Aayush Hitesh Sanghvi Yogi Hetal Shah

# Project 3: Real-time 2-D Object Recognition

**Abstract:**
This project introduces a real-time 2D object recognition system designed for overhead camera perspectives, capable of identifying specified objects on a white surface with translation, scale, and rotation invariance. The algorithm uses computer vision techniques, including preprocessing, segmentation, and feature extraction, the system accurately characterizes objects based on shape, contour, and texture features. By implementing algorithms for orientation estimation and bounding box calculation, objects are precisely annotated with their categories and positions in real-time output streams. Algorithm also features training mode and inference mode to train the system to identify various object depending on the features extracted. System uses the same feature set to compare it with the features of unknown objects placed in front of the camera and assigns a label recognising the object.

**Result:**
**Task 1**: Threshold the input video.

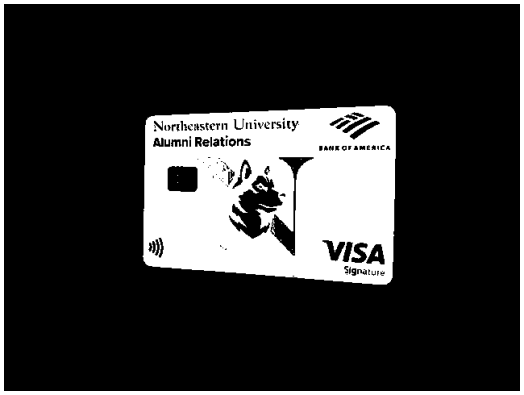



Fig 1. Threshold image of credit card

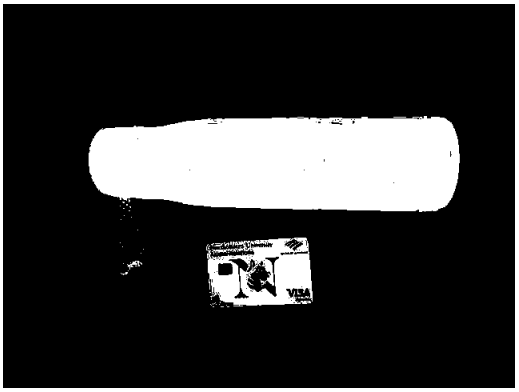


Fig 2. Threshold image of credit card and bottle.

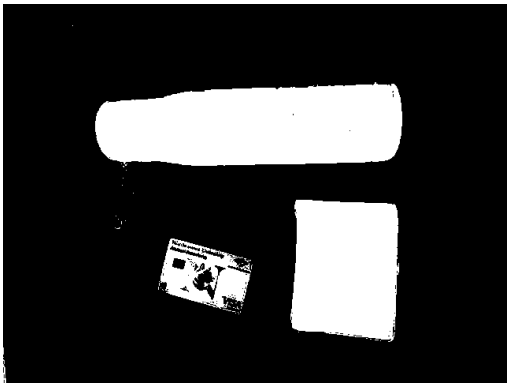


Fig 3. Threshold image of credit card, bottle and wallet.

**Task 2**: Clean up the binary image.

Fig 4. Cleaned up image of credit card

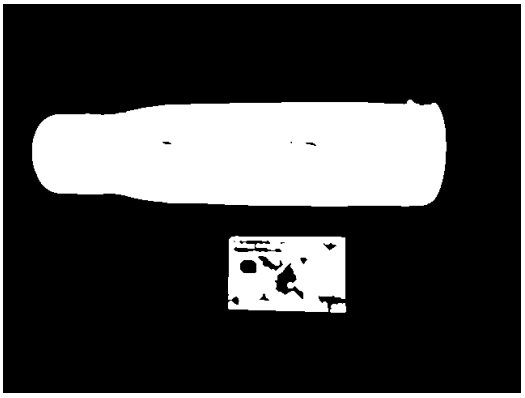


Fig 5. Cleaned up image of credit card and bottle.

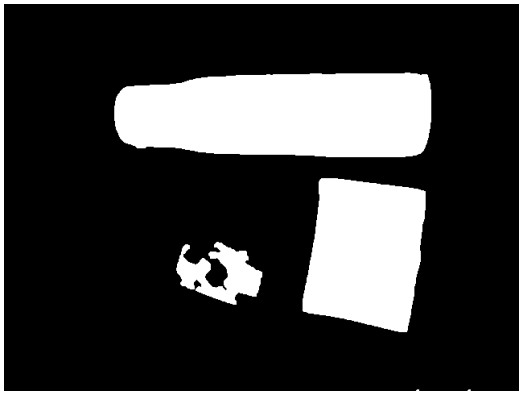


Fig 6. Cleaned up image of credit card, bottle and wallet.

**Task 3**: Segment the image into regions

Fig 7. Segmented image of credit card.

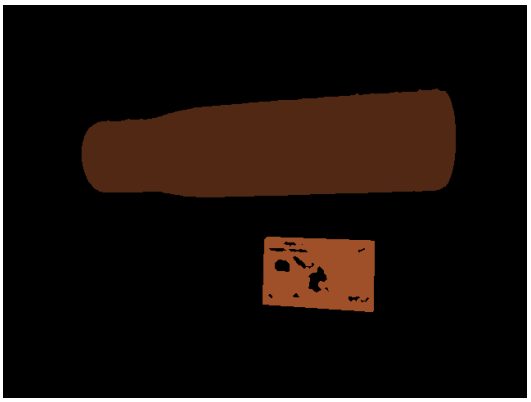


Fig 8. Segmented image of credit card and bottle.

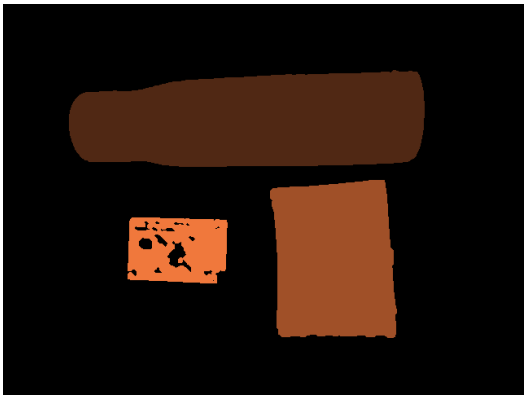


Fig 9. Segmented image of credit card, bottle and wallet.

For segmentation, we used brown color for the first region detected. For 2nd region, we have used a lighter shade of brown and finally for the 3rd region in the image we have used orange color to represent the region.

**Task 4**: Compute features for each major region

Fig 10. Central Moment axis and bounding box around bottle.

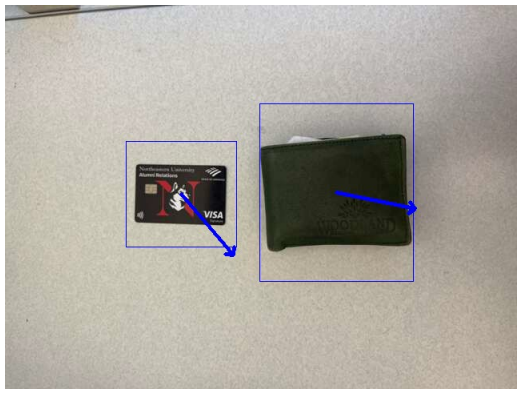


Fig 11. Central Moment axis and bounding box around credit card and wallet.

Fig 12. Central Moment and axis and bounding box around wallet, bottle and credit card.

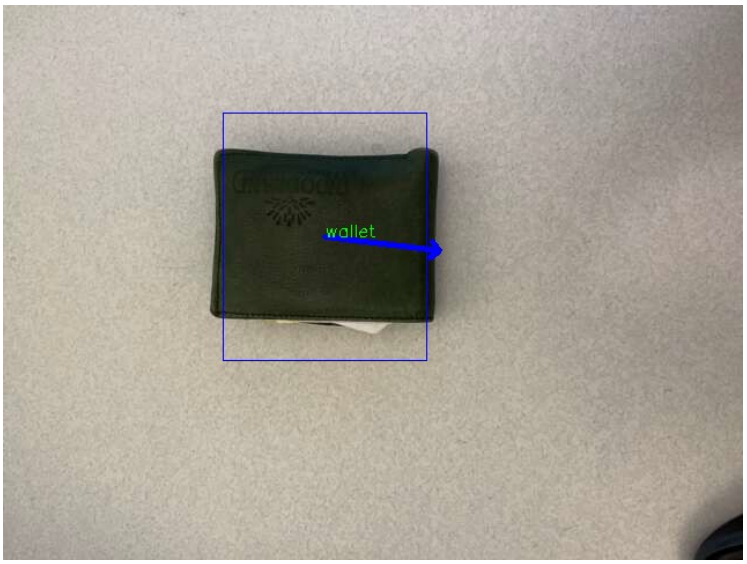
We have also computed the Hu Moments, Percent filled by the region in the bounding box and the width to height ration for each region and added each of these values into a csv file for future purpose. We used HuMoments function of OpenCV to denote the moments of region as a feature vector. Hu Moments features were used during the training mode of our dataset and based on these moments, the algorithm infers type of the object placed in front of the camera.

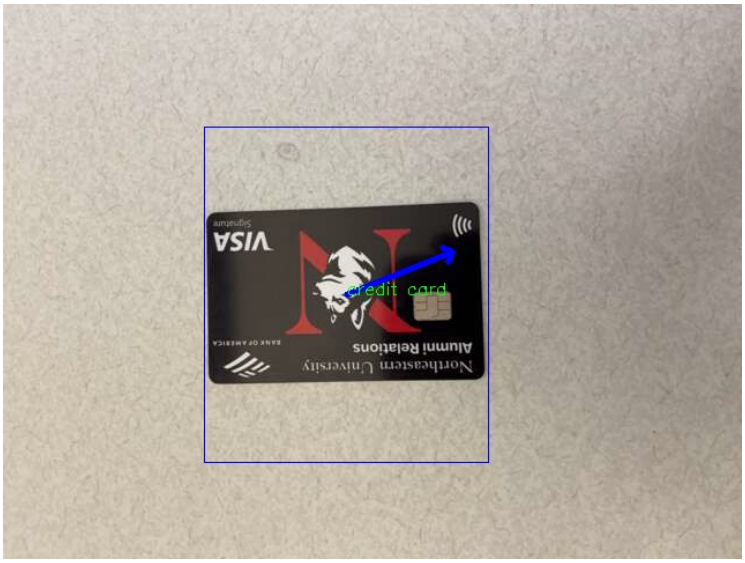
**Task 5**: Collect training data.
The training system of our algorithm is designed such that when the program is executed, it works in inference mode. As the user presses the “t” key, system enters into training mode and system displays the first region detected by the in the image, it calculates the moments of that particular region and stores it in a vector, it then prompts the user to enter a key identifying the type of object in the image. The Look-up table for the algorithm is as shown below:

*{'p', "pen"}, {'a', "alligator"}, {'h', "hammer"}, {'g', "glasses"}, {'r', "round"}, {'c', "credit card"}, {'b', "bottle"}, {'k', "key"}, {'m', "mouse"}, {'x', "binder clip"}, {'w', "wallet"}, {'y', "pliers"}*

Once the feature is calculated and label is assigned to a region, system displays the next region and ask the user to type in the key identifying the object. The procedure goes on until all the regions are labeled. As all the regions are labeled, the system switches back to inference mode and calculates the moments of the region present in the image, compares it with all the stored feature values, computes the distance metrics and the the distance metric falls within a predefined threshold, it assigns the label corresponding to that particular feature value to the region present in front of the image.
In this way, unknown objects present in front of the camera is recognized.

**Task 6**:

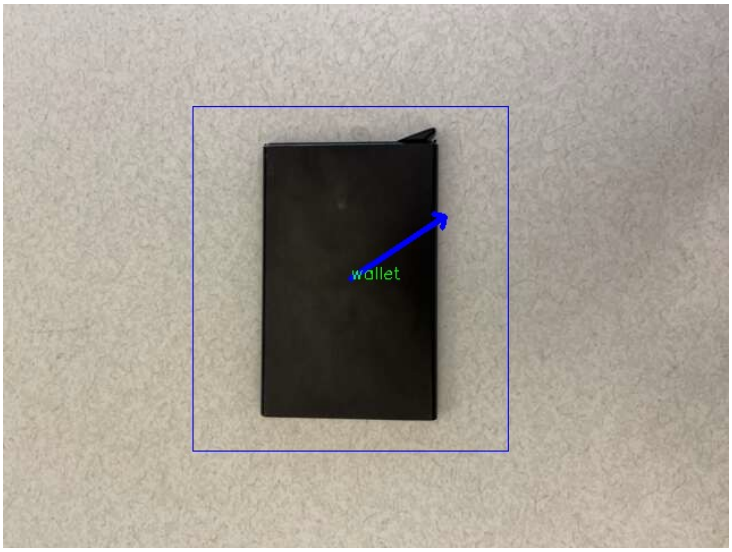


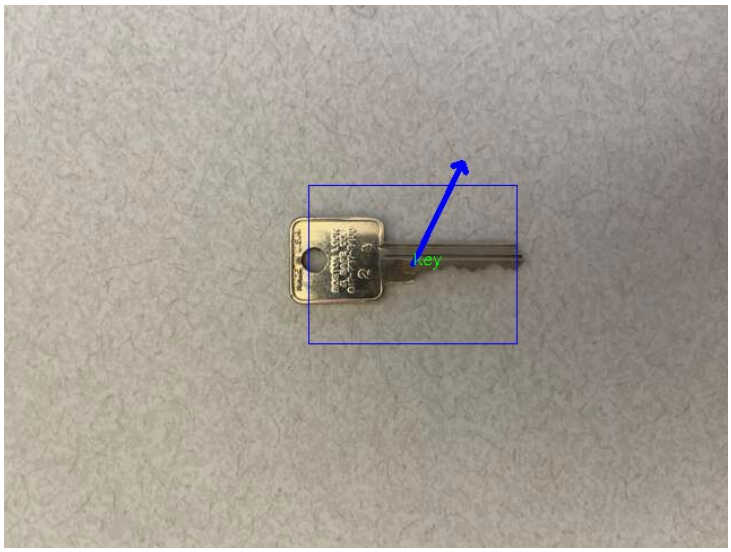





The image displayed above are used for training purpose, for every object displayed above, moment feature vector were calculated and stored in a vector which was then used to compare with the features of unknown object during the inference mode.

During the inference mode, the system was able to recognized unknown object and classify them into different categories. Some examples are shown below.






**Task 8**:
Video Link: https://drive.google.com/drive/folders/1QS2_HdQ8CJTiYeNehuhkXaQ8qirneRjK?usp=sharing

**Task 9**:
k-NN classifier was implemented as second training model. It classifies objects based on the majority class among their k nearest neighbors in the feature space. While calculating we considered a value of k as 2 which can be changed as per need and this algorithm doesn't involve the voting method.
We compared the result with the baseline training system and got similar results.