in case of face recognition you have parts like eyes. Eyes are complex objects composed of several parts. You have pupil, iris, sclera and even eyelids. Every eye has them. Would you be able to detect the eye only by one pixel or only by one part? Probably not. Only the whole in specific order make sense.If you train neural network on image of eye it will only works if the eye will be on the same exact position on the image every time. When you move, scale or rotate the eye, the network will inevitably fail to predict the correct output.We need some way to look for specific patters instead of individual pixels. And that is what convolutional neural networks do.

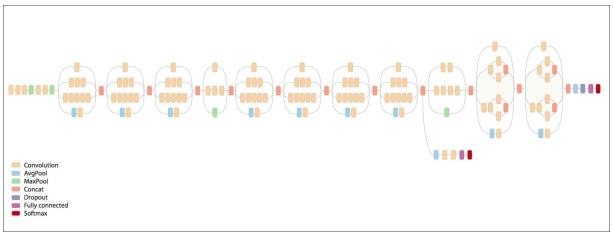


In case of image detection, the input is an image transformed into matrix of pixel values. If you have grayscale image (like MNIST) each pixels has only one value, whereas colored image has three values per pixel (RGB). During learning, convolutional network will choose appropriate kernels and all you need to do is define their amount (more filters can detect more features, but increases required time for convolution) and size (usually 3x3 or 5x5). This step is called convolution.

This is then passed on to an activation function which adds non linearity to the network. Then, the image resolution is reduced by using polling layers. And finally multiple of such groups of layers are given to a fully connected network.



1) CNN Architectures: There are many defined CNN architectures which are known to give good classification results.
Some of them are:
-AlexNet
-Resnet
-Inception V2
-Inception V3
-Inception V4
For the classification of leaf diseases ,we used Inception V3 architecture.



Inception-V3 Architectue

V. Conclusion

The first version of SnapCrop is already deployed on Play-Store. However, in order to make the application truly useful to farmers additional features like notifying neighbouring farmers of the presence of the disease are yet to be added. An application for experts is another possible extension wherein experts experienced in crop diseases can receive notifications from farmers who are not convinced with the diagnosis provided by SnapCrop.

VI. Acknowledgements

We would like to extend our gratitude to Mr Sreejith V , Instructor In-charge of the course Software Development for Portable Devices, for his guidance and support. We would also like to thank Mr Hari S R for his valuable inputs.

References

[1] Vojtech Pavlovsky, *Introduction To Convolutional Neural Networks*, 2017.

[2] Seunghoon Hong, Hyeonwoo Noh, Bohyung Han, *Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation*, NIPS, 2015.

[3] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna, *Rethinking the Inception Architecture for Computer Vision*,2015.