SLIC_SVM based leaf diseases saliency map extraction of tea plant

Team:
16CO121 - Namrata Ladda
16CO131 - Polkampally Keerthana

Submitted to:
Dr. Venkatesan
CSE Dept , NITK

Date : 21-11-2019

Objectives

- To improve the performance of feature extraction tea plant leaves under complex backgrounds in terms of visual effects.
- To improve the performance of classification algorithm

Literature Survey

- Extraction of diseased plants features
  - Edge discontinuity, edge information loss or edge blur etc.
- Region-based segmentation algorithm
  - Pixel color similarity in the same area
  - Target area is continuously merged with similar pixels.
- SLIC algorithm with Markov's chains
  - Improved detection accuracy and effectively
- Region-based color modelling for joint crop and maize tassel segmentation
  - Graph-based segmentation algorithms
  - Robustness to illumination

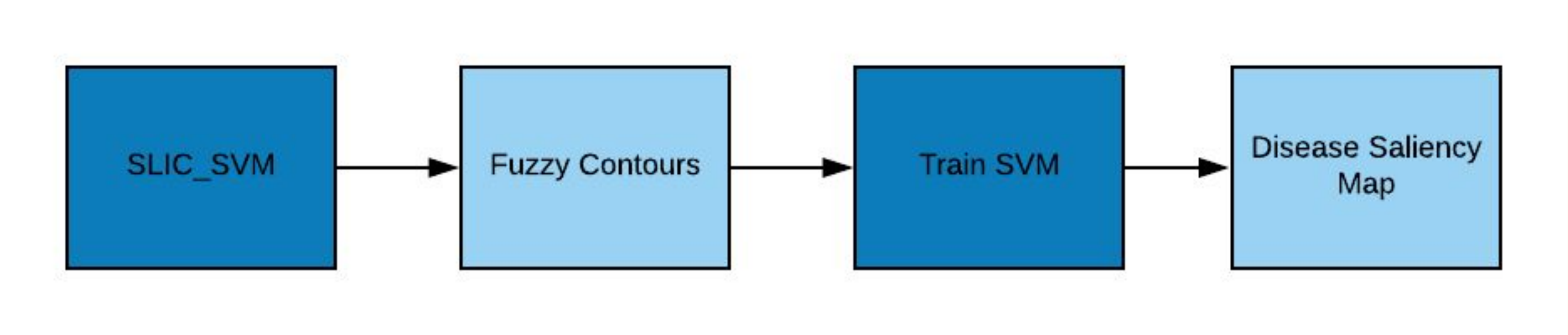
Introduction

- An algorithm combining SLIC (Simple Linear Iterative Cluster) with SVM (Support Vector Machine) is proposed in this paper.
  - Firstly, super-pixel block is obtained by SLIC algorithm, significant point is detected by Harris algorithm, and fuzzy salient region contour is extracted by employing convex hull method.
  - Secondly, the four-dimensional texture features of super-pixel blocks in salient regions and background areas are extracted
  - The classification map is obtained by classifying the super-pixel blocks with the help of SVM classifier.

DATASET

- The dataset is obtained from "A low shot learning method for tea leaf disease identification" paper.
- The dataset consists of diseased tea leaf images which has 3 types of diseases
    - Leaf blight
    - Red leaf spot
    - Red scab
- The dataset consists of both test and train data which contains about 20 images of each disease. It constitutes about 60 images of train data and 60 images of test data.

Flow of the Algorithm



IMPLEMENTATION

- **DATA PREPROCESSING :** The dataset folders of both train and test data are converted into csv format.
- **TRAINING ALGORITHM :**
  - **SLIC ALGORITHM :** Superpixel segmentation is obtained from implementation of slic algorithm on train data.
  - **HARRIS DETECTION :** Significant points from the superpixel segmented image were detected by Harris algorithm.
  - **CONVEX HULL :** Fuzzy contour of images is extracted using the convex hull technique.
  - Identification of the features to distinguish salient region and background super-pixel blocks and extract it by GLCM function.
  - The texture features (energy, entropy, contrast and correlation) of super-pixel blocks are used to train Support Vector Machine (SVM).

SLIC ALGORITHM

A superpixel can be defined as a group of pixels which have similar characteristics. It is generally color based segmentation. Superpixels can be very helpful for image segmentation.

SLIC is the state of the art algorithm to segment superpixels which doesn’t require much computational power. In brief, the algorithm clusters pixels in the combined five-dimensional color and image plane space to efficiently generate compact, nearly uniform superpixels.

- It has a different distance measurement which enables compactness and regularity in the superpixel shapes, and can be used on grayscale images as well as color images.
- A 5 dimensional **[labxy]** space is used for clustering. CIELAB color space is considered as perpetually uniform for small color distances.

HARRIS ALGORITHM

STEP 1. It determines which windows (small image patches) produce very large variations in intensity when moved in both X and Y directions (i.e. gradients).

STEP 2. With each such window found, a score R is computed.

STEP 3. After applying a threshold to this score, important corners are selected & marked.

Convex Hull Algorithm

Step1: Find the leftmost of significant point that has smallest y, find the significant point with smallest x instead if there are more than one smallest y;

Step2: Set P0 as the origin of coordinate by coordinate translation;

Step3: Argument α is calculated by decreasing order.

If there are equal α values, put those closer to P0 in the front.

Convex Hull Algorithm

Step4: Create a stack with P0 and P1 in bottom, and then push [P0, P1, , P(n-1)] into the stack in order. If the two points on the top of the stack are not formed a 'turn to the left' relationship with P0, push to stack until no point needs to be popped after the current point into the stack.

Step5: After completing the process of all the points, the points within stack are exactly the salient points on convex hull, and these points are also the points on fuzzy contour of salient region.

Feature extraction

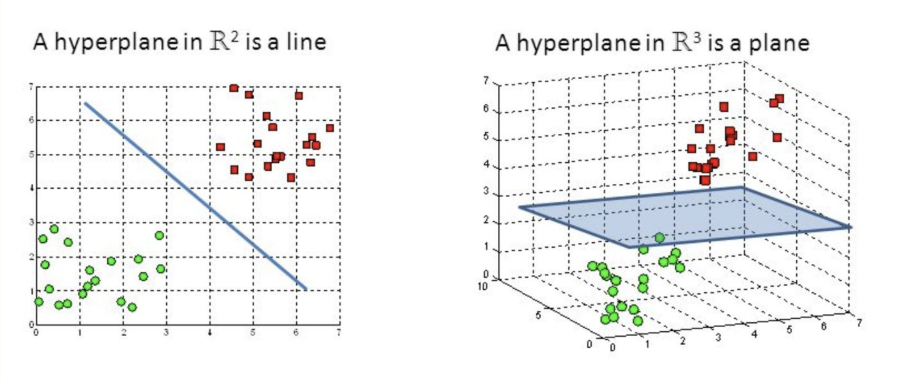
Step1: Gray Level Co-occurrence Matrix (GLCM) algorithm extracts 4-dimensional texture features of each super-pixel block, and then defines super-pixel block labels to form super-pixel block feature vectors.

Step2: Salient region and background training sample blocks automatically is selected based on pixel block serial numbers, then they are fed into SVM architecture for conducting model training.

- The calculation formula of polynomial kernel function is K (x , x1) = ((gamma)x^T xi + r) d
- The final classification image is output by considering the classification on -1 or 1.

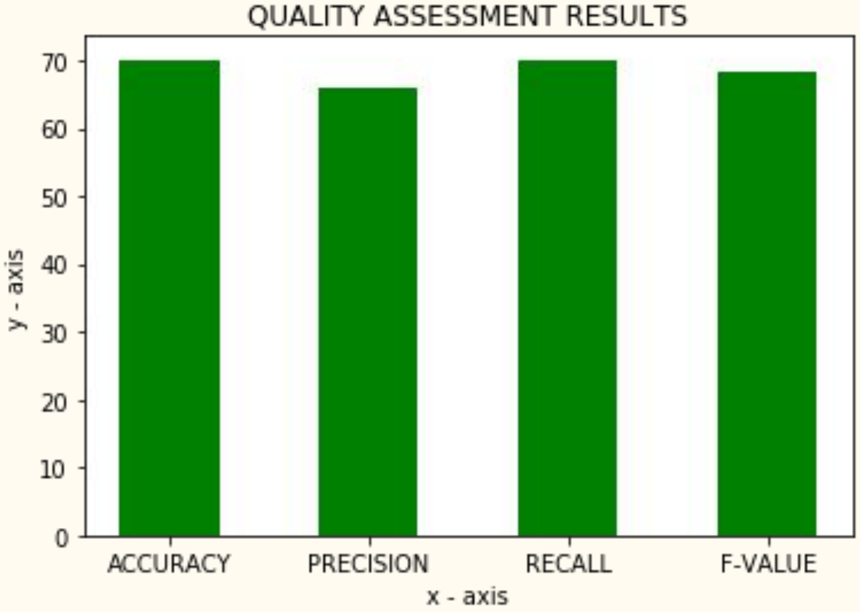
SVM ALGORITHM

- The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.
- Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features.



IMPLEMENTATION DETAILS

- Accuracy = (tp+tn)/(p+n)
- Precision score = (tp)/(tp+fp)
- Recall score = (tp)/(tp+fn)
- F-value = (2*recall+precision)
  /(recall+precision)
- Quality assessment results contains
  - Accuracy
  - Precision score
  - Recall score
  - F-value



IMPROVEMENTS

- Decision Tree using gini index is used as the classification algorithm.
- The accuracy percentage is improved by 2%.
- Improvised SLIC

```
In [95]: from sklearn.tree import DecisionTreeClassifier
         from sklearn.metrics import accuracy_score
         from sklearn.metrics import classification_report
         clf_gini = train_using_gini(x_train, x_test, y_train)
         clf_entropy = tarin_using_entropy(x_train,x_test, y_train)

In [96]: y_pred_gini = prediction(x_test, clf_gini)
         cal_accuracy(y_test, y_pred_gini)

         Accuracy :  72.0
```

THANK YOU