Project 3 – Real-time 2-D Object Recognition

Abinav Anantharaman | NUID: 002774223

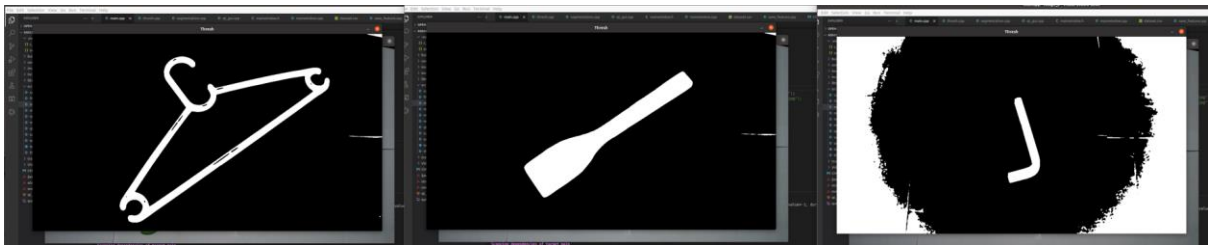
Satwik Shridhar Bhandiwad | NUID: 002920338

Introduction:

In this project, we worked on real-time 2-D object recognition. In task one the input video is thresholded to separate an object from the background. In task two, the binary image is cleaned up using morphological operations, which solves issues such as noise and holes. In task three, the image is segmented into regions using connected components analysis, and the system ignores regions that are too small. In task four, we computed translation, scale, and rotation invariant features such as hu moments as well as height to width ratio of Bounding Box and Percentage area filled. We created a CSV file containing features of all the objects referred to as feature Database. We performed object detection by calculating scaled Eucledian distance and attaching the label with least distance. We also performed classification using KNN for higher accuracy.

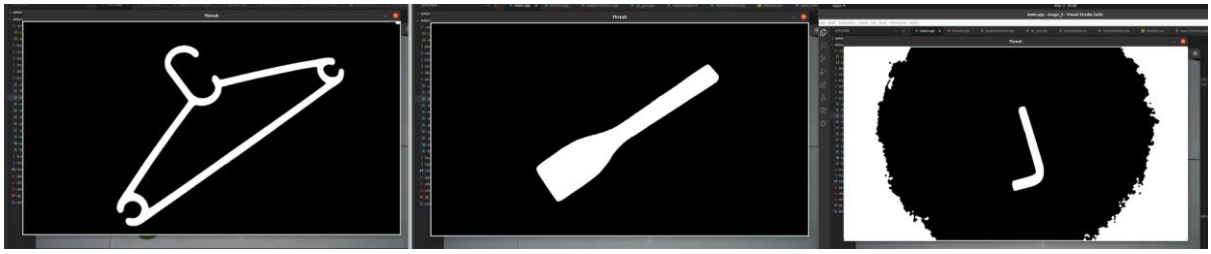
Task1: Threshold the input video

Above are the original images of Hanger, Spatula and Allen Key before thresholding. These images will be used as reference for further tasks.



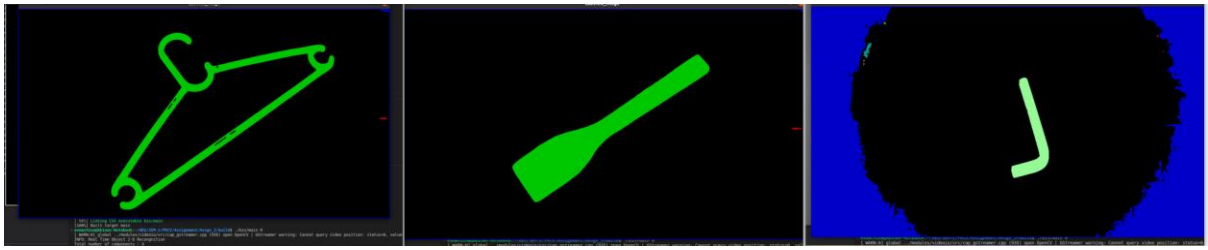
Output Image of Hanger, Spatula and Allenkey after thresholding. A fuction is implemented to calculate Threshold value by using the Otsu Thresholding method.

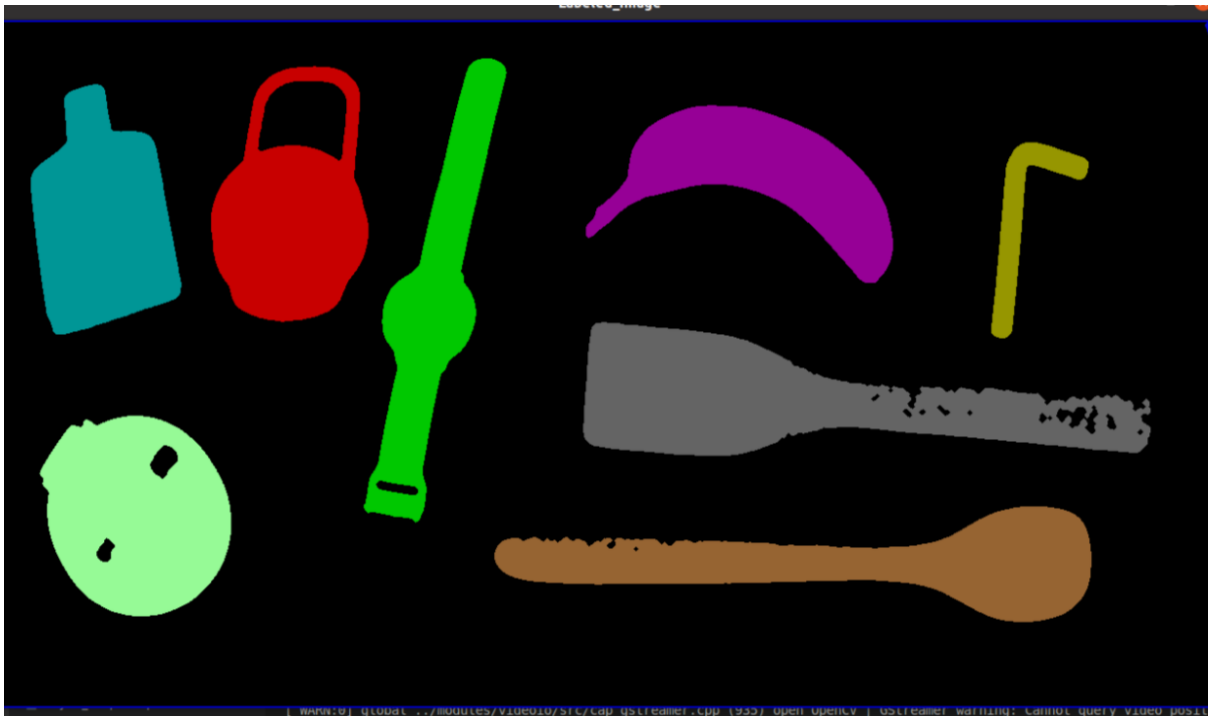
Task2: Clean up the binary image



Output Image of Hanger, Spatula and Allenkey after applying Grassfire Transformation and performing erosion and dilation with distance of 3 from foreground. A function is implemented for Grassfire transformation.

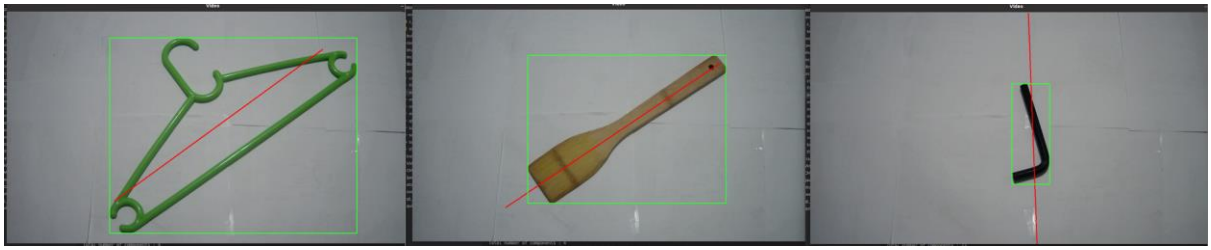
Task3: Segment the image into regions





Output Image containing different objects sepereated into different segments and represented using different colors. Used connected components functions from OpenCV.

Task4: Compute features for each major region



Output Image of Hanger, Spatula and Allenkey with Bounding Box and axis of least central moment.

The features extracted are Height to Width ratio, Percentage area filled, h0, h1, h2 and h3 Hu moments. A function is implemented to calculate all the 4 Hu moments by first calculating the central moments and then Hu moments using formula:

$$\mu_{ij} = \sum_x \sum_y (x - \bar{x})^i (y - \bar{y})^j I(x, y)$$

$$\eta_{ij} = \frac{\mu_{i,j}}{\mu_{00}^{(i+j)/2+1}}$$

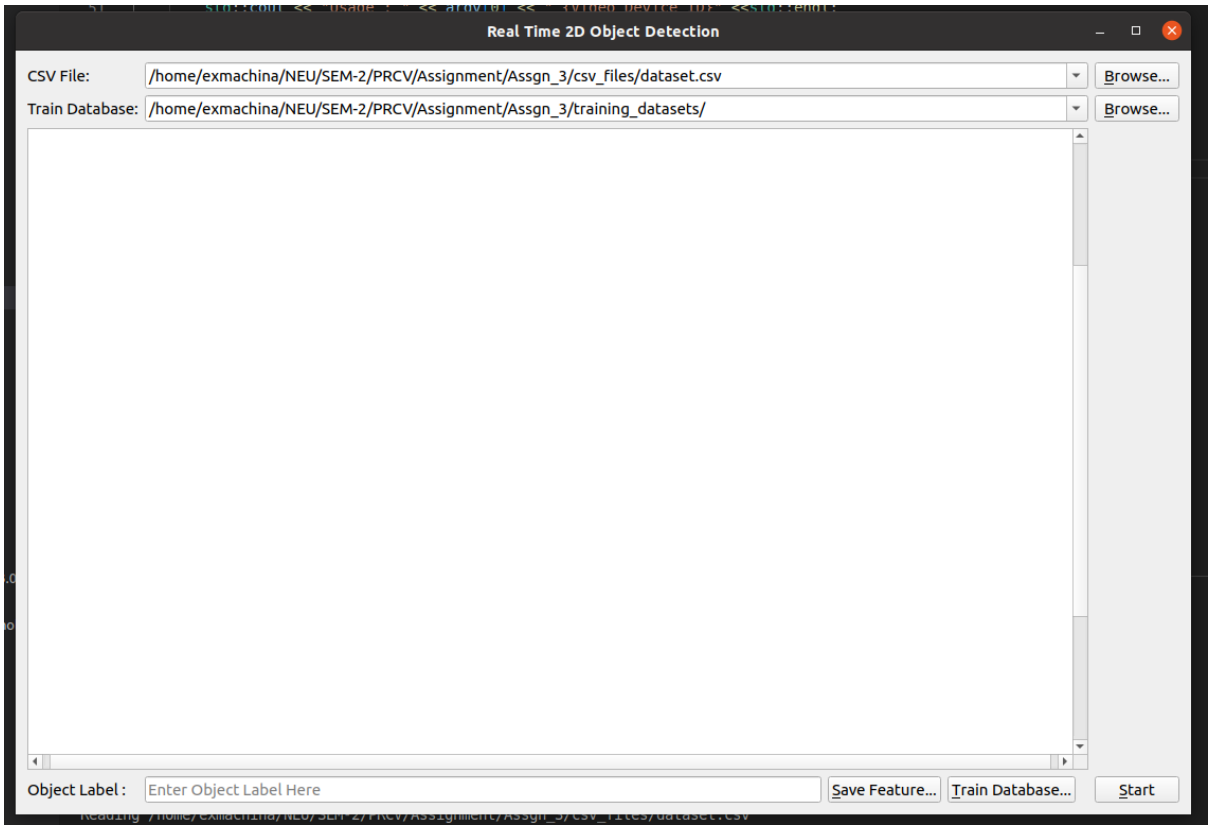
$$\begin{aligned} h_0 &= \eta_{20} + \eta_{02} \\ h_1 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ h_2 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ h_3 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ h_4 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (6)$$

Reference: [Shape Matching using Hu Moments (C++ / Python) | LearnOpenCV]

Task5: Collect training data

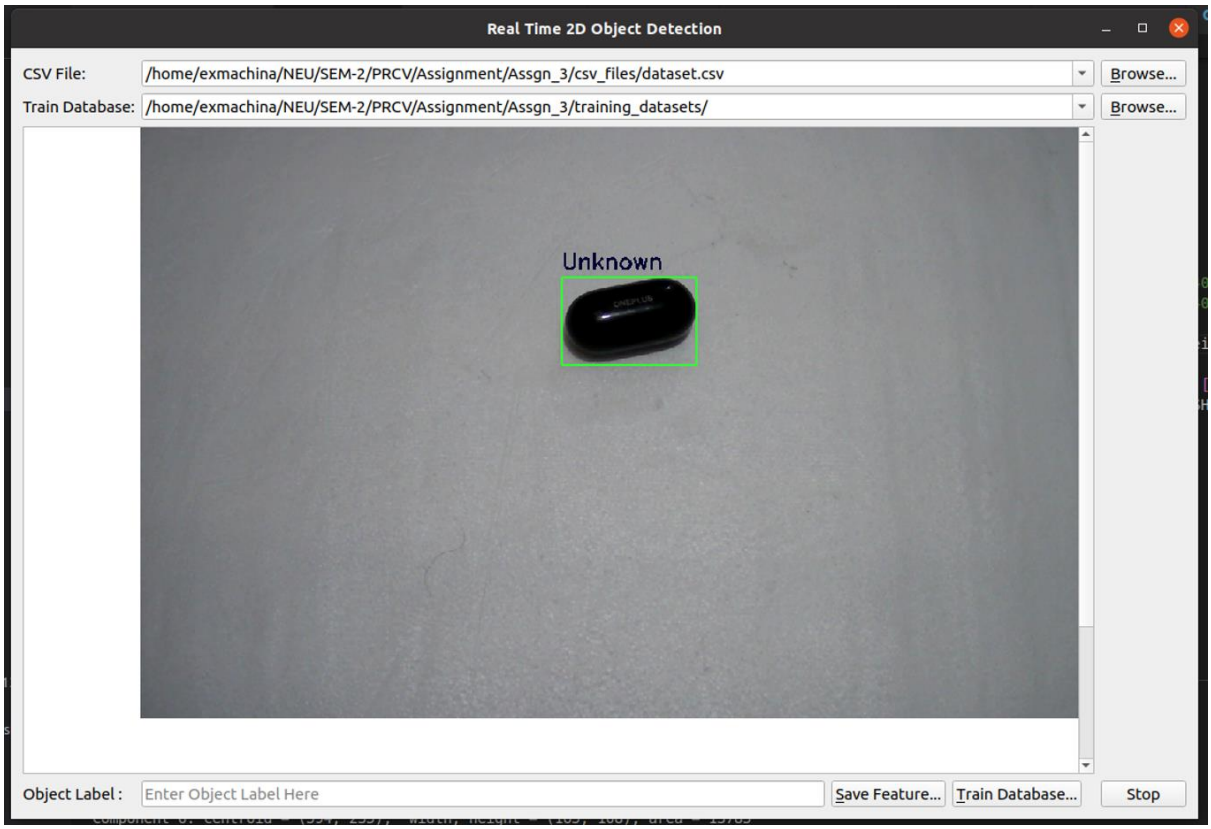
A GUI is provided for training and real-time object detection.
There are two ways to collect training data.

Option1: Collect data using pre captured images.



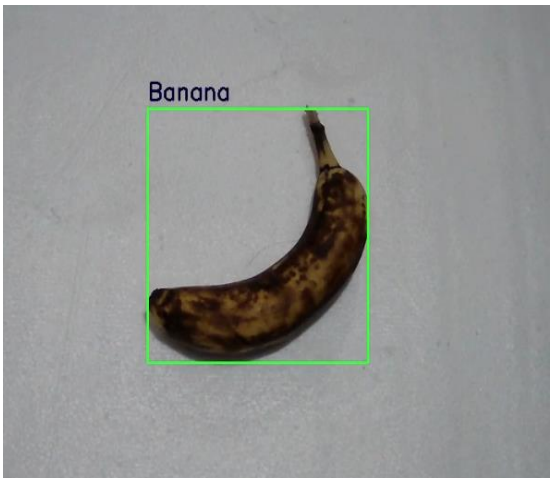
Enter the csv file name by broesing the filepath to store training data.
Select the folder containing images of an object in different orientation in Train Database field. Enter the label to be stored for this object. Ex: Soap_Box in the Object Label field.
Finally, click on Train Database button. This will compute the features and store in CSV.

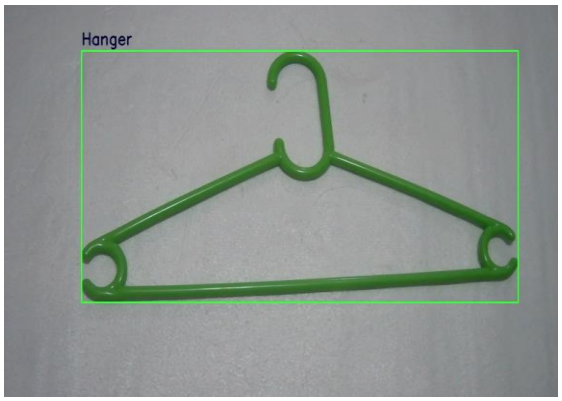
Option2: Collect training data from live feed.

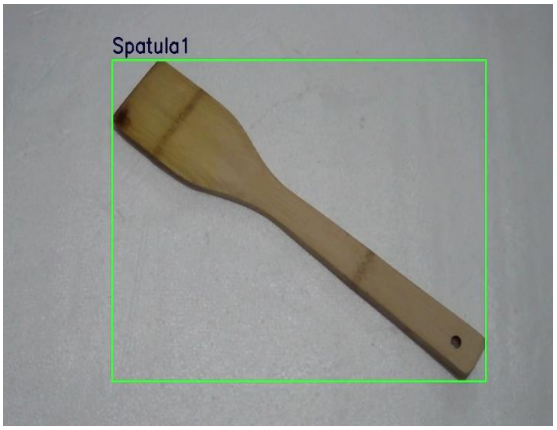


Start the live feed for object detection by clicking start in the lower bottom corner.
When an unknown object arrives, enter the new label for this object in Object Label field and click Save Feature Button. This will save the feature of this new object in the CSV file.

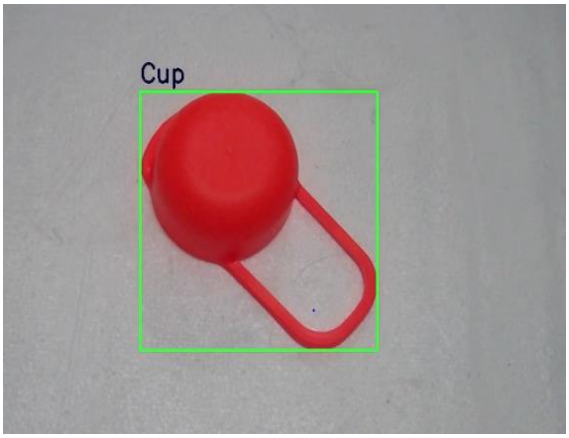
Task6: Classify new images

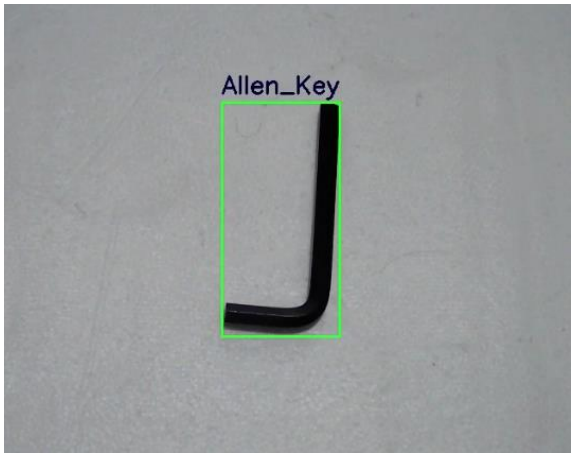







Spatula2


Cup


Allen_Key


Watch


Lid

Extension1: Unknown Objects in live feed

Task7: Implement a different classifier.

KNN classifier is implemented with K = 3. Scaled Eucledian distance of target image with 3 images from each category is calculated and their sum is taken to assign the label. The sum with least value is used for assigning the label.

Task8: Evaluate Performance of the system.

Classes (X) / Object Under Observation(Y)	Allen_Key	Banana	BodyLotion_Dispenser	Cup	Hanger	Lid	Perfume_Bottle	RollingPin	Spatula1	Spatula2	Watch
Allen_Key	11.2257	67932.2	inf	inf	1399.32	inf	inf	inf	20.3143	30.172	54.0373
Banana	379.247	12.6199	284.553	2676.94	5163.07	inf	21046.2	2997.49	334.749	343.384	765.95
BodyLotion_Dispenser	260.721	130.909	10.6806	1362.34	4740.85	inf	2627.81	716.289	203.452	189.829	453.637
Cup	323.863	704.55	322.87	4.08513	4384.02	53605.1	3730.29	1526.68	255.174	227.957	622.518
Hanger	973.426	inf	inf	inf	17.3037	inf	inf	inf	479.161	102.993	1307.72
Lid	456.293	1480.39	413.992	97.672	8296.53	12.3734	23.1141	1269.9	304.008	266.839	722.055
Perfume_Bottle	408.646	1273.05	342.845	43.6123	6698.34	1745.51	5.35485	1226.09	291.973	256.675	700.732
RollingPin	236.311	26983	13642.9	inf	11267.2	inf	inf	13.4814	73.0261	79.9853	39.8837
Spatula1	66.4339	inf	inf	inf	784.882	inf	inf	inf	7.00216	31.9948	255.158
Spatula2	2510.06	inf	inf	inf	4585.81	inf	inf	inf	290.916	12.1274	3021.72
Watch	23.9708	45951	27999.7	inf	632.32	inf	inf	20572.2	33.3221	30.1431	5.29583

A confusion matrix is plotted.
We use scaled euclidean distance as the distance metric for comparison. We find the min of distances for all features with the feature of the object under observation. If any object match

falls below a threshold of 20 we call it a match else the object is said to be unkown to avoid false detection. In the confusion matrix we have shown the distance of all feature vectors available for each object in database with the feature vector of object under observation. All matches fall below a value of 20 and are shown in green and Values between 20 and 50 are next close matches and shown in yellow. All value which fall above 10^6 values are written as INF and shown in red, these are 100% no match objects.

Task9: Demo of system working.

https://drive.google.com/file/d/15dM4onFmvvXxrJntgBqWfXYrFrRhlSP1/view?usp=share_link

Extension1: As mentioned above a GUI is implemented for training and detection.

Extension2: Unknown object detection and facility to add the new object to the training database.

https://drive.google.com/file/d/1qptI_yHT2K67ENRm3Ya-ENFdviwW4fX4/view?usp=sharing

Project Outputs:

https://drive.google.com/drive/u/0/folders/1ZdxCAeZjJt6d__WswPsQ4F_9jTAPC2d6

Learning Outcomes:

1. Gained experience with a range of image processing techniques such as thresholding, morphological filtering, and connected components analysis.
2. Learnt how to extract meaningful features that are translation, scale, and rotation invariant, such as moments around the central axis of rotation.
3. Gained experience with building real-time systems that can process image data in real-time.
4. Problem solving: Learnt how to break down a complex problem into smaller, more manageable sub-problems and develop strategies for solving each sub-problem.

Acknowledgments:

Shape Matching using Hu Moments (C++ / Python) | LearnOpenCV

https://hbyacademic.medium.com/otsu-thresholding-4337710dc519

https://www.researchgate.net/publication/342038946_OVERVIEW_OF_DIFFERENT_THRESHOLDING_METHODS_IN_IMAGE_PROCESSING

http://www.ripublication.com/ijaerdoi/2015/ijaerv10n9_20.pdf