Object Segmentation and Estimating the dimensions of the objects in the image using reference object method

Tools/Libraries used:

1. Spyder to use python
2. Opencv for image processing and pre-processing
3. Libraries include matplotlib (to plot images), imutils (assists in using cv2 functions), argparse (helps in getting the command line arrguments), scipy (provides other functionalities like math functions), numpy (to help with fast numbers processing)

**OBJECT SEGMENTATION**

Important terms:

1. Contours: It is simply a curve joining all the continuous points have the same color or intensity. The concept of contours helps in easily identifying an object in an image.
2. Thresholds: Applying thresholding on an image helps in binarizing the image. It takes an input image, a threshold value and a maximum value. It will compare the pixel values with the threshold value. If the pixel value is more, the pixel is set to high value and if not, it is set to low value generally. Thus, binarizing the image which helps in object segmentation.

[Multiple thresholding techniques](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html) are available, such as inverting the output from the one discussed above. There is also, **adaptive thresholding** which helps in cases where there is uneven lighting in the image.

**Otsu’s thresholding** technique helps in determining the threshold value which can sometimes not be determined manually.

1. Argument Parser: To create this project as a library, there is a need to call it parsing some arguments. For this, an instance of **ArgumentParser** is created and the required arguments are listed out along with sufficient hints for the user to call the library functions effectively.
2. Reference Object: A standard object whose dimensions are known is also a part of the image. It is recommended to place the reference object at some fixed end or use a standard object that stands apart in color/shape when compared to other objects whose dimensions need to be measured.

Preprocessing of the image:

1. Images with any general extension is provided as input. If the image is really big, to work on this image and identify various effects on it, it has to be scaled down. This is done using the resize method provided by the imutils library. The ratio to which it is scaled down is made not of which is useful in scaling the distance computed in the later part.
2. In this step, the image is converted to grayscale. This is done since it is very efficient to apply thresholding on a grayscale image. Otherwise, the RGB values of a pixel has to be compared with a threshold tuple of RGB values. This is not an efficient way. Hence, it is converted to grayscale. Also, contouring works best with black and white images which is a result of thresholding. Conversion to grayscale is accomplished using the method from cv2 (OpenCV) library.

[Original](https://i.stack.imgur.com/43cy1m.png)  [Grayscale Image](https://i.stack.imgur.com/eQ4KIm.png)

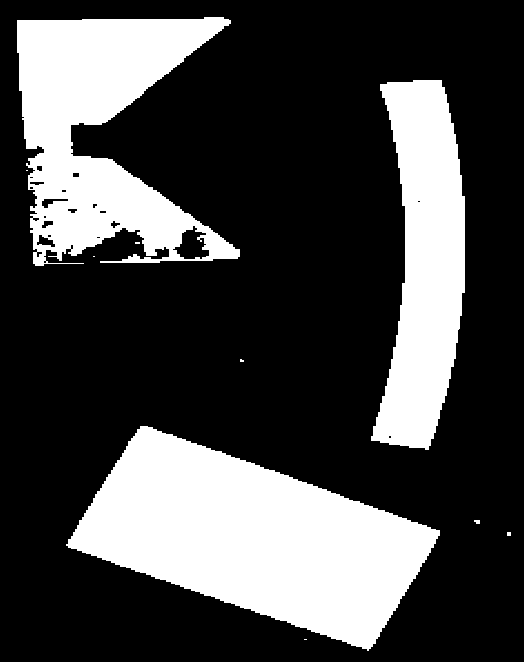
1. Once it is converted to grayscale, blurring of image is done. The object whose dimension is to be found need not always have a clear surface. Especially in cases where the blocks provided in mechanical workshops, which are so often prone to rust and often comes with irregularities. Various methods are available to blur an image like simple blur, Gaussian blur, Median blur, et cetera. Each have their own advantages and disadvantages. All these available in [OpenCV library](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_filtering/py_filtering.html).

[](http://srome.github.io/images/heatblur/output_18_1.png)

1. The blurred image serves well for thresholding/binarizing the image. The concept is discussed already.

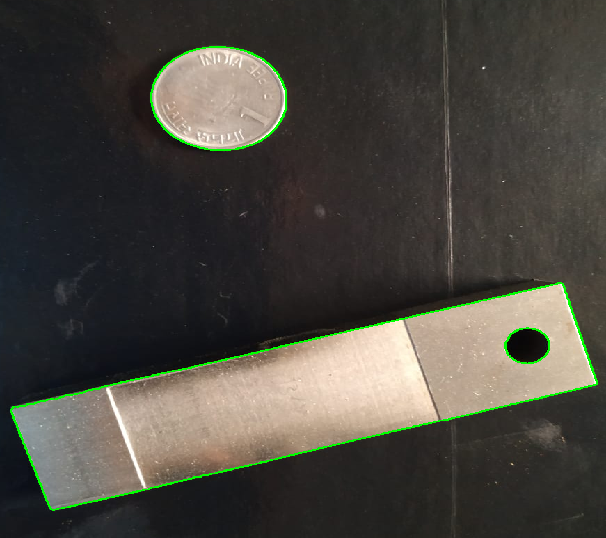
[](https://www.tutorialspoint.com/opencv/opencv_adaptive_threshold.htm)

1. Using Canny edge detector to helps in identifying the edges in an image. Alongside this, dilation and eroding of the image is done which helps in closing in on minute gaps and approximating the edges.

Input threshold Image Applying erosion and dilation

1. With everything set, this last image shown above is given as input to findContours and grabContours method provided by the OpenCV and imutils library respectively. It also provides option to roughly draw the contours or draw contour pixel wise, which can be provided as a parameter which specifies the contour approximation method. It finds all the contours and returns a list of contours in the image. The contours are a useful tool for shape analysis and object detection and recognition.
2. Now all that is left is to iterate through contours and delete smaller contours which are not the objects of interests. These are generally the irregularities which escaped all the preprocessing.
3. Now iterating through different contours, different edges are identified specifying the coordinates using approxPolyDP method which enables to approximating the edges and draw a contour around it. It returns a set of coordinates of the corners of each contour.



1. Once the coordinates are computed, the distance between every successive point is computed.
2. In the above image, the coin serves as the reference object. The ratio of the actual dimensions to the dimension provided by the algorithm is computed. The dimension of other objects is then multiplied with the ratio, thereby giving the real-time dimensions of the object.