Disease and deficiency prediction in plants using leaf classification and color segmentation

Harishankar P
M.Tech Student,
Computer Science and Engineering
Vidya Academy of Science and Technology, Thrissur, Kerala, India

Abstract : Plants are the backbone of life on earth, as it provides us food and oxygen. Hence, a good understanding of plants is needed to help in identifying new or rare plant species. Such identification will in turn improve the drug industry, balance the ecosystem as well as the agricultural productivity and sustainability. With increasing population it has become inevitable but to increase agricultural productivity. But various diseases and conditions affecting the crops and lack of knowledge on crop management is affecting cultivation. A method to classify plants or crops and provide knowledge about cultivating the species and managing harmful conditions will be useful for increased production. Recognition of Plant from images is a challenging computer vision task. The various types of challenges are many parts of the plant, which need to be identified, are also diverse in nature with high intra class variations and small inter class variations. Object detection is one of the most important topics in digital image analysis. In object detection the system automatically locates an object from the given input image and then classify the object into one of the different available categories. Object detection has found its use in many systems like autonomous cars, video surveillance and many other applications. There are different object detection models available which are broadly classified into 2, region based and regression based. Now regression based models use CNN along with deep learning and are considered to be best suited for object detection. CNN(Convolution Neural Networks), an extension of ANN(Artificial Neural Networks, simulates the working of human neural network using a multilayer structure that incrementally extract features from the given input image from lower to higher layer until it comes across an ideal feature for pattern classification. For an object detection model to work we have two main requirements i.e. a dataset to train the model and then a good graphical processing unit(GPU).

IndexTerms - agriculture, plant classification, convolutional neural networks, deep learning, computer vision.

I. INTRODUCTION

Plants are the basis of every life on earth. Every species including mankind depend on plants for their daily survival. But now a days ecosystem is facing a huge danger in the form of deforestation and land filling as it is destroying the plants all around us. Now it may not be able to stop these atrocities completely, but there might be rare plant species which couldn't be obtained easily among those which are destroyed. If we can identify them we can protect them and its environment. An application which can classify plants from an image can be useful in this purpose as it can be used by anyone even if he/she does not have any knowledge in botanical sciences. This project could also help the agricultural field. Through mass cultivation of crops we receive our daily food items. With increase in population there arises a need for increased production. But global climatic changes and use of chemicals that affect the fertility of soil and lack of knowledge about proper crop maintenance make plants susceptible to diseases. Usually manual intervention by agricultural organizations is used in times like this to enlighten the farmers about management of diseases and deficiencies. But it is impossible to reach out manually to every irrigation area across the nation. Also the analysis will depend on observational skill of the individual. Multiple persons can have multiple opinions about a particular crop or its condition. Hence a monitoring application or network has become a necessity in agriculture. While it is possible to achieve this using camera and sensors, applying this in huge areas of cultivation across a nation is very expensive. Instead an application that takes an image of the plant and output its condition and the species it belongs to can be used for monitoring. By connecting to internet multiple opinions from the experts can be analyzed and the best option can be chosen. Object detection and color segmentation can be used for this purpose. Object detection is one of the important tasks in digital image analysis. It has found its application in many systems such as autonomous cars, security surveillance etc. Many methods were designed over the years to implement object detection. Deep learning combined with Convolutional Neural Network(CNN) is best suited for this purpose. CNN is a multilayer structure that simulates the functioning of human brain. The data extraction is performed incrementally from lower to higher layer until an ideal classification pattern is found. In my approach I am using leaves to categorize plants as they are the fingerprints of their species. Each plant has a unique leaf structure. The effect of most diseases affecting the plant species can be seen on leaves as discoloration and spots. Also any deficiency in nutrients will cause the leaves to change their color. Hence through leaves we can identify the species, disease and deficiencies. Here I am using You Only Look Once Version 2(YOLO V2) object detection model for classification training. YOLO is a regression based object detection model and one of the best models available. Color segmentation is performed using OpenCV which is a library for image processing that can be used in multiple platforms.

A database can be managed centrally or privately in order to incorporate new knowledge and trained data so improve the application over time. Thus each time someone makes any input, it can be sued by everyone else who uses the application. I used Google Colaboratory for training my dataset. It is free and efficient and can be used by those interested in or deep learning.

II. PROPOSED APPROACH

I use a pre-trained Convolutional Neural Network (CNN) model for classifying leaf species. A YOLO model can have different CNN architectures. In my approach I used Full-Yolo architecture for training. The convolution layer convolves the input layer with adjustable weight filters, namely kernels. One of the most important challenges in dealing with high-dimensional data such as images is to connect the nodes of current layer to all the nodes in the previous layer.

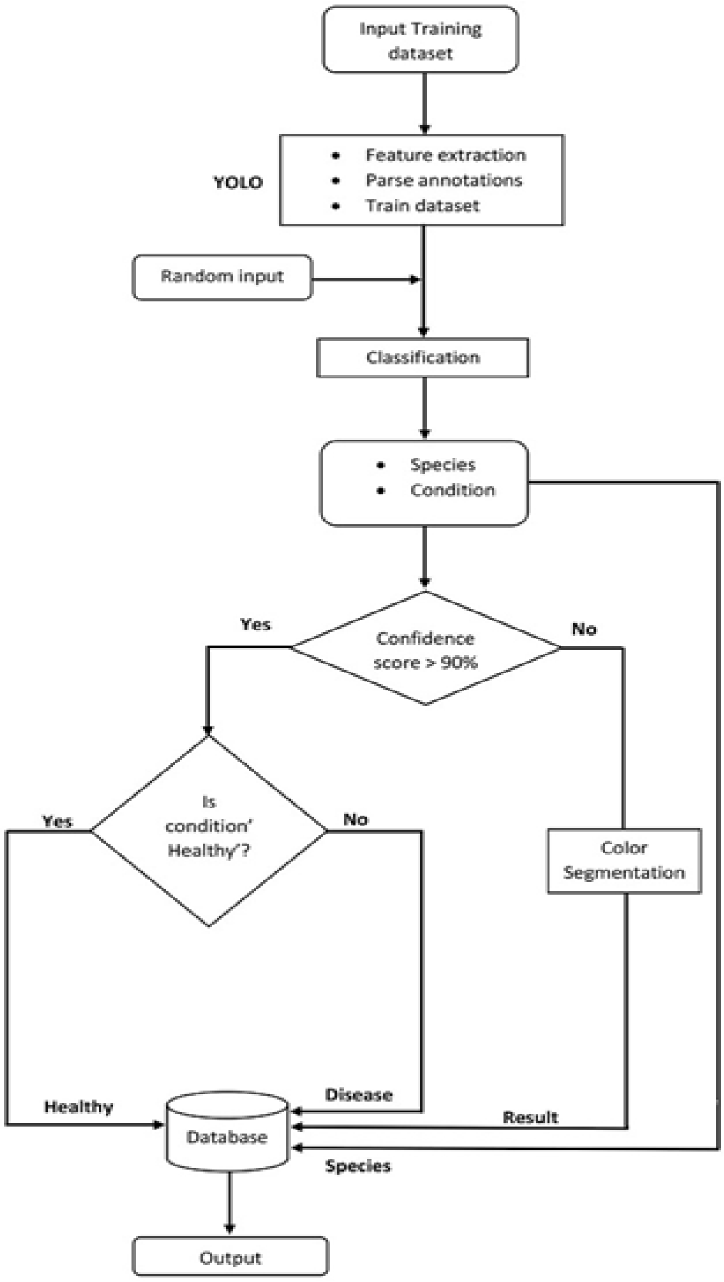

Fig1. Architecture of proposed approach

LabelImg application is used to generate annotation files for the images. Images and annotations were uploaded to Google drive in order to train using Google Colaboratory. A segmentation method was implemented for instance segmentation of the image. In this a mask is generated to identify the background and foreground of the image. The background is then turned to black. The foreground image is separated and is given a semi-transparent over layer. Nonblack pixels are identified and the over layered image is placed on it. The mask is removed from the background which leaves us a segmented leaf in the original image.

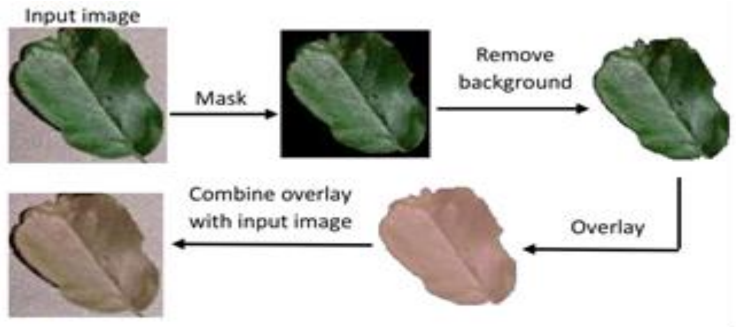

Fig2. Instance segmentation approach

A. You Only Look Once (YOLO)

YOLO is one of the fastest object detection algorithms out there. YOLO, like SSD, is also a single shot detector. It uses feature from the entire image to predict each bounding box. It simultaneously predicts all bounding boxes for an image across all classes. During training and testing YOLO sees the entire image unlike region proposal and sliding window approaches. Contextual information about classes and their appearance is encoded implicitly by YOLO. Its design enables real-time speeds and end-to-end training while maintaining high mAP.

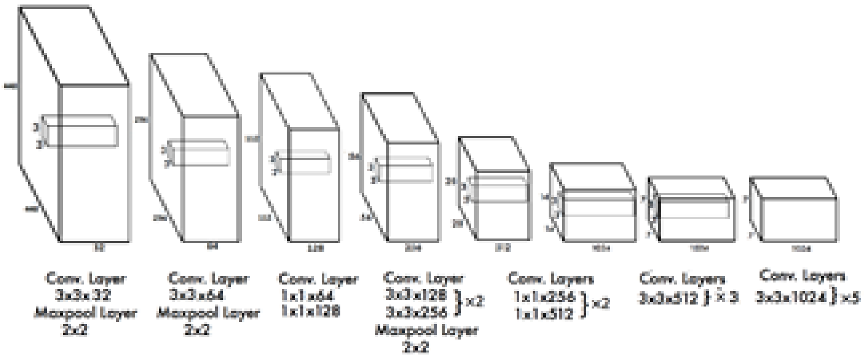

Fig3. Architecture of YOLO

A single CNN network is used for both localization and classification of the object. The image is passed once through the FCNN it outputs prediction. In YOLO each image is divided into an S x S grid and confidence and bounding boxes are predicted for each grid. The confidence refers to the accuracy of the bounding box i.e. whether the bounding box actually contains an object of any class[2]. If each grid has N bounding boxes then a total of SxSxN bounding boxes are predicted.

The confidence scores reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts[4]. Confidence scores is zero if there is no object in the cell. Otherwise the confidence score equals the intersection over union (IOU) between the ground truth and the predicted box. Confidence score is formally defined as Pr(Object) $*$ IOU.

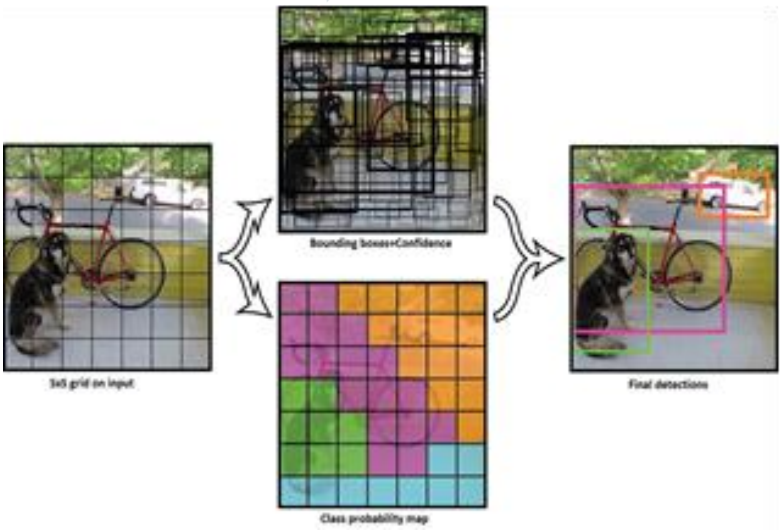

Fig4. Working of YOLO

At runtime, we have to run our image on CNN only once as it sees the complete image at once as opposed to looking at only a generated region proposals in the previous methods. Hence, YOLO is super-fast and can be run on real time. Also it will not break down easily with unexpected inputs or new domains as it generalize representations of objects. However, one limitation for YOLO is that it only predicts one type of class in one grid hence, it struggles with very small objects. YOLO does not make independent detections using multi-scale feature maps. Instead, it partially flattens features maps and concatenates it with another lower resolution maps. For example, YOLO reshapes a 28 × 28 ×512 layer to 14 × 14 × 2048. Then it concatenates with the 14 × 14 ×1024 feature maps. In order to make predictions, YOLO then applies convolution filters to the new 14 × 1 4 ×3072 layer.

Loss Function

YOLO's loss function consists of classification, localization and confidence losses. For calculating loss YOLO uses sum squared error between the predictions and the ground truth[3]. Non-maximal suppression is applied in YOLO to remove duplications having lower confidence scores.

$$\lambda_{\mathbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\mathbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

B. YOLO V2

YOLO V2 is another version of YOLO. Error analysis of YOLO shows that even though YOLO has good classification accuracy it makes a significant number of localization errors. In addition, YOLO has low recall when compared to region based models. Therefore it mainly focuses on recall and localization improvement while the accuracy of classification is maintained. Arbitrary assumptions are made by YOLO on the boundary boxes[3]. This cannot work for all objects as the boundary boxes are not arbitrary in reality and will cause steep gradient changes. For example cars have very similar shapes.

Starting with diverse guesses which are common in the case of real-life objects will make the initial training more stable. YOLO V2 uses anchor boxes instead of arbitrary boxes for prediction of the anchor offsets. This will simplify the problem and make the network training easier. By limiting the offset values, the diversity of the forecasts can be maintained and each forecast can be focused on a specific form that makes the initial training more stable. By adding batch normalization to all convolutionary layers in YOLO, we can improve mAP (mean average precision) by more than 2 percent. Classification network is first fine-tuned for 10 epochs on ImageNet at a resolution of 448 × 448 giving network time for adjusting the filters for working better on input with higher resolution. This increases mAP by nearly 4 percent. While YOLO can only predict 98 boxes for a single image YOLO V2 can predict more than one thousand using anchor boxes. YOLO receives a mAP of 69.5 mAP with 81 percent recall without anchor boxes while YOLO V2, using anchor boxes, receives mAP of 69.2 with 88% recall. Although the mAP decreases, the increase in recall makes the model more room for improvement. The modified YOLO detector runs on top of a 13 × 13 × 2048 feature map so that fine grained features can be predicted. This results in a 1% increase in performance.

C. Google Colaboratory

Colaboratory is a free research tool from Google for machine learning education and research. It’s a Jupyter notebook environment that requires no setup to use and runs entirely on cloud. All Colaboratory notebooks are stored in Google Drive and can be shared. Colab uses Tesla K80 GPU with 12GDDR5 VRAM. It has more than 2000 CUDA cores and uses a 2.3 GHz single core hyper-thread processor. It is fast and is perfect for those who wants to perform deep learning on a large dataset and cannot afford high performance systems or costly GPUs.

D. Dataset

My dataset, which downloaded from Github, contains more than 30000 images. I chose 20000 images which comprised of 26 classes including healthy and disease affected leaf images with each class having a maximum of 1000 images. The images were all set to size 256x256 which was also the size of the filter of my detection model. This considerably reduced pre-processing time as the necessity of initial resizing was removed. There were no separate validation images, instead 30% of train images were used for validation. The annotation files and images along with application files were uploaded into my Google drive for training using Colab.

III. Results and discussions

The training took approximately an hour for each epoch. I trained the dataset for 3 epochs in order to generate initial weight files. Then using the weight file as pre-trained weights. The actual training was performed. Dataset was divided into a batch of 100. There was a total of nine epochs before early stopping. For each species trained I got a confidence score above 90%. The weight file was almost 580MB. The model identifies the species and classify whether the leaf is healthy or has any of the given diseases. If the leaf is categorized as healthy and has a low confidence score it is most likely a color change. In that case color segmentation is performed to identify the deficiency. The color segmentation works fine and outputs the percentage of discoloration in leaf which is useful at estimating the amount of suppliants needed.

A database is maintained which is used to compare the color change or disease with its data and outputs the respective contents to the user. Performance is greater if the image has a plain background instead of a complex, colorful one. So in order to get maximum efficiency it will be wise to take a leaf sample and place it in a paper or a plain surface and then use the application.

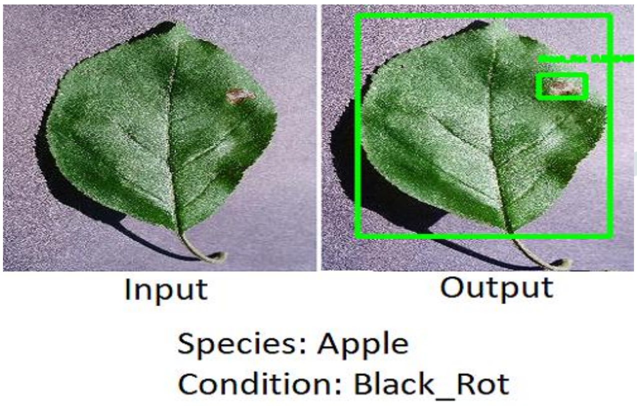

Fig5. Classification of an apple with black rot

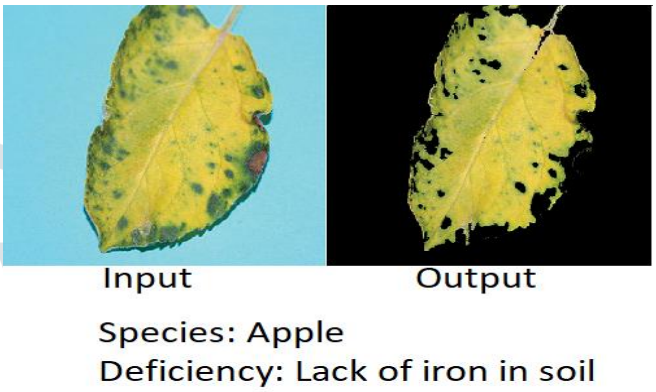

Fig6. Color segmentation of an apple leaf

IV. Conclusion

Choice of object detection model is crucial when it comes to deep learning. Different models are suited for different purposes. In my approach I am receiving commendable results while using YOLO V2. My application classifies the given leaf and identify whether it is healthy or not. It provides the user with details of the disease or deficiency and its management if the leaf is suffering from any. This application can be further improved by maintaining a central database and social network through which users and experts can share their findings and provide help without being physically present at the area. Once provided a user interface this application can be used by anyone can be used to increase quality and quantity of cultivation and thus improve the agricultural sector. Also this application could help in identifying rare species and thus prevent them from getting destroyed. Moreover this application could help people who has no prior knowledge in farming or botany get interested in these fields. Thus it could aid in improving agriculture, environment and economy of our nation.

References

[1] Hulya Yalcin, Salar Razavi, Visual Intelligence Laboratory, Electronics & Telecommunication Engineering Department, Istanbul Technical University, “Plant Classification using Convolutional Neural Networks"
[2] http://cv-tricks.com/object-detection/faster-r-cnn-yolossd/
[3] https://medium.com/@jonathan_hui/real-time-objectdetection-withyoloyolov2-28b1b93e2088
[4] https://towardsdatascience.com/yolo-you-only-lookonce-real-timeobjectdetection-explained-492dc9230006
[5] https://github.com/tzutalin/labelImg
[6] https://github.com/spMohanty/PlantVillage-Dataset
[7] https://github.com/experiencor/keras-yolo2
[8] Lei Wang, Yanning Zhang,Runping Xi, Study on Image Classification with Convolution Neural Networks, Springer Conference, October, 2015.
[9] Joseph Redmon, Santosh Divvala, Ross Girshick, YoubOnly Look Once:Unified, Real-Time Object Detection, IEEE Xplore, December, 2016.
[10] Shijian Tang, Ye Yuan, Object Detection based on Convolutional Neural Network, Stanford, 2015.
[11] https://pjreddie.com/darknet/yolo/
[12] F Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, Faster RCNN: Towards Real-Time Object Detection with Region Proposal Networks
[13] Jifeng Dai, Yi Li, Kaiming He, Jian Sun, R-FCN: Object Detection via Region-based Fully Convolutional Networks
[14] Tianmei Guo, Jiwen Dong, Henjian Li Yunxing Gao, Simple Convolutional Neural Network on Image Classification
[15] Timothy J. Jassmann, Rahman Tashakkori, and R. Mitchell Parry, Department of Computer Science Appalachian State University, Boone, NC 28608, USA, Leaf Classification Utilizing a Convolutional Neural Network

[16] Sue Han Lee, Chee Seng Chan, Paul Wilkiny Paolo Remagninoz, Centre of Image & Signal Processing, Fac. Comp. Sci. & Info. Tech., University of Malaya, Malaysia Dept. Natural Capital & Plant Health, Royal Botanic Gardens, Kew, United Kingdom, Comp. & Info. Sys., Kingston University, United Kingdom, Deep-Plant: Plant Identification With Convolutional Neural Networks

[17] Siang Thye Hang, Masaki Aono, Toyohashi University of Technology, Japan, E-mail: hang@kde.cs.tut.ac.jp, aono@tut.jp, Open World Plant Image Identification Based on Convolutional Neural Network