

DETERMINING SHAPE AND SIZE OF OBJECTS IN A GIVEN IMAGE

GROUP MEMBER:

<u>Name</u>	<u>Roll Number</u>
Jap Purohit	AUI940109
Purvam Sheth	AUI940151
Nihar Patel	AUI940119
Raj Gariwala	AUI940118
Mohit Prajapati	AUI940171

OBJECTIVE OF THE PROJECT

- Measuring objects within an image or frame can be an important capability for many applications where computer vision is required instead of making physical measurements.
- This application note will cover a basic step-by-step algorithm for isolating a desired object and measuring its diameter/dimension.
- It will also try to determine the 2D shape using the concept of *Contours*.
- This application can be used in determining the dimension and shape of objects in a given image or frame using the 2D image we are available with.

FLOWCHART OF THE ALGORITHM

Start

Asking for the image path and output path from the user

Removing background from the image/frame provided by the user

Converting the resulted image in black and white form

Calculating the pixel per cm from the dot per inch of the image that was fetched from image

Then finding the contours and removing small contours to increase precision

Applying canny function on the resulted image and then applying dilate and erode function

Applying Gaussian Blur on the resulted image

Finding the coordinates and applying Euclidean geometry to find dimension of each contour

Then writing the dimension of the contour on the image

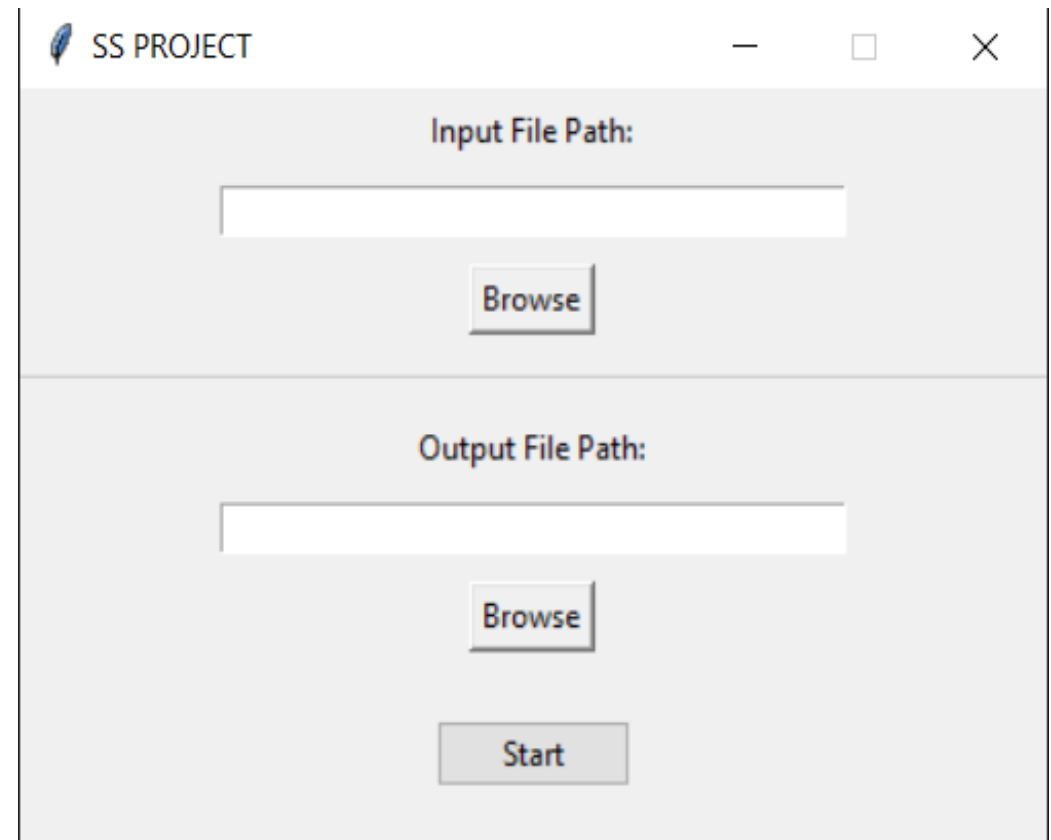
Then find shape of the object based on approxPolyDp and writing it on the image

And show casing the image and output to the user

End

USER INTERFACE-INPUT AND DEMO

- The user interface has majorly allowed user to provide the path of the image where it is stored.
- After all image processing and the user is displayed with 5 images that are background removed image, grey image, blurred grey image, blurred image, and determined shape and dimension image.
- At last all the results output is saved in the path provided by user in a new folder with folder_name as time stamp when application was accessed.
- Images saved will included all outcome after performed each and every image processing function.



The screenshot displays a software window titled "SS PROJECT" with standard window controls (minimize, maximize, close). The interface is divided into two main sections. The top section, labeled "Input File Path:", contains a text input field and a "Browse" button. The bottom section, labeled "Output File Path:", also contains a text input field and a "Browse" button. At the very bottom of the window is a "Start" button.

VALIDATION IN USER INTERFACE AND FACTORS AFFECTING RESULT

■ Validation in User Interface

- This application also provides validation and error detection for user input.
- It only take .png file format or jpg file format image file
- Check whether path provided by the user is correct or not.

■ Factors affecting Results

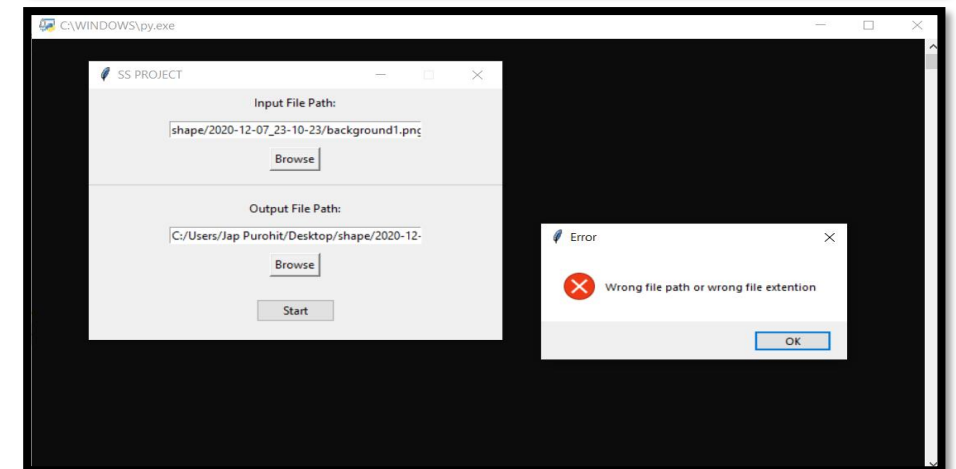
- Image should be strictly 2D image.
- Photograph should be taken perpendicular to the plane of the object where it is placed.

```
validextension = ['.jpg', '.png', '.jpeg']
inputPath = input_entry.get()
try:
    if len(inputPath.strip()) == 0:
        answer()
        return
    kind = filetype.guess(inputPath)
    kindtype = str(kind.extension)
    if kindtype not in validextension:
        answer()
        return
except FileNotFoundError:
    answer()
    return

outputPath = output_entry.get()
try:
    if path.isdir(outputPath) == False:
        answer()
        return
except FileNotFoundError:
    answer()
    return

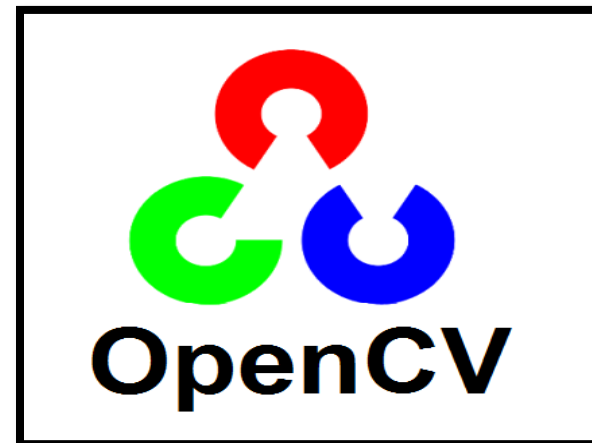
filePathDict['inputPath'] = inputPath
filePathDict['outputPath'] = outputPath

print(filePathDict)
mainfunction(filePathDict)
```



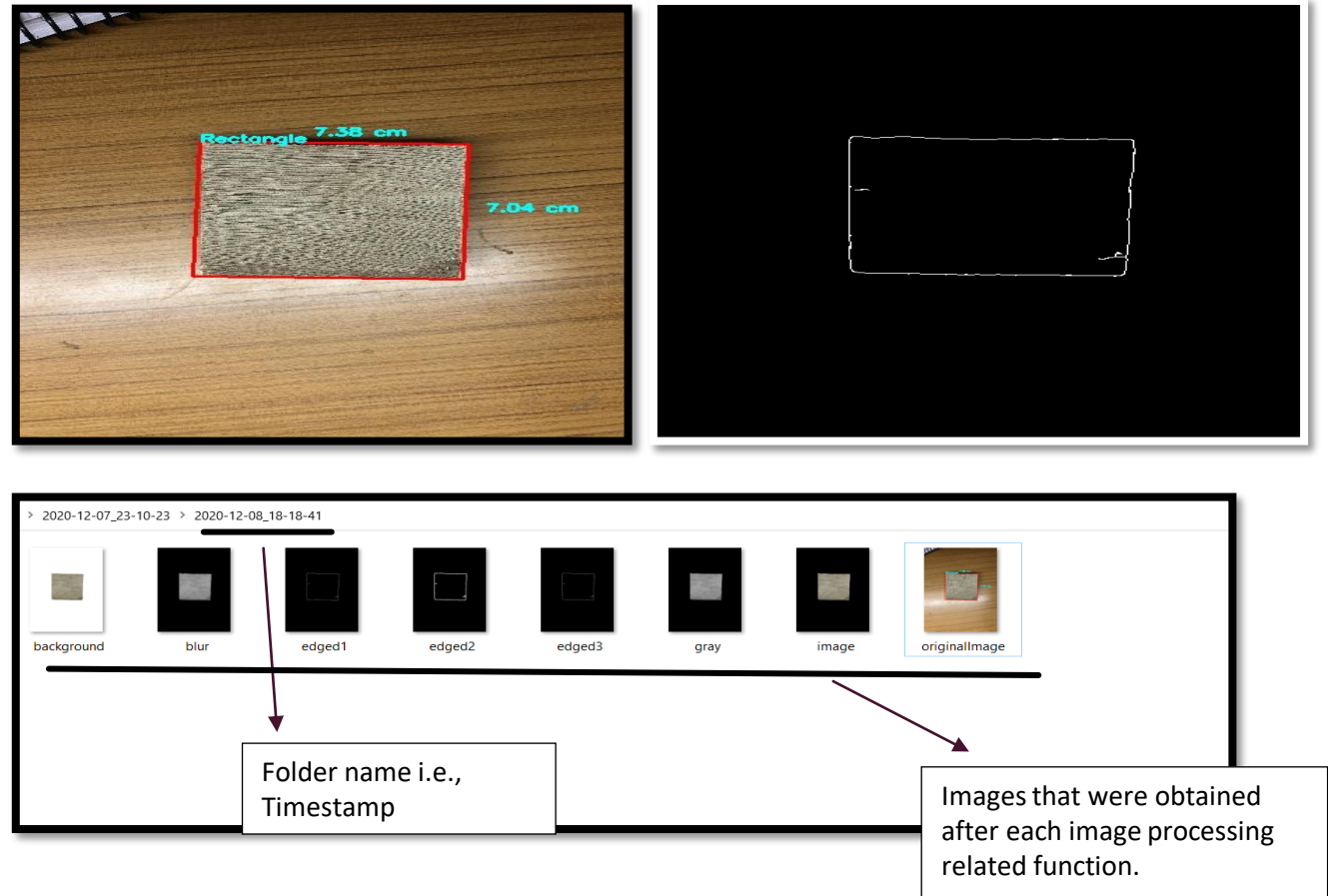
TOOLS USED IN MAKING THIS PROJECT

- Programming Language used for this project is **“Python 3.7.3”**
- Library used in making this project:
 - scipy for measuring Euclidean distance
 - imutils for contours
 - remove_bg_api for removing background
 - numpy for handling image data
 - cv2 for image processing
 - tkinter for Interface creation
 - os library for handling with file system and validation of user input.
- File submission would be in the format of “.py”



RESULTS OBTAINED

- While Displaying the image:
 - Majorly five images are displayed to the user : removed background image, black and white image, blurred image, edge detected image, final output image.
- Images Saved:
 - All images obtained after each step of image processing. All this image is are stored to the output path provided by user.



CONCLUSION

- On a closing note, this project was a great chance for us to learn the principles of Canny i.e. Gaussian convolution to smooth the input image and remove noise.
- In addition to the fact that we learned substantially more about it and applied into the project. In addition to this, there are many other elements for resizing of an image by decimation and interpolation.
- The big task while making this project for us was to create an user interface which would take the input and output location of the image into which validation for the image extension is also added as this was the first time we were supposed to merge the locations into a same interface.
- But apart from that, engaging with the team was a remarkable opportunity, and getting their input on how to go forward to tackle a specific challenge and eventually finished the project. Each member of the group made an equal contribution to the analysis of the project.