Leaf Type Identification Using Computer Vision

Author: Nikola Marin[1], Mentor: Peter Rogelj

***Abstract*— This project comes as part of Computer Vision subject at Computer Science postgraduate studies. The main goal of this project is to make software which input is leaf image from which software recognizes the leaf type. This project put in use vast majority of theoretical knowledge learned on Computer Vision subject.**

I. INTRODUCTION

An increased understanding of leaf is important in a number of fields: in food and non-food crops, for example short rotation forestry as a bio-fuels feed-stock, leaf area is intricately linked to biomass productivity; in paleontology leaf shape characteristics are used to reconstruct paleoclimate history. Such fields require measurement of large collections of leaves, with resulting conclusions being highly influenced by the accuracy of the measurement process. This project developed a new tool for automated leaf type detection. The software as a prototype uses data-set of 29 images for comparison with input image and as it is capable of detecting four leaf types (apple, chestnut, elm, willow and wine plant). Next chapters will show main techniques used to build this this prototype which provides an efficient automatic leaf detection.

II. IMPLEMENTATION

This project has been implemented as MATLAB script. It is based on mexopencv library which is development kit of MATLAB MEX functions for OpenCV library. The package provides MATLAB MEX functions that interface with hundreds of OpenCV APIs. Also the package contains a C++ class that converts between MATLAB's native data type and OpenCV data types. The version of mexopencv used in development of this prototype is compatible with OpenCV 3.4.1. This software is used to identify object as leaves in image and to calculate properties of those objects in an automated fashion. Automated analysis requires no user intervention after setting the desired input image. Images in a data-set are named by leaf type (wine-1.jpg, wine-2.jpg elm-1.jpg, elm-2.jpg etc.). The software computes leaf contours, and it compares contours of input leaf image to contours of all leaf images from data-set. For each comparison of contours software calculates Euclidean distance and at the end it writes out the name of leaf image from data-set with smallest Euclidean distance compared to the input leaf image. Since this software is still in prototype phase it prints out euclidean distance on every iteration (for every image from data-set) since this data are giving a lot of information to developer and it will be used while further development.

The computational process involved can be described in the following steps.

*This work was not supported by any organization

[1]Author: Nikola Marin is with Faculty of Natural Sciences, Mathematics and Information Technologies, Glagoljaka 8, SI-6000 Koper, Slovenia marindzoni@gmail.com

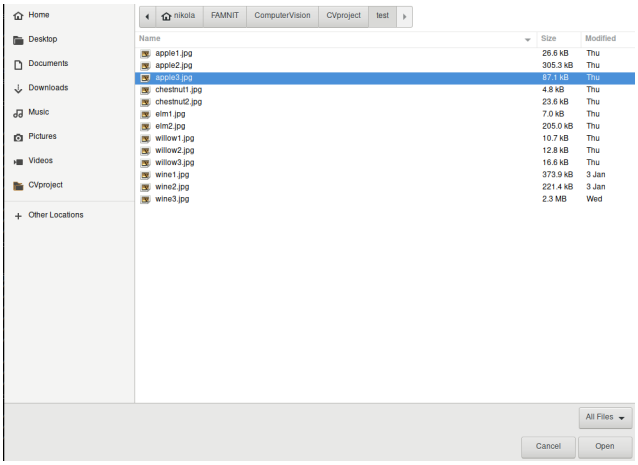

Fig. 1. Input image choosing window

```
File = images/willow-3.jpg
D =  37.213
File = images/willow-4.jpg
D =  38.397
File = images/willow-5.jpg
D =  30.093
File = images/wine-1.jpg
D =  30.161
File = images/wine-2.jpg
D =  28.801
File = images/wine-3.jpg
D =  44.996
File = images/wine-4.jpg
D =  45.652
File = images/wine-5.jpg
D =  47.479
File = images/wine-6.jpg
D =  26.562
File = images/wine-7.jpg
D =  9.9053
File = images/wine-8.jpg
D =  29.876
File = images/wine-9.jpg
D =  26.156
-------------------------------------------
matching_file =
{
  [1,1] = apple-4.jpg
```

Fig. 2. Results

III. MAIN COMPUTATIONAL STEPS

A. 2D convolution (Image Filtering)

As in one-dimensional signals, images also can be filtered with various low-pass filters(LPF), high-pass filters(HPF) etc.

LPF helps in removing noises, bluring the images etc. HPF filters helps in finding edges in the images. Image bluring is achieved by convolving the image with a low-pass filter kernel .It is useful for removing noises. It actually removes high frequency content from the image. So edges are blurred a little bit in this operation. In this project images are filtered with median blur filter. Median blur filter takes median of all the pixels under kernel area and central element is replaced with this median value. This is highly effective against $salt-and-pepper$ noise in the images.

B. Thresholding

Posterior to filtering thresholding is performed to find candidate picture elements(pixels) that represent leaf. In the thresholding process, all pixel intensities are reduced from the typical grayscale range of 0-255 to either 0 (off ;pixel is background) or 1(on; pixel potentially belongs to a leaf object). As input, the blue channel intensities are used rather than the entire RGB image. The rationale behind this strategy is that while leave can be green, orange, red or even black they are very rarely blue. On a white background, non-blue objects can, with high accuracy, be distinguished from the background using global thresholding [1]. The optimal threshold value is choosen by the Otsu algorithm.

C. Morphological Transformations

Since leaf cracks can produce "false negatives" while tresholding, morphological transformations are performed to remove noise and presence of contaminants in the images (false negatives). Morphological transformations are some simple operations based on the image shape. It is normally performed in binary images. Morphological transformations can remove noise from binary image but also they can damage contours of object on the image. Since in this project leaf recognition is hardly based on leaf contours and it is very important to preserve original leaf contours. Consequently morphological transformation "Opening" is iterated once on the image with kernel size 3x3 and it fixes only small scratches and cracks.

D. Shape matching using Hu Moments

Image moments are a weighted average of image pixel intensities. For simplicity, let consider a single channel binary image I. The pixel intensity at location (x, y) is given by $I(x, y)$. For two shapes to be the same, the image moments will necessarily be the same, but it is not a sufficient condition. We would like to calculate moments that are invariant to translation(move in x or y direction), scale and rotations. We can in fact calculate such moments and they are called Hu Moments [2]. OpenCV has a built-in function for calculating Hu Moments which returns seven Hu Moments. As input it takes the central moments of the image which can be calculated using also OpenCV function called moments. Since obtained Hu Moments have a large range there is possibility that not all seven Hu moments will be comparable in magnitude (e.i. $hu[0]$ is not comparable in magnitude as $hu[6]$). Therefore log transform (1) given below is used to bring them in the same range.

$$H_i = -sign(h_i) \times log|hi| \tag{1}$$

After above transformation(1), the Hu Moments are of comparable scale. As mentioned earlier, all seven Hu Moments are invariant under translations, scale and rotation. If one shape is the mirror image of the other, the seventh Hu Moment flips in sign. Using Hu Moments we can find the distance between two shapes. If the distance is small, the shapes are close in appearance and if the distance is large, the shapes are farther apart in appearance. OpenCV provides an easy to use utility function called matchShapes that takes in two thresholded images or contours and finds the distance between them using Hu Moments. Since this project is intended to show students understanding of material, instead of using OpenCV function matchShapes distance measure between two shapes is calculated as Euclidean distance between Hu Moments given by

$$D(A, B) = \sqrt{\sum_{i=0}^{6}(H_i^B - H_i^A)^2} \tag{2}$$

IV. Results

In order to test this program and to provide some results for further development and experimental design, program is tested with 13 input images and at least with two different leaf input images for every leaf type. All test input images are at least 10% different from images in data-set. After the final testing on all 13 input images program gave correct results for 10 input images (76%).

V. Conclusion

The first and most important fact is that this project applied theoretical knowledge into practical use and it gave base for further learning and development of projects related to computer vision. Testing showed that this program in most cases works correctly. In further development problem with incorrect results should be solved by building larger data-set of various leaf images but also algorithm should be improved to more complex level.

References

[1] Max Bylesj, Vincent Segura, Raju Y Soolanayakanahally, Anne M Rae, Johan Trygg, Petter Gustafsson, Stefan Jansson and Nathaniel R Street, LAMINA: a tool for rapid quantification of leaf size and shape parameters, BMC Plant Biology July 2008.

[2] Satya Mallick, Shape Matching using Hu Moments, 10 December 2018.

[3] Abid Rahman K, OpenCV-Python, June 2012.

[4] Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited, 2011.

[5] Dwayne Phillips, Image Processing in C, Second Edition, R & D Publications, 1994.