**Name: Zeyad Mahmoud Abdelrahman Bassiouny**

**ID: 20201374047**

**Unknown Object Size Detection**

This project aims to detect the size of any unknown object using an A4 paper as it’s reference.

**Step 1: Imports used**

numpy and cv2

**Step 2: obtaining input from the user**

The input can either be live video feed using webcam or jpeg.

**Step 3: Obtaining contours**

In this step I created a function to obtain the contours of the entire image.

But first, we need to change the image to gray scale, apply gaussian blur and apply canny edge detection which is needed to get the contours of the image.

I also applied some dilation and erosion to make edges smoother and more precise.

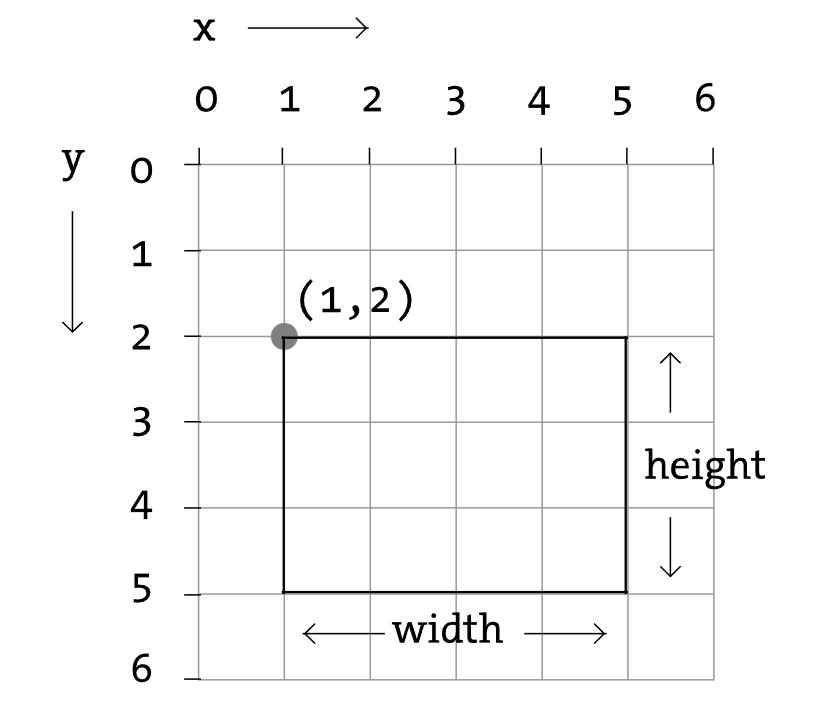
I then used **cv2.findContours** which returns the contours of the image and the hierarchy (i.e. shapes inside shapes. In this case, we call outer one as parent and inner one as child.)

Now that we have all the contours, we need to find the shapes of each object in the image. We loop over the list of contours obtaining the area using **cv2.contourArea** (as to ignore very small objects), perimeter using **cv2.arcLength**, approximation of object curves using **cv2.approxPolyDP** and finally the bounding rectangle using **cv2.boundingRect** which contains the corner points of our object. Next, we append all this information into our list of final contours.

Since we’re using A4 paper as our reference point with every other object inside of it then it must be the biggest object in the image. So, to easily find out which contour belongs to the A4 paper is in the list we sort it based on area and get the object of index 0.

**Step 4: Warping The Image And Top-down View.**

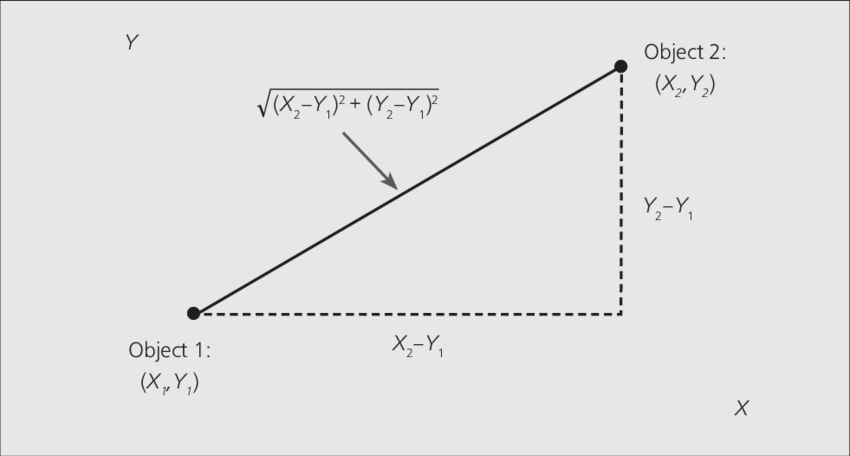
Since the A4 paper is our reference point we’re going to crop anything outside of it and warping it to fit the screen. The first step is finding the correct order of points (i.e. top left, bottom right, etc.). For this I created a reordering function, which finds the sum and difference of the coordinates of each point. The point with the smallest sum is the closest to origin and the point with the largest sum is the furthest one (i.e. top left and bottom right). Next we assume that the height is greater than the width so by finding the difference between the coordinates the smallest difference should be the top right point the biggest difference is the bottom left point.



Now that we have our ordered points, we need to warp said to the corners of the original image. For this step we’re going to need the transformation matrix which we’ll get using **cv2.getPerspectiveTransform** and use **cv2.warpPerspective** to warp the images with said matrix. Finally we add some padding to make the edges smoother when the image is shown.

**Step 5: Finding Objects And Their Size In The A4 Paper**

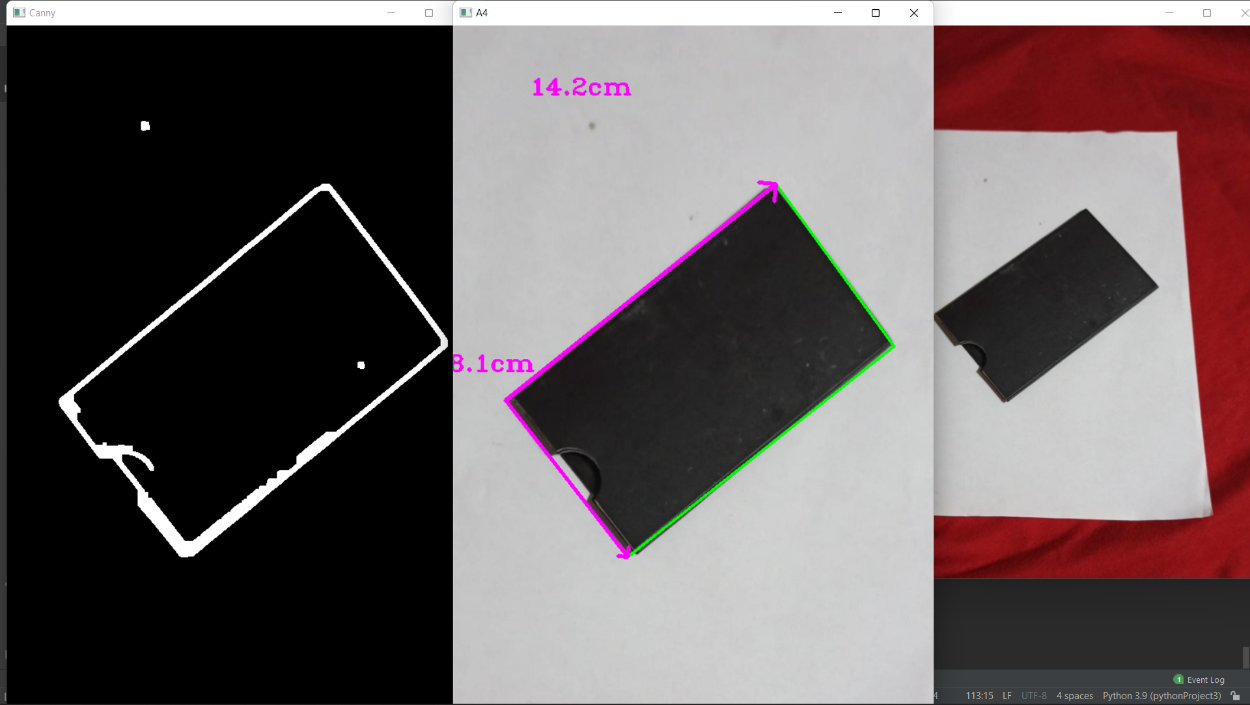
Now that we have our A4 image, we can use the **getContours** function to obtain the contours of the objects within the paper. Next we loop on said objects while skipping any object that doesn’t have 4 corner points, for each object we draw a rectangle/square on it using **cv2.polylines** function to mark it on the A4 image.

Now we need to reorder the points of each object to obtain the lengths of it’s sides. To get said lengths I created a function that implements the Euclidean distance theorem. 

Finally, I used **cv2.putText** to add the size of the object as text on the image.

**Testing:**

Image:



Video:

