CS5330_Project3

A short description of the overall project in your own words. (200 words or less)

This project focuses on 2D object recognition. It is a real-time recognition system, which classifies the objects based on their shape features.

Under Inference mode, the system can detect at most 3 items at one time, and the largest 3 items will be detected. Under training mode, the system allows users to label each item and store it in a database.

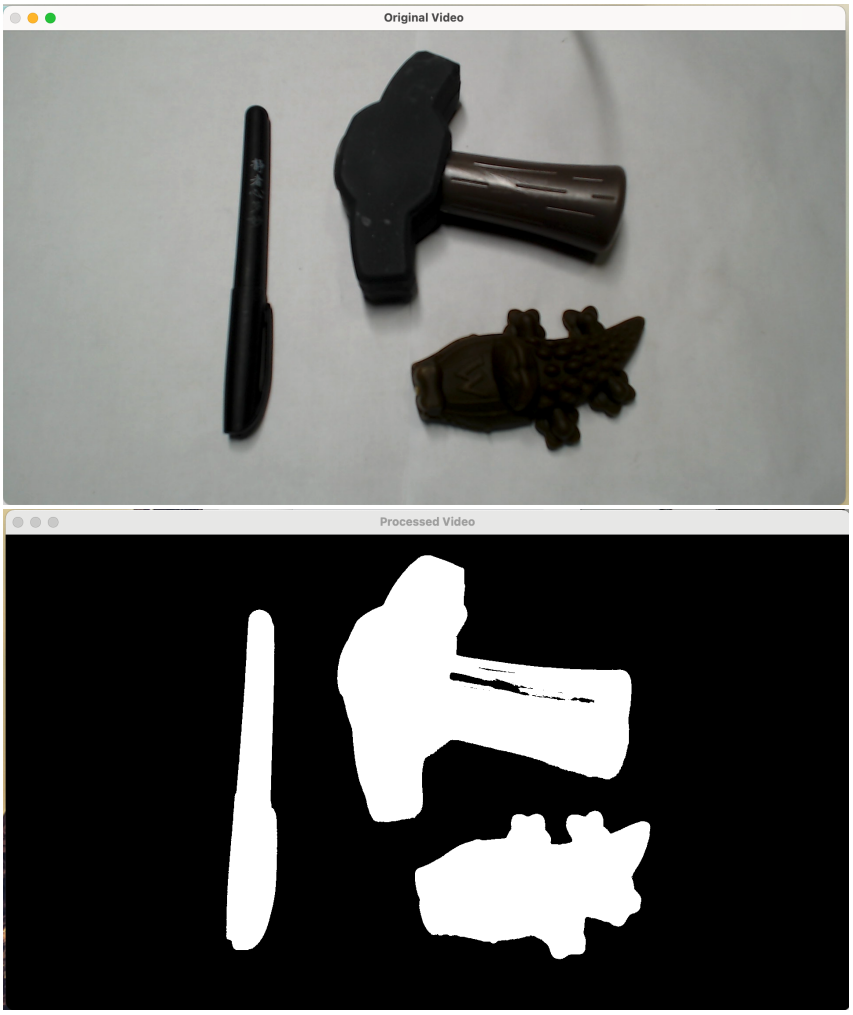
The system is able to recognize 14 2D objects: pen, alligator, hammer, glasses, round shape, cat, bone, key, mouse, binder clip, watch, credit card, spanner, pliers

Any required images along with a short description of the meaning of the image.

Task 1- Thresholded Objects

Convert the image to grayscale. Get the min and max element values for the background only(under my environment settings, the range is about 155 - 205).

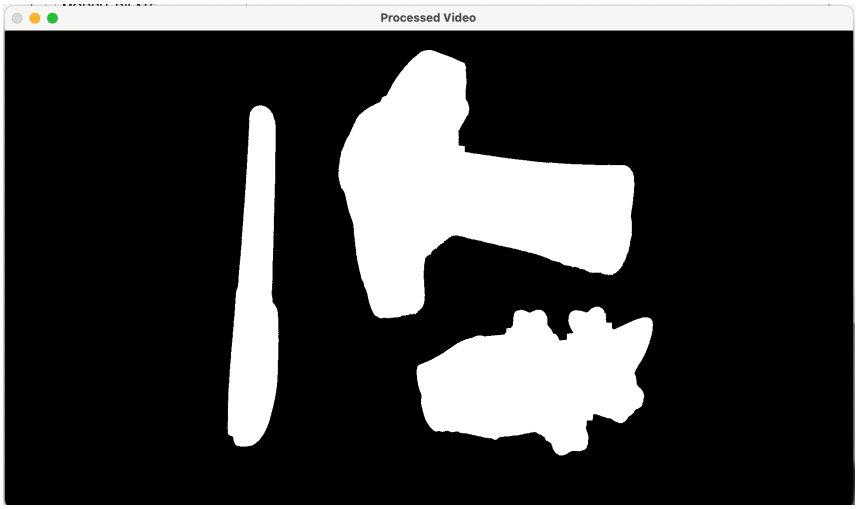
Set any pixels with a value smaller than the min value of the background to be foreground. After several tests, I set the threshold to be 130, which works best for my settings.



Task 2 - Cleaned Up Images

There is a hole in the hammer, so I decided to close it by the dilation of an image followed by erosion.

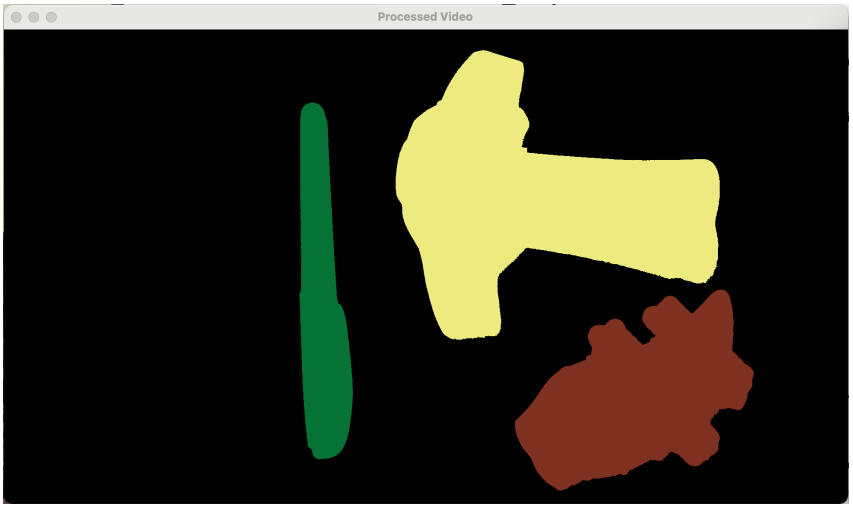
This is achieved by OpenCV built-in morphologyEx(). The size of the kernel is 25 x 25 to fix the hole without over closing.



Task 3 - Region Maps

The labeled regions are found by OpenCV connectedComponentsWithStats(), then I sorted the areas to get the largest N regions.

The system will ignore any regions that are too small(in my case any regions with an area less than 5000) and only recognize the largest N(3 in my case) regions.



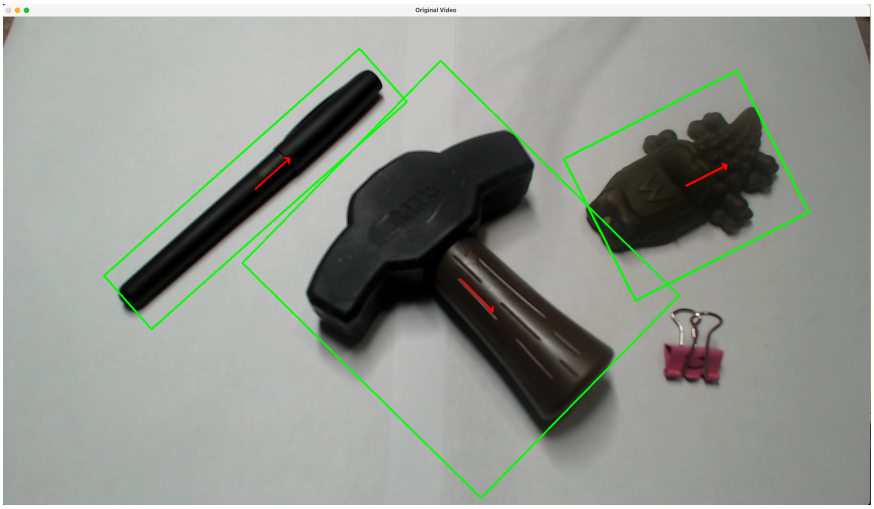
Task 4 - Regions with Axis of Least Central Moment and Bounding Box

The features are mainly calculated by two OpenCV functions, moments() and HuMoments().

moments() calculates a set of attributes of a region, for example, the centroid, size or area, moments, and central moments. However, none of the moments or central moments can satisfy the requirements of scale, translation, and rotation invariant. Therefore, we calculate HU moments based on the results of the central moments, and the final results are seven invariant moments.

The bounding box is also calculated based on the result attributes of moments(). The centroid and alpha(angle between the x-axis and least central x-axis) are used to build the new x-axis and y-axis. Then each point in the original coordinate is projected to the new coordinate to get the rotated bounding box.

Also, the system will only recognize the largest 3 objects so the binder clip is ignored, so the binder clip is ignored.



Task 5 - Training Systems

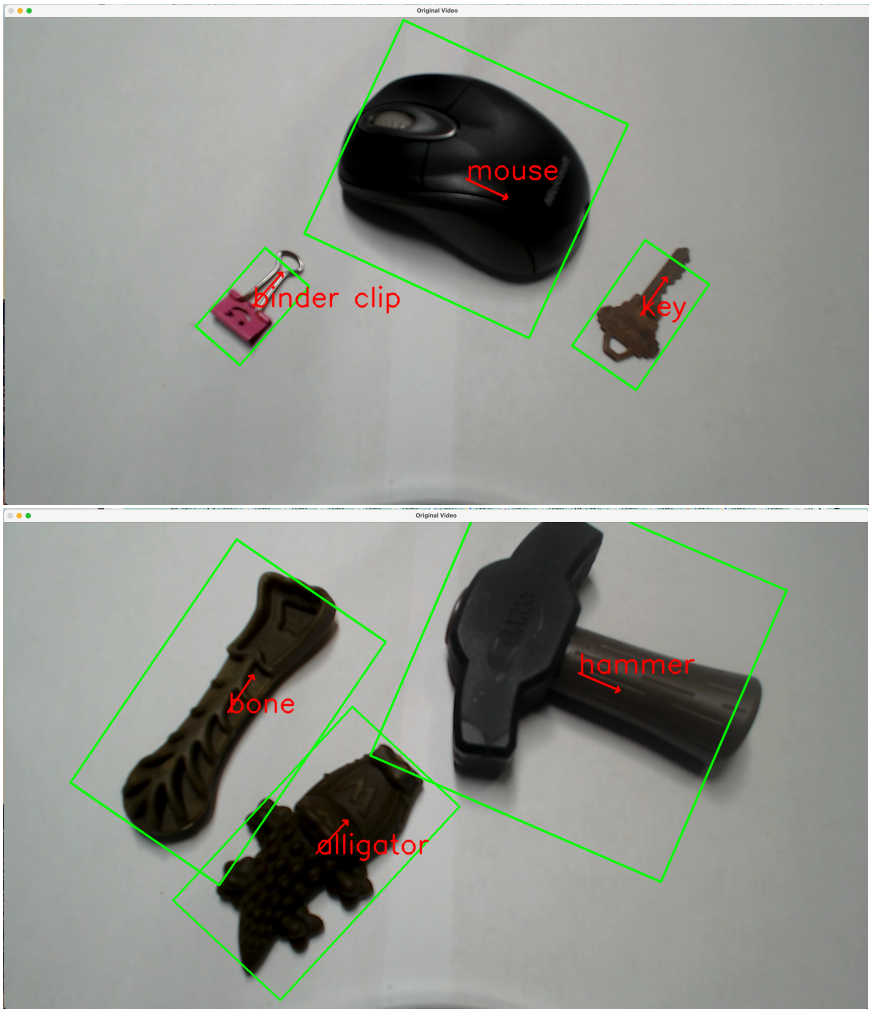
When the user pressed 't', the system will enter training mode. Under training mode, the system will display the object one by one, and ask the user to enter the name for this object. Then it will store the name and feature vector in the database.

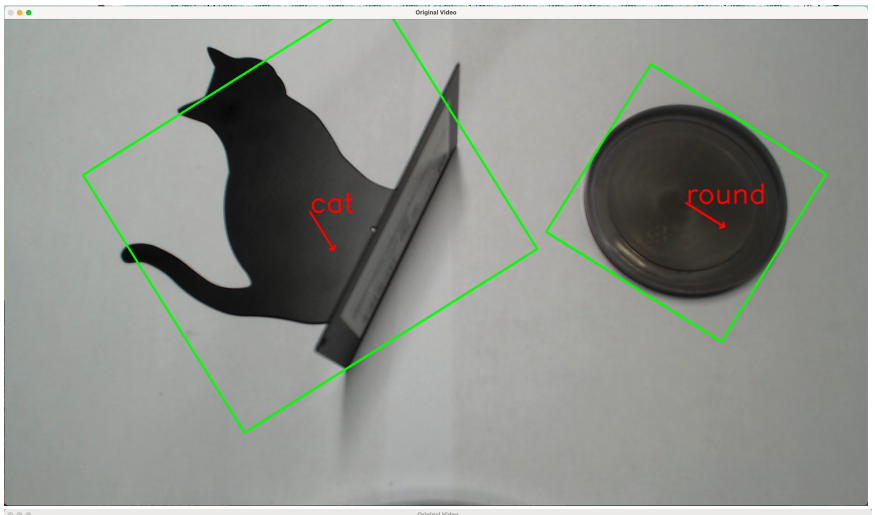
After displaying and collecting data for all the 3 objects, the system will switch back to the inference mode automatically and start classifying objects.

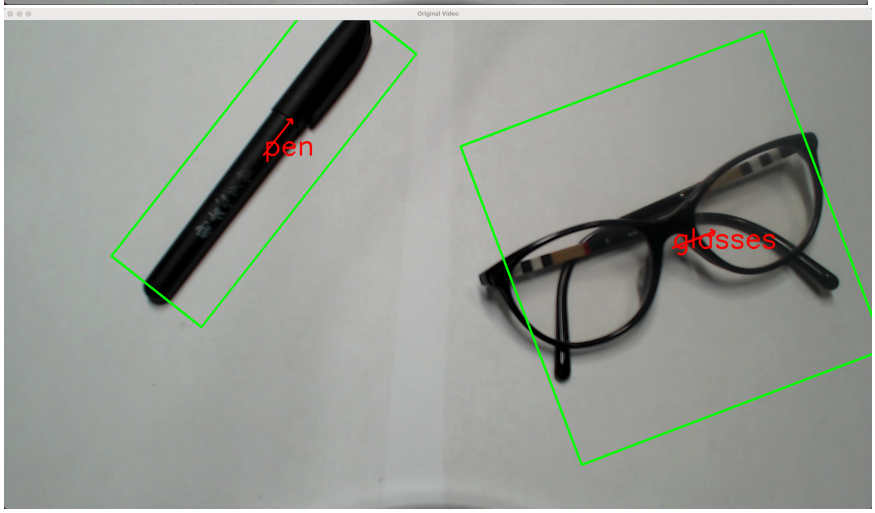
Here are some sample training data.

```
key,0.291413,0.0414217,0.013546,0.00559821,4.87439e-05,0.00113794,8.05639e-07
cat,0.225648,0.0104333,0.00167806,0.000469156,3.4916e-07,4.71615e-05,2.26653e-07
mouse,0.170047,0.00289416,1.15549e-05,2.44366e-07,-2.38909e-13,-1.13806e-08,3.33968e-13
key,0.291597,0.0415573,0.0135058,0.00557891,4.84204e-05,0.00113591,7.77819e-07
cat,0.224815,0.0102169,0.00163348,0.000467613,3.41166e-07,4.67082e-05,2.25005e-07
pen,1.03404,1.03515,0.0181808,0.0164644,0.000284848,0.0167179,-1.89688e-06
key,0.29057,0.0411078,0.0132966,0.00545512,4.64532e-05,0.00110463,7.70158e-07
glasses,0.403893,0.0884853,0.00519962,0.00229612,7.42407e-06,0.000679835,2.79772e-06
```

Task 6 - Objects with Label




Original Video
cat
round


Original Video
pen
glasses

Task 7 - Different Classifier

K_ Nearest Neighbor is implemented as a second classifier.

Task 8 - Performance of the system

Try to classify each object 10 times, and here are the results.

We can see that for some simple shapes (round, credit card, pen), the system works very well. However, complicated shapes like the cat are a little bit confusing.

Pairs with similar shapes(bone and watch, binder clip and key) are also easy to be confused with each other.

	pen	alligator	hammer	glasses	round	cat	bone	key	mouse	binder clip	watch	credit card	spanner	pliers
pen	10													
alligator		9				1								
hammer			8			2								
glasses				10										
round					10									
cat			2			7						1		
bone							8				2			
key								9						1
mouse									10					
binder clip								3		7				
watch							3				7			
credit card												10		
spanner													10	
pliers														10

Task 9 - Demo

https://youtu.be/fPHjQhJjmHE

A description and example images of any extensions.

Extension 1 - Recognize 3 objects at the same time

After getting the regions of every object, the system will first sort the regions by area, and the largest 3 regions will be kept. Then the system will loop the 3 objects to extract the region as a separate image, calculate the features/take training data, and display the class name in the video.

The image is displayed in task 6.

Extension 2 - Recognize more than 10 objects

This system can recognize 14 items. The image is displayed in task 6.

Extension 3 - Identify whether the item is in the database

When classifying the object, set a threshold. When the Euclidean distance is large between the object and other objects in the database, the system knows that this object is not in the database.

Therefore, no class name will be displayed and the user should enter training mode to add data about this object.

This picture shows that no name is displayed for the spanner because it is not in the database yet.

A short reflection of what you learned.

I learned about the workflow of object recognition and got familiar with a lot of related OpenCV functions. This project gave me a better understanding of how different light sources and the 2D shape of objects can influence the result of object recognition. I also learned a lot about how the moments of an object are calculated. The math is hard to understand but it helps me to understand what moments and features I am calculating.

Acknowledgment of any materials or people you consulted for the assignment.

- Morphology Transformations https://docs.opencv.org/3.4/d3/dbe/tutorial_opening_closing_hats.html
- Connected Component OpenCV Example https://docs.opencv.org/3.4/de/d01/samples_2cpp_2connected_components_8cpp-example.html
- Connected Component OpenCV Example https://gist.github.com/JinpengLI/2cf814fe25222c645dd04e04be4de5a6
- Intro to HuMoments https://zhuanlan.zhihu.com/p/117344473
- Intro to HuMoments https://cvexplained.wordpress.com/2020/07/21/10-4-hu-moments/
- Read and Write CSV Files https://www.gormanalysis.com/blog/reading-and-writing-csv-files-with-cpp/