INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Mechanical Engineering

ME395 : ENGINEERING PRACTICUM

End Semester Report

Project Name : Image Processing Software

Team Members :

Madhu Sri Kiran Batchu	-	1901ME15
Meenakshi Charanthu	-	1901ME17
Ganesh Kumar	-	1901ME23
Manoj Kumar Madugula	-	1901ME39
Vijay Mudavath	-	1901ME45

Mentors :

Dr. Somnath Sarangi

Dr. Murshid Imam

Title:

Software to measure dimensions of the selected objects in a given input image

Problem Statement:

Creating an Image processing interface using python tools and an interactive web interface.

Introduction:

In this project we would like to present to you an Image processing interface which is made using the python tools and modules and a Web interface , in which we give our input image file and with a specified scale we select the boundary of the object we wish to measure on the image. Now after the image runs through the code gives us the required output of measurements such as Area , perimeter, Major axis length and Minor axis length.

How does this work ?

This software contains several components each has a task to do.

- First the Web interface:
    - This is a web interface made using HTML , CSS , Bootstrap to give a fresh feel to the user. This is where the user will give the desired input image which is accessed from the system.
    - The graphical interface provides a freehand tool to select the boundary of the desired object from the image.
    - Next is specifying the scale. The image processing software measures the images in pixels . (Every image is a matrix of pixels)
    - Then the image is returned to part of the software which contains the Image processing code.
- A digital image is a two- or three-dimensional matrix, and can be classified to: Binary, Gray, or RGB colour images.For example an RGB image is an 3 Dimensional matrix which has 3 independent image planes whose primary colours are Red , Green and Blue respectively stacked on top of each other, basically a $(m \times n \times 3)$ matrix. The matrix element is called a pixel. Each pixel data is just numbers representing the RGB data value. Each pixel data value has 3 components showing the combination of intensities stored in each RBG plane. Each component can have 256 values ranging between 0-256.
- But we convert the RGB image to a Grayscale then Binary image. Now the pixel can either have a value of 1 -when it's part of the pattern- , or 0 -when it's part of the background- i.e. there is no gray scale level. (We will assume that pixels with value 1 are black while zero valued pixels are white). In order to identify objects in a digital pattern, we need to locate groups of black pixels that are "connected"

to each other. In other words, the objects in a given digital pattern are the connected components of that pattern.

- Our second component Image processing code made of python:
    - This part imports some python modules and libraries like Opencv for image processing and Matplotlib for plotting the output and measuring the pixels' intensity.
    - Using these , the input given image is read and converted to an RGB valued matrix of pixels.
    - Then they are identified or filtered based on the threshold intensity of the RGB components.
    - But to measure the area inside the selected boundary , we have to convert the RGB to Grayscale Black and white then to Binary
- Our third part Measuring the area inside the selected boundary:
    - The Binary turned image now has each pixel with value 1 or 0, where 1 is object and 0 is background.
    - Now among the pixels inside the selected boundary , we have to find the pixels with value 1 and similar pixels connected to it , which gives the pixels covered by the object.
    - Calculating the number of pixels along the selected boundary gives us the perimeter of the object.
    - Maximum number of pixels calculated in a column or a row gives the Length of major axis while minimum number of pixels calculated gives the length of minor axis.
    - But what we receive is the output in pixels but we need the output in the metric system.
- Our fourth part converting the pixels into metric system:
    - We have specified a scale (from pixels to mm or cm) in the user interface which will be used and equated for all the pixels , that gives the result in the required metric system.

Explaining the resources used till now:

- Frontend :
    - HTML
    - CSS
    - Bootstrap
    - Javascript

- IMAGE PROCESSING:

- Opencv: This is one of the most famous and widely used open-source python libraries for computer vision tasks such as image processing, object detection etc. It has more than 2500 algorithms, which are helpful in performing the required tasks. Some of them are:
  - imread() : this function reads any type of image such as jpg, jpeg, png and also gifs too.
  - cvtColor() : this is used to convert the given images color components.
  - color_BGR2RGB : this converts the given color image BGR plane format to RGB plane format which is useful for standard calculation of threshold intensities to be filtered.
  - COLOR_BGR2GRAY : this converts the image from RGB to grayscale.
  - Threshold : threshold is used to classify the pixel values, it means that it selects the pixels with intensity exceeding the given threshold value.
  - THRESH_BINARY : this is used to convert the pixels into binary by making values of pixels with intensity greater than given threshold to 1 and less than given threshold to 0. This is used often when selecting the pixels at the boundary of the selected area.
  - Adaptive_thresh_mean_c: this is used to convert the pixels into binary but the threshold is taken as the mean of the intensities of pixels in all the neighbourhood.
  - Adaptive_thresh_gaussian_c: this is used to convert the pixels into binary but threshold is taken as a gaussian weighted sum of intensity of pixels in neighbourhood.
  - Plt : this is used to plot the output image
  - Imshow() : this is used to show the plot of the output

Tasks completed in this Semester :

- Decided the topic of the Project - Software to measure dimensions of objects from a given image.
- Started learning Python and necessary libraries needed for image processing
- Started learning the html ,css for frontend.
- Had a few meetings with the mentors and got the gist idea on Steps involved in making the software.

- Made the software sample in matlab using matlab image processing tools.
- Tried to do the same with python and give it a web interface.
- Used the Scikit - image libraries for the image processing part , but faced many errors while accessing the modules. Then changed the libraries from Scikit - image to Opencv libraries.
- Made a few sample codes testing the functions of the Opencv library used for image processing to get the idea about the flow and how these functions work.
- Made a refreshing frontend part by referring to various templates available online using HTML,CSS,Bootstrap,JavaScript.
- Got some errors while reading and giving the threshold values in the image processing part.
- Resolved the errors for threshold and now we can get results up to adaptive mean threshold for a given image.
- Started working on the portion to filter the contours to get the border or boundary of the objects in the image.