Pattern Recognition Project Report

Title: Support Vector Machines for classifying plant species from the images of it's leaves

Introduction:

The most popular method for classification of images is Convolutional Neural Network, it is an efficient algorithm compared to neural networks for image data. But in this project we demonstrate how we can classify images using SVM after feature extraction and comparison of the results for various kernels.

Dataset Description:

In this project, we have collected HD leaf images of 10 different plant and tree species, also including the subclass, Diseased and Healthy leaves for each plant species.

We use SVM after feature extraction to classify 10 different plant species and in turn classify them as diseased or healthy.

- Total number of leaf Images in the dataset: **4236**
- After feature extraction, number of attributes obtained: **68**
- Total number of classes in target vector 1: **10**
- Total number of classes in target vector 2: **2**

  Here, note that we have 2 target vectors, 1 is for plant species classification and the other target vector is to classify the health condition of the plant species.

Methodology:

- Firstly, we have downloaded the plant leaf dataset from the link . The data set contains images of different kinds of leaves like mango, guava, jatropha, etc. and further they are classified into diseased and healthy.
- Images provided in the dataset are of very high resolution (6000 x 4000), due to which the feature extraction becomes computationally expensive. So, we have reduced

resolution of images to 720 x 480 maintaining the same aspect ratio using image resizing library from OpenCV Python.

Healthy Arjuna Tree Leaf

Diseased Arjuna Tree Leaf

Healthy Chinar Tree Leaf

Diseased Chinar Tree Leaf

- Perform feature extraction on resized images . For feature extraction and image processing OpenCV and Mahotas libraries are used. Mahotas is an open source library for identifying texture features of images.

- After feature extraction all the features are saved into a .csv file along with the class labels. Now, let us see how the feature extraction is performed in detail,

  Feature extraction:

  - The feature extraction is an important engineering process, for it is the main information about the raw data that the algorithm identifies.
  - Selecting the most meaningful features is a crucial step in the process of classification problems.

- For feature extraction, we have used inbuilt functions under OpenCV, mahotas and sklearn libraries. Here, the mahotas library is used to extract texture features from the images.
- Next, we have converted the images into gray scale and calculated Hu Moments. This can be achieved using HuMoments in-built function under cv2 library. Hu Moments (or rather Hu moment invariants) are a set of 7 numbers calculated using central moments that are invariant to image transformations.
- Based on Hu moments of two images we can compare whether they are identical or not, irrespective of the image transformations like translation, scaling, rotation and reflection.
- Finally, from the RGB histograms of each image we have taken histogram values from 16 bins from each of the R, G and B components of the image which will result in a total of 48 attributes in this step.
- ***So, total number of features extracted after feature extraction:***
  7(Hu Moments) + 13(Haralick texture features) + 16*3 = 68

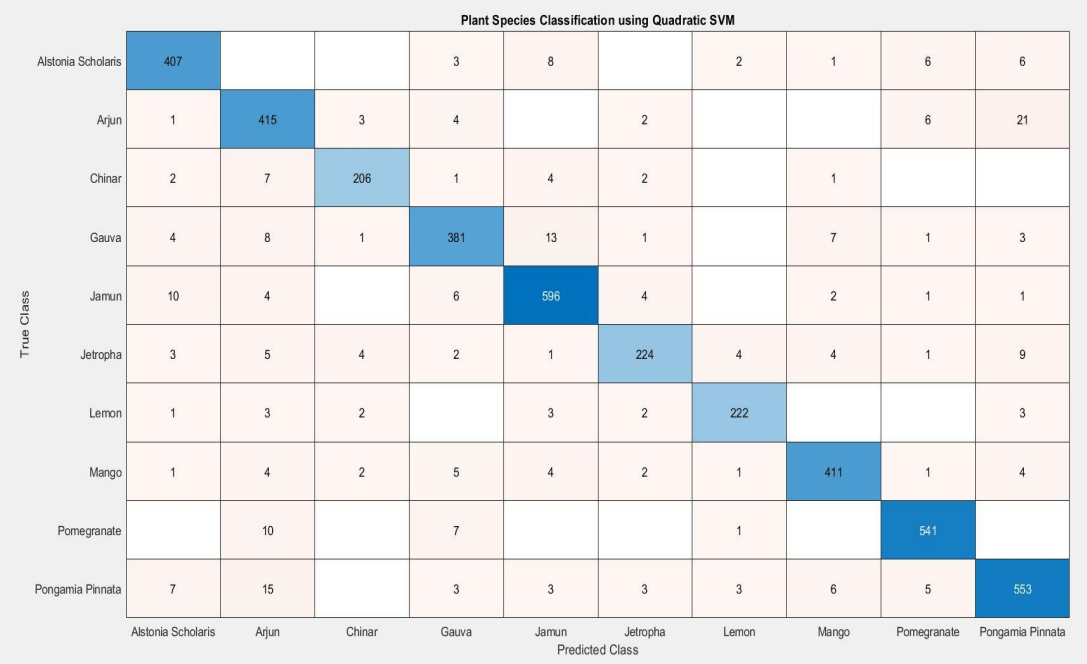
- A python code is written, which on execution, iterates through all the images in the dataset and performs feature extraction, labelling the instances and saves the overall data into a '.csv' file. This file is later read in a matlab file to train the classifiers.
- For training the SVM classifier in the MATLAB, classification learner app is used. The Steps involved here are:
  - Firstly, load the features data from .csv file to MATLAB workspace
  - Create a session in the classification learner app for training different classifiers and then import predictors data and target vectors(class labels) from the workspace.
  - Finally, Train SVM with different kernel functions and the results have been compared.

Results:

Plant Species Classification :

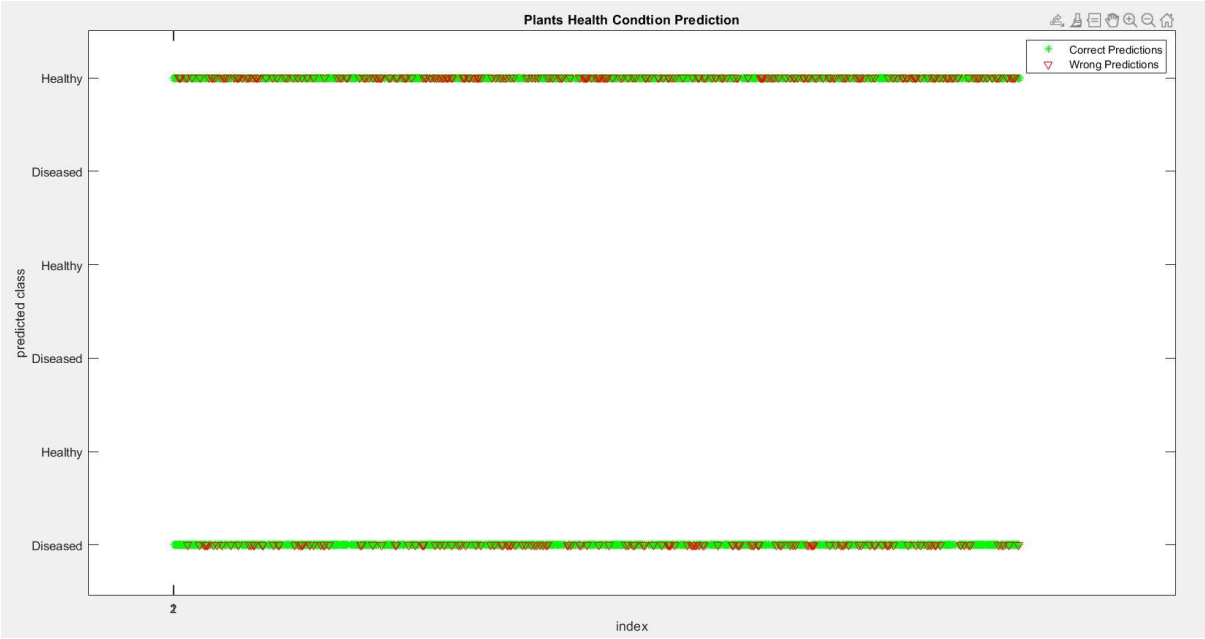
1 ☆ SVM	Accuracy: 89.2%
Last change: Linear SVM	68/68 features
2 ☆ SVM	Accuracy: **93.4%**
Last change: Quadratic SVM	68/68 features
3 ☆ SVM	Accuracy: 88.6%
Last change: 'Kernel function' = 'Gaussian'	68/68 features
4 ☆ Linear Discriminant	Accuracy: 85.4%
Last change: Linear Discriminant	68/68 features

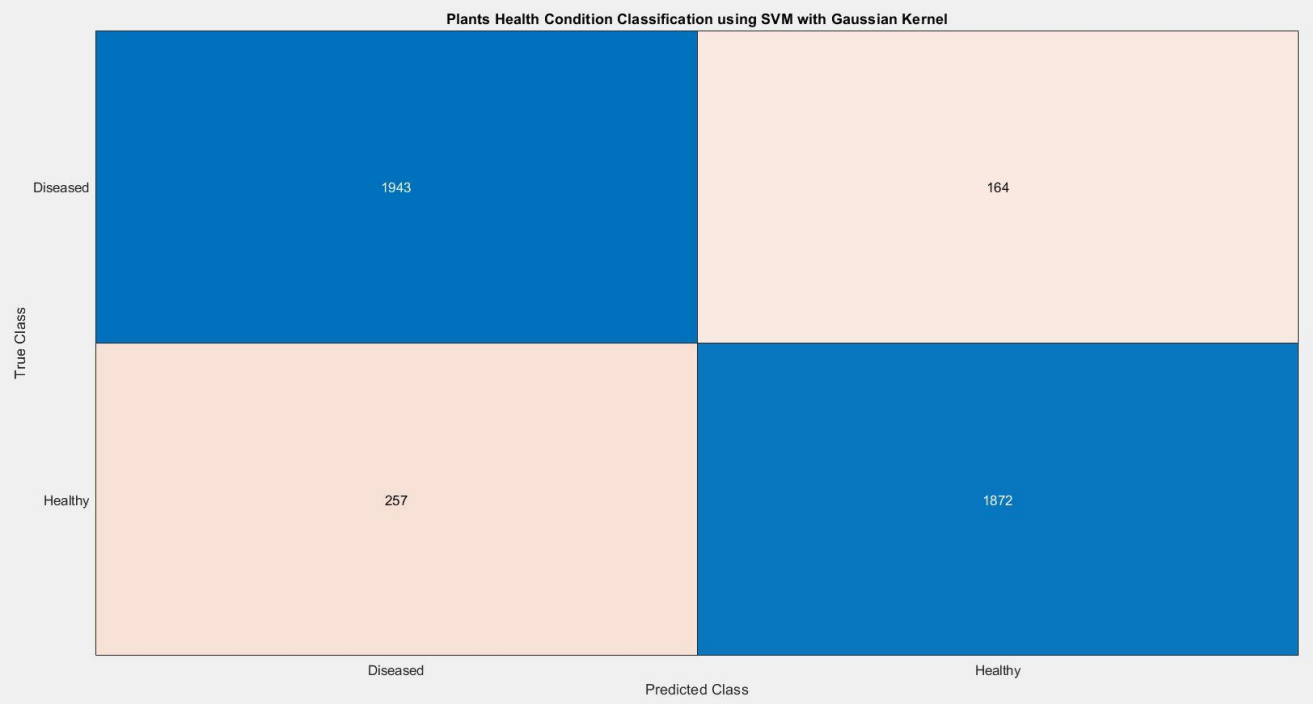
The training data for plant species classification is trained and tested on four types of classifiers 1) Linear SVM, 2) Quadratic SVM, 3) SVM with Gaussian Kernel and 4) Linear Discriminant Classifier. The Quadratic SVM performed the best among the four with an accuracy of 93.4%.



Plant Health Condition Classification :

The training data for the plant's health condition classification is trained and tested on three types of classifiers 1) Linear SVM, 2) Quadratic SVM and 3) SVM with Gaussian Kernel. The SVM with Gaussian Kernel performed the best with an accuracy of 89.8%.






```
Confusion Matrix for Plant Species Class Prediction
   407     0     0     3     8     0     2     1     6     6
     1   415     3     4     0     2     0     0     6    21
     2     7   206     1     4     2     0     1     0     0
     4     8     1   381    13     1     0     7     1     3
    10     4     0     6   596     4     0     2     1     1
     3     5     4     2     1   224     4     4     1     9
     1     3     2     0     3     2   222     0     0     3
     1     4     2     5     4     2     1   411     1     4
     0    10     0     7     0     0     1     0   541     0
     7    15     0     3     3     3     3     6     5   553

Accuracy for Plant Species Class
    0.9339

Confusion Matrix for Plant Species Class
        1922         185
         264        1865

Accuracy for Plants Condition
    0.8940
```

Conclusion:

From the above observations, we can say that plant species can be classified using SVM by having the appropriate features recognized and extracted from their images. This

reduces the complexity of the problem significantly as compared to the usage of Deep Neural Networks and other Image Classifiers.