CS5330 Project 3 Report

Bryan Teck Yean Ang

Description of Overall Project

Image recognition under computer vision has been a keen interest of study since the sixties. Methods like edge detection, segmentation and thresholding have been early pioneers in that space. In this project, image recognition is implemented using these methods. Specifically in the following steps:

1. Thresholding an object from the background
2. Cleaning up the binary image with erosion and dilation
3. Segmentation to get a map of regions/ items in the image.
4. Computing the features of the region of interest through moment calculations
5. Creating a database of known items and their respective features
6. Finally, recognizing new objects with the given database using algorithms like k-nearest neighbors.

Required texts and images

Task 1

For the first task of thresholding, the input image is first turned to grayscale and blurred to decrease any noise. Then it is thresholded using OpenCVs built-in libraries. All candidates were examined, but ultimately the Otsu's Binarization was used. Otsu's Binarization generates a histogram of the image and selects a threshold value that is between two peaks of the histogram. This provided more consistent thresholding over a simple threshold value, and better than adaptive thresholds that wouldn't threshold big objects well.

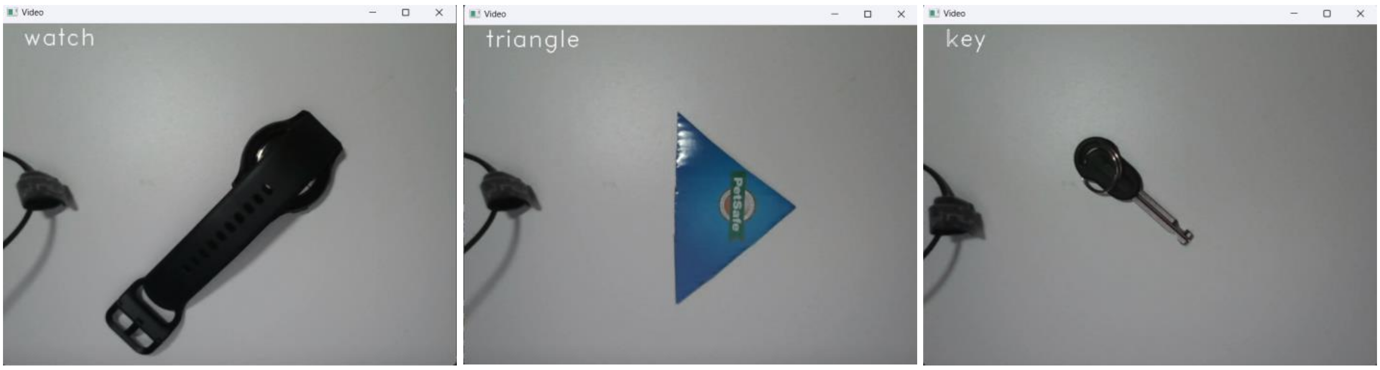


Image: The 3 original raw, unfiltered objects. An extra cable is present to demonstrate segmentation.

Image: The images after being pass through the Otsu binarization thresholding algorithm.

Task 2

The morphological strategy used was to first run 4 passes of dilation, followed by 2 erosions, 4 more dilations, and finally 6 erosions to ensure regions don't grow. This was done because the input images are very clean from noise and this dilation can be used first. The images used also struggle with light colored objects or bright highlights, presenting holes in the object, thus such a high number of dilations were used.

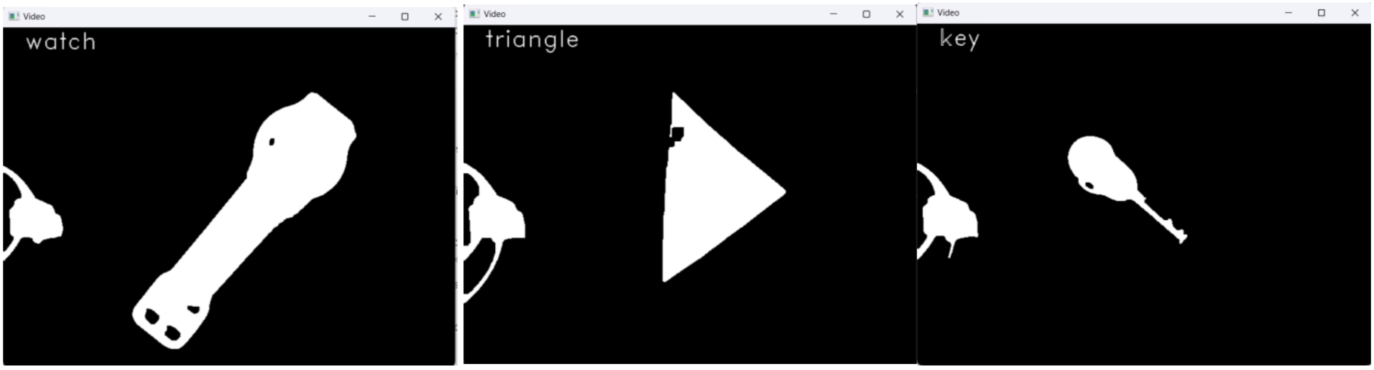


Image: The 3 examples after morphological processing was applied. Note the filling of the holes in all objects, even in the shadow of the cable in the middle picture.

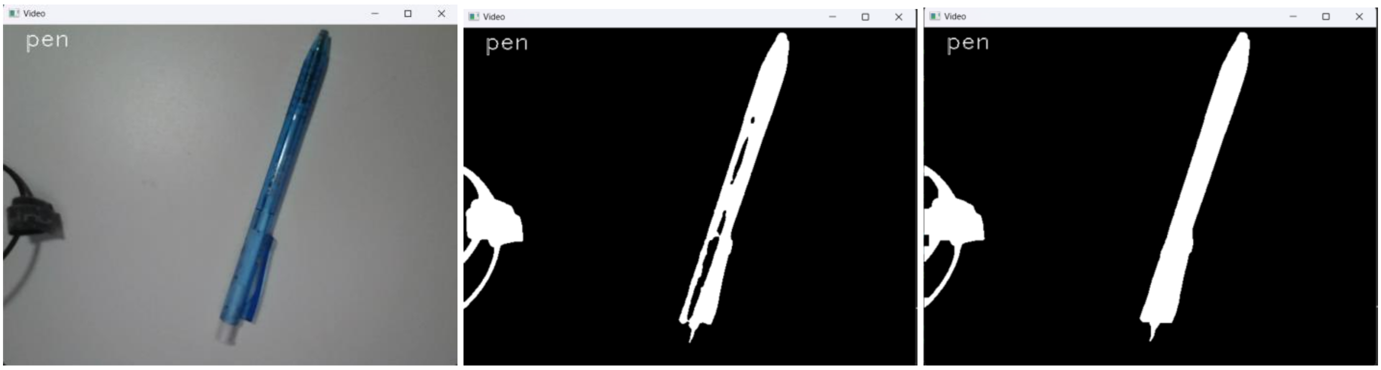


Image: A more obvious example of the morphological strategy succesfully filling holes for an object with bright hightlights.

Task 3

For task 3, the region growing algorithm was implemented from scratch, generating a region map for each region.

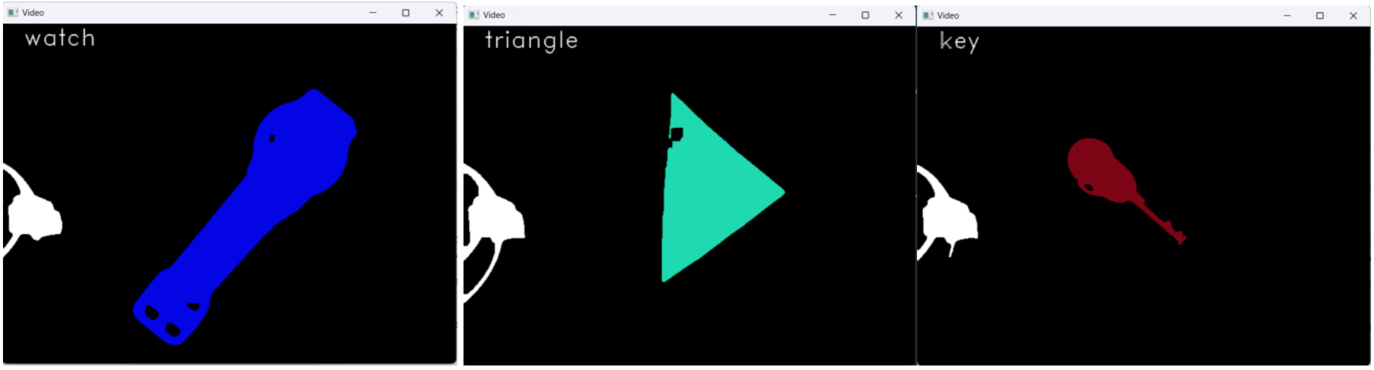


Image: The 3 examples after segmentation with the region growing algorithm.

Image: An example of an image with multiple regions with randomly assigned colors.

Task 4

First of all, after the region map is produced, the image is filtered to only leave behind the biggest region at the center of the image, giving only one object to work with. And then the following were computed and used as features for each object:

- Percentage of bounding box filled.
- Longer side to shorter side ratio of the bounding box.
- The 7 central moments from OpenCV humoments function.

Percentage filled: The bounding box is first found using minAreaRect(), which finds the *rotated* bounding box for the object. The width and height can then be calculated using the vertices of the bounding box. The area is then used to calculate the ration of object size to bounding box area.

Longer side to shorter side ratio: This ratio was used instead of height to width ratio to make it rotation invariant. So no matter whether the x length is greater than y height, it is the same ratio for the same object.

The 7 central moments from OpenCV humoments function:

The 7 central moments from the humoments functions are central moments that are invariant to translation, scale, and rotation, and reflection. These moments describe properties of the object like size, orientation, symmetrical-ness, elongation, pointiness, and curviness.

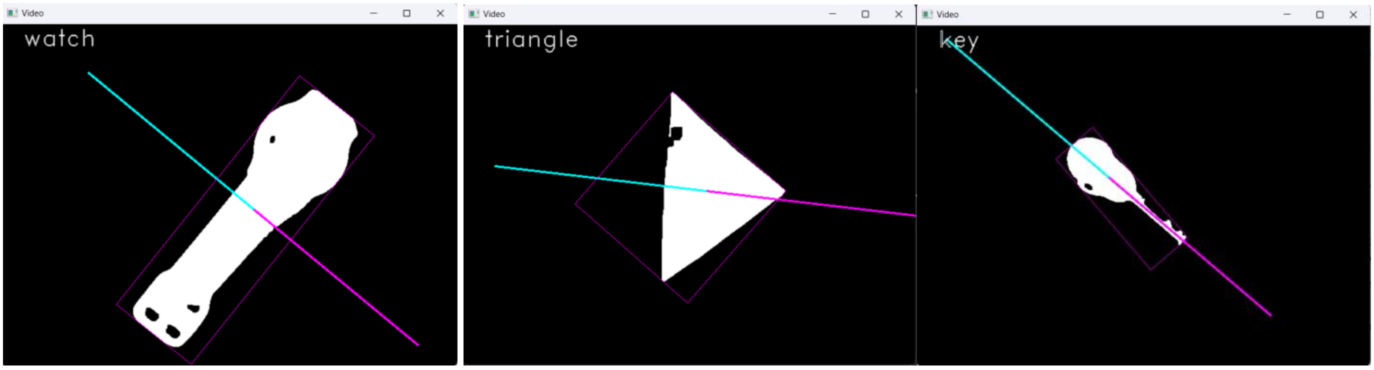


Image: The 3 example images with the rotated bounding boxes in cyan, and the axis of least central moments running through their center.

Furthermore, the angle of the axis of least central moment was calculated from scratch using the formulas presented in Professor Bruce Maxwell's lecture. The axis is then displayed as a line originating from the centroid of the object.

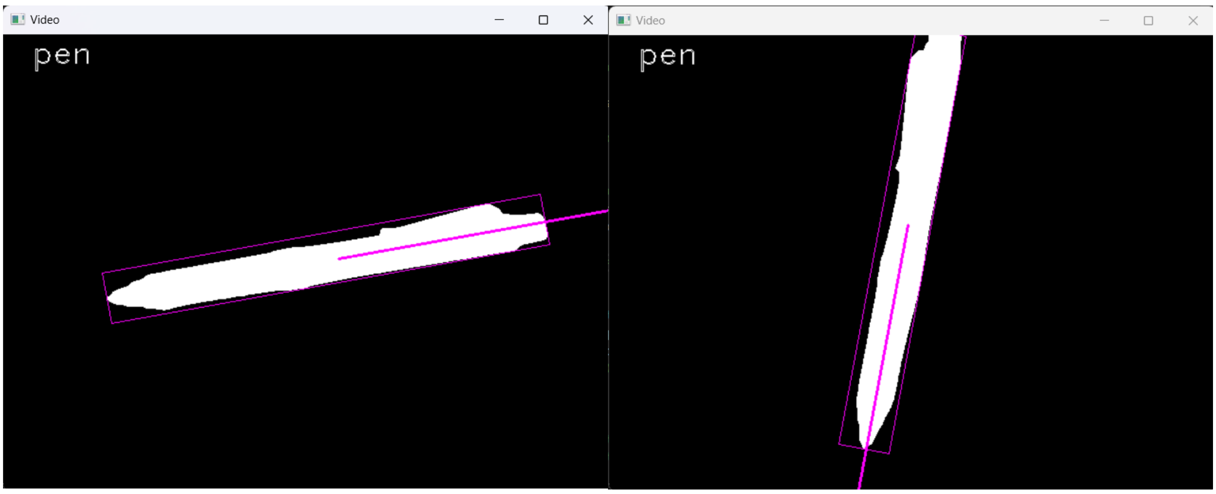


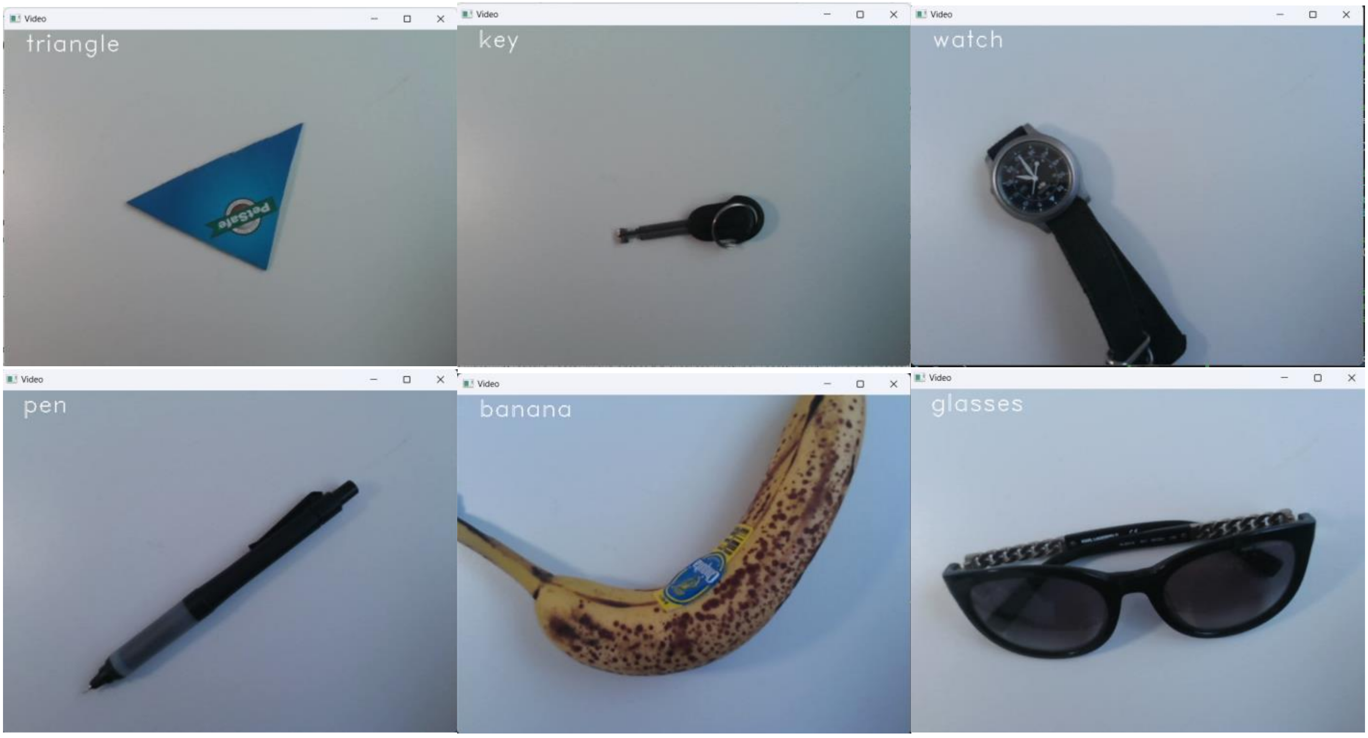
Image: An image of a long pen showing how the axis of least central location and bounding box is invariant to rotation and translation. In either orientations the axis runs through the long part of the pen.

Task 5

A training system was implemented in the GUI directly by allowing users to press the spacebar (' ') to save an object and it's features. By pressing the spacebar, the name of the label is entered into the console, and all it's features and label is saved into a database file (db.txt).

Task 6

For this task, the single nearest neighbor algorithm was implemented to calculate distances. Every feature was scaled Euclidean-ly. The standard deviation is calculated and used to scale. The following 12 objects were used to train the database. Each object was moved and rotated to get 2 data points for each object.



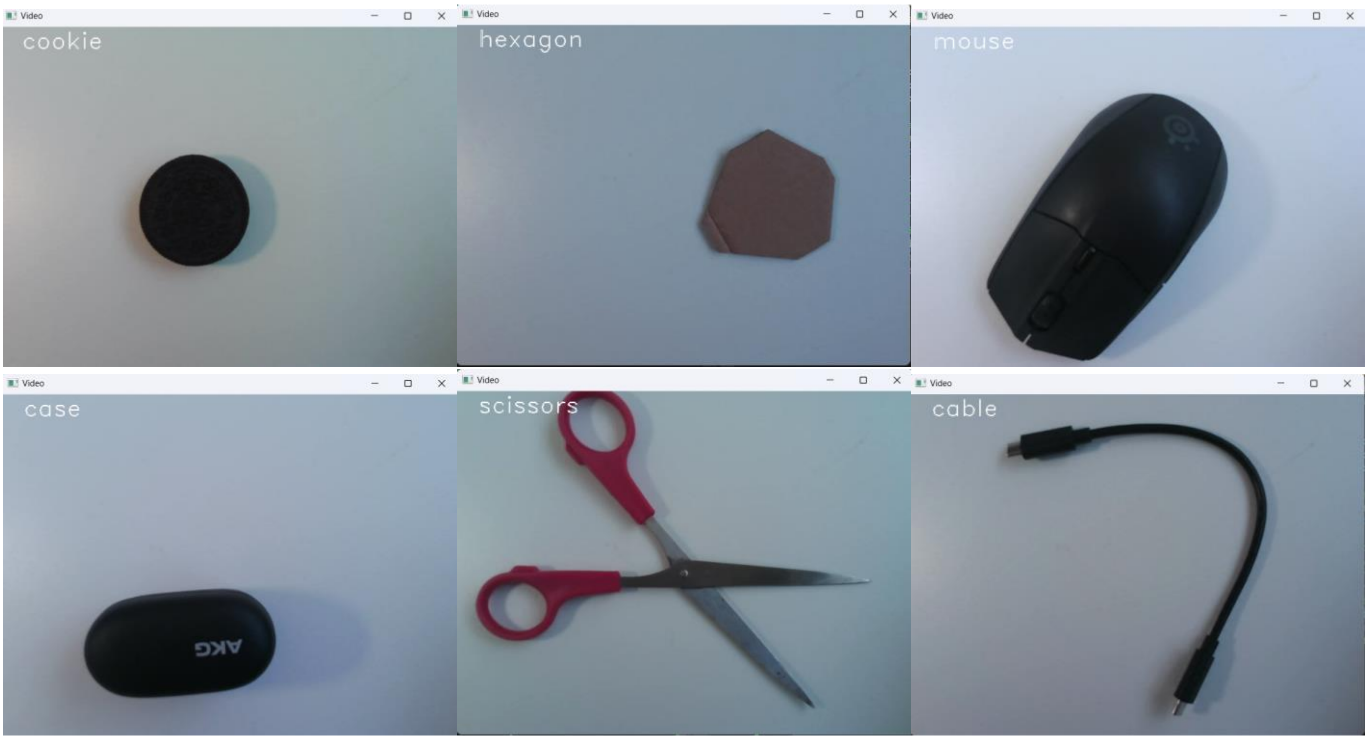


Image: All 12 items trained with and their respective labels

Task 7

K-nearest neighbor was implemented.

Task 8

The following confusion matrix was obtained (this matrix was also generated in the recording of the system). A perfect system would have a diagonal of check marks.

		Actual											
		banana	cable	case	cookie	glasses	hexagon	key	mouse	pen	scissors	triangle	watch
	banana	1											
	cable												
	case			1									
	cookie												
	glasses					1							1
	hexagon		1								1		
	key							1					
	mouse				1		1		1				
	pen									1			
	scissors												
	triangle											1	
Predicted	watch												

Task 9

The linked video goes through the real time classification of all the objects trained on. At the end, the different visuals are also demonstrated, showing how to switch between the raw image, and all the various processed images shown in the report.

https://drive.google.com/file/d/186RCFxFdpr6KZQ04bO0Xz8Kxu__MRHLA/view?usp=share_link

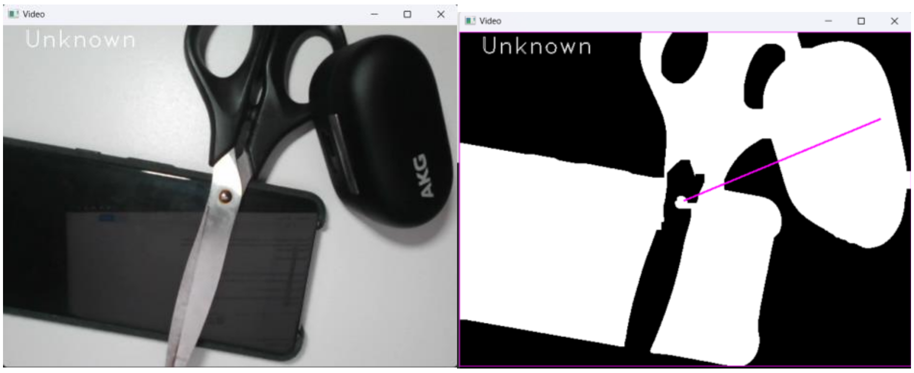
Extensions

GUI: The GUI has been extended to be able to show all different steps of the pipeline with various button toggles. 2 different features (bounding box and axis) were shown for the feature view.

12 Objects: Additional 2 objects were added

Extra implementations: The region growing and calculations for moments around axis of least central moment was implemented from scratch.

Unknown classification thresholding: A threshold of 2 standard deviations units was used to allow the system to classify unknowns. The formula used was to take the minimum distance to a data label in k-nearest, and divide by the sum of all feature standard deviations.



Reflection

Reflecting on the project, I have learnt many things.

Binary Thresholding: I've learnt the multiple different methods of binary thresholding, from simple ones, to adaptive ones to Otsu's binarization

Different ways of segmentation: In this project we used region growing or a 2-pass segmentation to find regions, but over the course of the project I've also learnt of other ways like using OpenCV's contours and edge detection to find regions.

Moments and rotated bounding boxes

Many many different OpenCV functionalities

K-nearest neighbor algorithms

Acknowledgement

Thresholding

https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html

Rotated bounding box tutorial
https://docs.opencv.org/3.4/de/d62/tutorial_bounding_rotated_ellipses.html

Hu Moments

https://docs.opencv.org/3.4/d3/dc0/group__imgproc__shape.html#gab001db45c1f1af6cbdbe64df04c4e944

https://www.researchgate.net/publication/224146066_Analysis_of_Hu's_moment_invariants_on_image_scaling_and_rotation

Santosh Vasa – for providing resources for central axis of least moment

Gopal Krishna – for providing help with feature extraction with humoments.