CS 5330 Project 3 : Real-time 2D Object Recognition Report

Name: Hang Zhao
Email:zhao.hang1@northeastern.edu
Team member: Hang Zhao

Project Description:

This project focuses on real-time 2D object recognition using static images as input. The system processes images to identify objects based on their shape, size, and other features, even if they are rotated or resized. The process includes thresholding to separate objects from the background, morphological filtering to clean up noise, segmentation to detect objects, and feature extraction to describe their properties.
For classification, we first implemented a Nearest Neighbor (NN) approach, which finds the closest match based on a scaled Euclidean distance metric. Later, we extended it with K-Nearest Neighbors (KNN) to improve accuracy. Additionally, we developed an automatic learning system that detects unknown objects and adds them to the database for future recognition.
The system was tested with multiple images, and performance was evaluated using a confusion matrix to compare true vs. predicted labels. While NN performed better than KNN due to limited training data, the system improves over time as more labeled objects are added.
This project demonstrates how computer vision and machine learning can be combined for real-time object recognition, making it an adaptive and scalable system.

Images and description

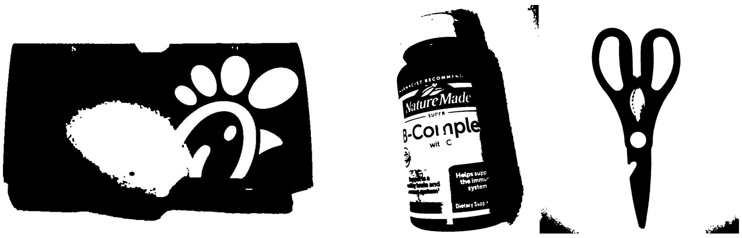
Task 1 Threshold the input video:
We use Otsu thresholding and adaptive Gaussian thresholding to segment objects from the background. Otsu's method is suitable for uniform backgrounds, while adaptive thresholding performs better when the lighting is uneven. To improve the effect, we first convert to grayscale, then Gaussian blur to remove noise, and finally apply the thresholding algorithm. The processed image clearly highlights the object, which is convenient for subsequent recognition.

origianl pictures

processed pictures

Task 2 Clean up the binary image:

We use morphological closing (filling holes) and opening (removing noise) to clean up the threshold image. Closing first dilates and then erodes, which can fill small holes inside the object, while opening first erodes and then dilates, which can remove isolated noise points. After this processing, the object outline is clearer, avoiding misidentification and improving the accuracy of subsequent recognition.



cleaned up pictures

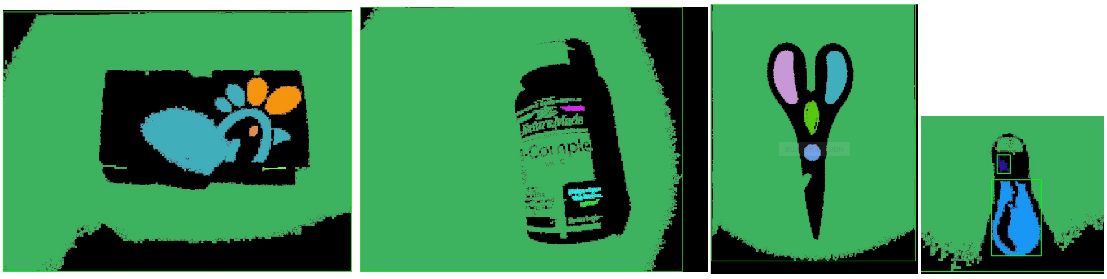
Task 3 Segment the image into regions

We use Connected Component Analysis (CCA) to segment the image and assign different random colors to each region to ensure that they are clearly visible. The background remains black to avoid interference, and noisy areas smaller than 500 pixels are filtered out to ensure that only valid objects are retained. In this way, we can intuitively see different object regions, which is convenient for subsequent feature extraction and classification.

region pictures

Task 4 Compute features for each major region

We calculated features such as Hu moment, aspect ratio, and filling rate to describe the shape and proportion of each region. Among them, Hu moment can maintain rotation, scaling and displacement unchanged, aspect ratio reflects the shape of the object, and filling rate reflects the proportion of the object in its bounding rectangle. We also drew the minimum bounding rectangle on the image to intuitively show the posture of the object and facilitate subsequent classification.



featured pictures

Task 5 Collect training data

This training system automatically reads the extracted feature data and then displays the corresponding images one by one. Users can manually enter labels. After clicking the "Save Label" button, the system will save the labels and feature data to training_data.csv and then jump to the next image. If the image is of an unknown category, the user can enter "unknown" or other appropriate labels, and these data can be used to improve the classification model later. After all images are labeled, the system will automatically save the data for subsequent classification tasks.

A	B	C	D	E	F	G	H	I	J	K	L
Label	AspectRat	PercentFi	Hu1	Hu2	Hu3	Hu4	Hu5	Hu6	Hu7		
scissors	0.7943217	0.9448052	0.1682070	0.0012507	1.7782077	6.3899580	-2.15E-12	2.2593057	8.075710424669829e-14		
clip	0.6567796	0.7142700	0.1879064	0.0067304	0.0005112	0.0001003	2.2711262	8.2063564	1.2305180649964011e-09		
clip	0.7241379	0.5199096	0.2257480	0.0150175	0.0020164	9.3764182	4.1085910	-8.71E-07	-4.06E-08		
clip	0.9365853	0.8229720	0.1705660	0.0001047	5.2062656	5.2670273	5.2434254	-5.24E-08	6.96980454266288e-11		
pill	1.2195313	0.9372017	0.1667397	0.0007393	3.5545530	7.2358592	3.5420138	-6.71E-09	9.595084679828909e-13		
cube	0.8384279	0.9682864	0.1679997	0.0007687	7.1552634	1.2908713	1.8177024	2.3576111	-3.48E-14		
chickfila	1.3431269	0.8686958	0.1831438	0.0055011	5.6774022	3.8545930	1.5433637	-1.94E-07	5.489366204562234e-11		
unknown	1.2	0.85	0.000123	0.000456	0.000789	1.2e-05	3.4e-05	5.6e-05	7.8e-05		

Task 6 Classify new images

We use Scaled Euclidean Distance for classification, which calculates the feature distance between the new object and each object in the training data and normalizes it to ensure that the influence of different features is balanced. The category with the smallest distance is the final classification result. If the minimum distance exceeds the set threshold (such as 2.0), it is marked as "unknown", indicating that the object does not belong to a known category. In the report, we provide an image with a label, fill rate, and aspect ratio for each category to clearly show the classification results and ensure recognition accuracy.

Task 7 Evaluate the performance of your system

The system is good at classifying most of the objects. But it sometimes confuses the objects with similar shapes, such as chickfila and cube, clip and cube.

	true label					
		chickfila	pill	cube	scissors	clip
classify label	chickfila	2		1		
	pill		3			
	cube			3		
	scissors				3	
	clip			1		2

Task 8 Capture a demo of your system working

Due to the problems with my Windows computer system, this part of the content is temporarily processed using colab. Please understand.

https://youtu.be/kY1Ex9EYtHA

Task 9 Implement a second classification method

I choose KNN(k=3) classifier, the reason is that because it typically provides better noise handling and generalization by considering multiple nearest neighbors instead of relying on a single match like Nearest Neighbor (NN).

How I implement KNN:

a. Extracted Aspect Ratio, Percent Filled, and Hu Moments as features.
b. Normalized features to ensure consistent comparison.
c. Used K=3 and Euclidean distance for classification.
d. Assigned the majority label among the 3 nearest neighbors.

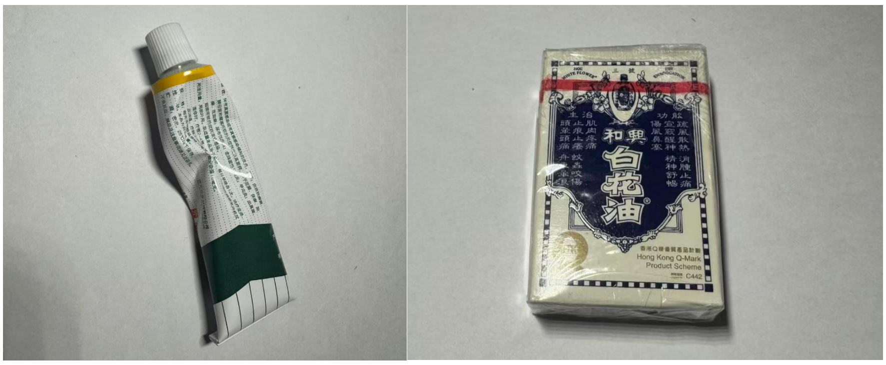
Comparison:

Nearest Neighbor Accuracy: 33.33%

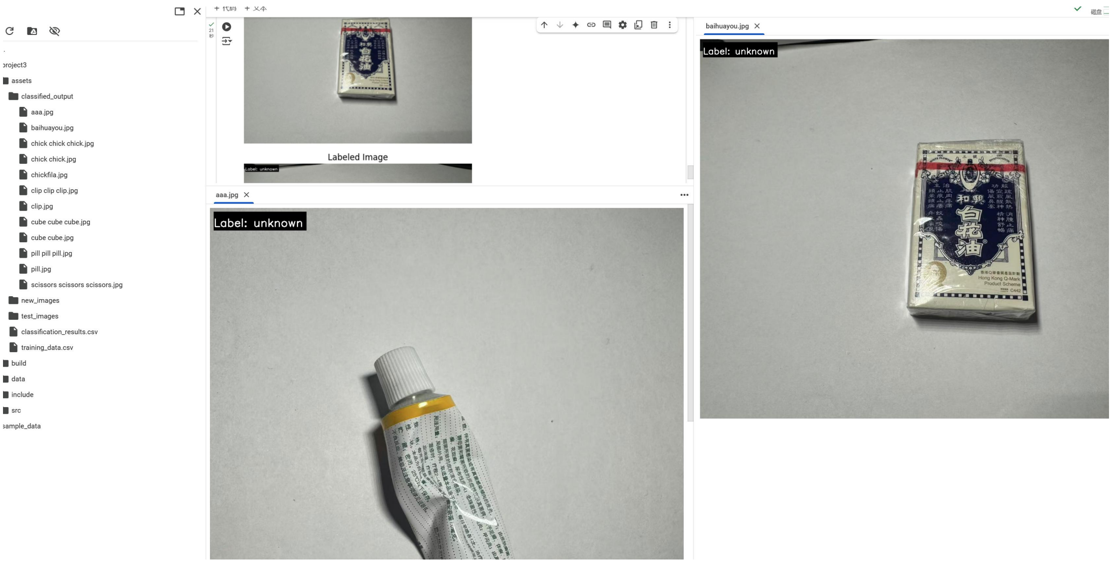
KNN (K=3) Accuracy: 6.67%

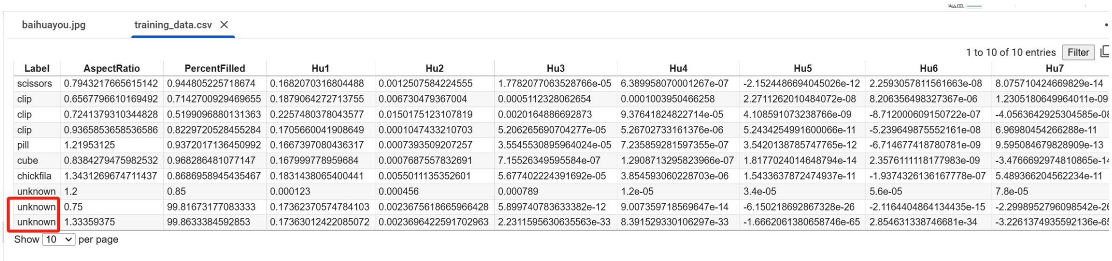
Extensions:

I implemented an automatic learning system that detects unknown objects, extracts their features, and adds them to the database for future recognition. If an object isn’t recognized, it gets labeled as "unknown" and saved, so we can update it later. This makes the system smarter over time without needing manual retraining. The more objects it sees, the better it gets at recognizing them!



new unknown objects




baihuayou.jpg training_data.csv

1 to 10 of 10 entries Filter

Label	AspectRatio	PercentFilled	Hu1	Hu2	Hu3	Hu4	Hu5	Hu6	Hu7
scissors	0.7943217665615142	0.944805225718674	0.1682070316804488	0.0012507584224555	1.7782077063528766e-05	6.389958070001267e-07	-2.1524486694045026e-12	2.2593057811561663e-08	8.075710424669829e-14
clip	0.6567796610169492	0.7142700929469655	0.1879064272713755	0.006730479367004	0.0005112328062654	0.0001003950466258	2.2711262010484072e-08	8.206356498327367e-06	1.2305180649964011e-09
clip	0.7241379310344828	0.5199096880131363	0.2257480378043577	0.0150175123107819	0.0020164886692873	9.37641824822714e-05	4.108591073238766e-09	-8.712000609150722e-07	-4.0563642925304585e-0
clip	0.9365853658536586	0.8229720528455284	0.1705660041908649	0.0001047433210703	5.206265690704277e-05	5.26702733161376e-06	5.2434254991600066e-11	-5.239649875552161e-08	6.96980454266288e-11
pill	1.21953125	0.9372017136450992	0.1667397080436317	0.0007393509207257	3.5545530895964024e-05	7.235859281597355e-07	3.5420138785747765e-12	-6.714677418780781e-09	9.595084679828909e-13
cube	0.8384279475982532	0.968286481077147	0.167999778959684	0.0007687557832691	7.15526349595584e-07	1.2908713295823966e-07	1.8177024014648794e-14	2.3576111118177983e-09	-3.4766692974810865e-1
chickfila	1.3431269674711437	0.8686958945435467	0.1831438065400441	0.0055011135352601	5.677402224391692e-05	3.854593060228703e-06	1.5433637872474937e-11	-1.9374326136167778e-07	5.489366204562234e-11
unknown	1.2	0.85	0.000123	0.000456	0.000789	1.2e-05	3.4e-05	5.6e-05	7.8e-05
unknown	0.75	99.81673177083333	0.17362370574784103	0.0023675618665966428	5.899740783633382e-12	9.007359718569647e-14	-6.150218692867328e-26	-2.1164404864134435e-15	-2.2998952796098542e-2
unknown	1.33359375	99.8633384592853	0.17363012422085072	0.0023696422591702963	2.2311595630635563e-33	8.391529330106297e-33	-1.6662061380658746e-65	2.854631338746681e-34	-3.2261374935592136e-6

Show 10 per page

Reflection:

Through this project, I learned how to process and analyze images for object recognition using computer vision techniques. I got hands-on experience with thresholding, segmentation, and feature extraction, and saw how different classification methods like Nearest Neighbor and KNN perform in real-world scenarios. One key takeaway was that more data improves accuracy, as KNN struggled with limited training examples. Implementing an automatic learning system was a great way to make the model more adaptive. Overall, this project showed me how computer vision and machine learning work together to build smart, real-time recognition systems.

Acknowledgement:

I would like to thank our teaching assistants and classmates for their valuable support and guidance throughout this project. Their help in troubleshooting technical issues and providing feedback was essential in completing our work.

I also want to express our gratitude to our professor for providing clear explanations and insightful examples that deepened our understanding of computer vision. The resources and tutorials shared during the course were incredibly helpful in learning OpenCV and applying it effectively to this project.