**Unhealty leaf detection**

**Nataša Kovačević**

**Snježana Simić**

1. Motivation

Early and accurate detection and diagnosis of plant diseases are key factors in plant production and the reduction of both qualitative and quantitative losses in crop yield. Diseases are often difficult to detect and control. Early detection of disease helps to find remedies and defense mechanisms against it. In this case, it is estimated whether the apple leaf is healthy or diseased.

2. Research questions

Detecting a healthy plant involves looking closely at its parts and the most obvious part to look at first are the leaves. Spotting the signs of a healthy plant is the first step in ensuring its growth success. The plant we chose to focus on is apple. Apple is among the most, if not the most enjoyed fruit around the world and in our region. It can be cultivated in any backyard. So, in order to help detect if their plant is growing and making progress as it should and not having any disease, our model detects whether the leaf is healthy or not.

Also, another reason for selecting apples among all the fruits is the fact that in found dataset (which is described in next paragraph), apples have, not only a lot of data collected, but almost equal amount of healthy and diseased leaves which is very crucial for the model training.

The dataset can be downloaded from this link: <https://data.mendeley.com/datasets/tywbtsjrjv/1>. PlantVillage is a not-for-profit project by Penn State University in the US and EPFL in Switzerland. They have collected - and continue to collect - tens of thousands of images of diseased and healthy crops. This dataset contains 39 different classes of the plant leaf. The dataset contains 61,486 images. The dataset contains 4 folders of apple leaves pictures. Folders Apple Scab, Black Rot, and Cedar Apple Rust contain sick leaves, while folder Healthy contains healthy leaves. There are 1526 pictures of sick leaves and 1645 pictures of healthy leaves.

Unhealthy leaves have spots, lines, discolorations, and other kinds of abnormalities.

3. Related work

This problem has been solved using Machine Learning, powered by TensorFlow Model for identifying plant diseases, like in this project [https://github HYPERLINK "https://github.com/Rishit-dagli/Greenathon-Plant-AI" HYPERLINK "https://github.com/Rishit-dagli/Greenathon-Plant-AI".com/Rishit-dagli/Greenathon-Plant-AI](https://github.com/Rishit-dagli/Greenathon-Plant-AI).

Many state-of-the-art DL models/architectures evolved after the introduction of AlexNet for image detection, segmentation, and classification. This section presents the researches done by using famous DL architectures for the identification and classification of plants’ diseases. Moreover, there are some related works in which new visualization techniques and modified/improved versions of DL architectures were introduced to achieve better results. Among all of them, the PlantVillage dataset has been used widely as it contains 54,306 images of 14 different crops having 26 plant diseases.

4. Methodology

The following diagram shows the main idea of how a problem is solved. The first part is loading the dataset and extracting features, and the second part is the training of the classificator on extracted features.

**Image preprocessing**

From 1645 pictures of healthy leaves, 1640 are processed, and similarly, from every other folder, 5 to 6 images are ignored.

The image contains a lot of information, only some of these information can be used to distinguish between different situations. so much of the information in the image must be converted to reduced representation set called the features process that extracts features from the image called features extraction.

The image has many features such as texture, color, and shape. These features can be used as mini-information representing the useful information in the image which can be used to distinguish between different situations.

Loaded pictures are converted to RGB and blurred. Removing the background is done by Adaptive Thresholding and closing. Color features that were important for selecting abnormalities and discoloration are red, blue, and green colors. Important texture features are entropy, inverse difference moment, contrast, and correlation. Entropy measures the randomness of the intensity image. Correlation shows how correlated a pixel is to its neighborhood. Inverse Difference Moment measures the texture homogeneity.

All data are written in csv.



**Classifier**

The classifier used to train our model is SVM classifier.   
First, we load the feature csv generated in the first stage, preprocessing leaves. After it is read, date set is shuffled, so that we have both healthy and diseased leaves split into training, validation and test samples in the most equal way possible. After that, we fill our y values based on class ‘isHealthy’ from the dataset, giving value 1 to the diseased leaves and 0 to the healthy.  
After, we split data to training, validation and test samples in measurements 70:10:20 respectively. Training sample is used to train our model, validation is used for tuning in the hyper parameters of SVM classifier and test is used to evaluate the accuracy of the final model. This concludes the first stage.

Second stage starts with scaling out samples as we want all of data to be in the same interval values. Therefore, we determined mean and standard deviation of our training set and then scaled all samples using those values. The most important part of second stage comes next. It is training our model and determining under which hyper parameters SVM classifier gives the best result, aka the best accuracy. We tried determining by assuming which parameters give best result, but we soon realized that it might take too much time. So we instead, relied on GridSearchCV from sklearn library. GridSearchCV does exhaustive search over specified parameter values for an estimator

Third and final stage was evaluating our model on test sample. We also imported several pictures that we had no data collected before and only one was incorrectly predicted.

5. Discussion

Removing background with findContours did not work well. For some pictures, it removed the background perfectly, like for this one



But for some other pictures results were very bad.

 

So, we just did thresholding and closing, and that gave good results.

During the training of our classifier, after we split all the data, we first tried to input hyper parameters ourselves to see how closely we could guess the best combination. We tried linear kernel with slack penalty C = 1 and we got validation accuracy 86,667%, then we increased C to a 100 and got the same accuracy. We then concluded that maybe it is best to not use linear SVM and tried rbf kernel with C=10 and gamma=0.001 and our validation accuracy was 89,5238%. Although, these are already good results, we decided to implement GridSearchCV to see if we could get even better results. The combination we tried were :



Best parameters were determined as: C= 100, gamma = 0.01 and kernel was rbf. Validation accuracy with these parameters was 96,19%, so we decided that these are best for our model.  
  
 In the final stage we evaluated our model on test sample and got accuracy of 95,079%. We also imported 21 images that had no previously extracted data and only one was incorrectly predicted.

6. References

[1] <https://arxiv.org/abs/1511.08060> - An open access repository of images on plant health to enable the development of mobile disease diagnostics (Must cite this paper if plant village dataset is used)