cs5330 realTime 2D Object Recognition

CS5330 Pattern Recognition & Computer Vision

NEU 2023 Fall

Instructor: Bruce Maxwell

Student: Shi Zhang

Last modified: 12/09/2023

Project Report

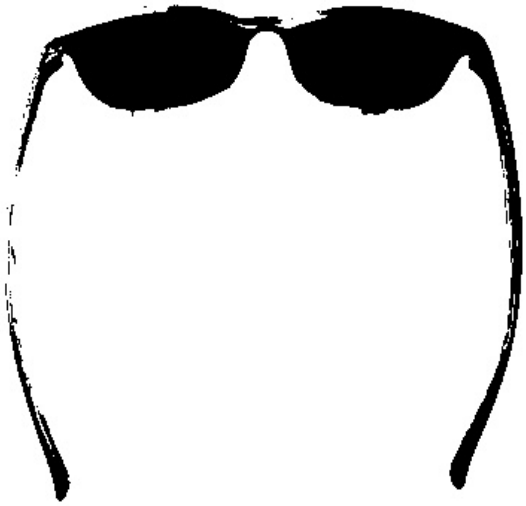
1. Introduction

This project focuses on creating an Object Recognition (OR) system that identifies objects in a given video input. Leveraging computer vision techniques and algorithms, the system processes video frames, and thresholds, and cleans images, segments them into regions, and computes features. Using these features and a database of known objects, the system classifies objects in real time. This report delves into the technical details, visual demonstrations, and reflections on this journey.

2. Visual Demonstrations

- **Task 1: Thresholded Objects**

Original video frame image of Sunglass.



Threshold Output of video frame image of Sunglass.

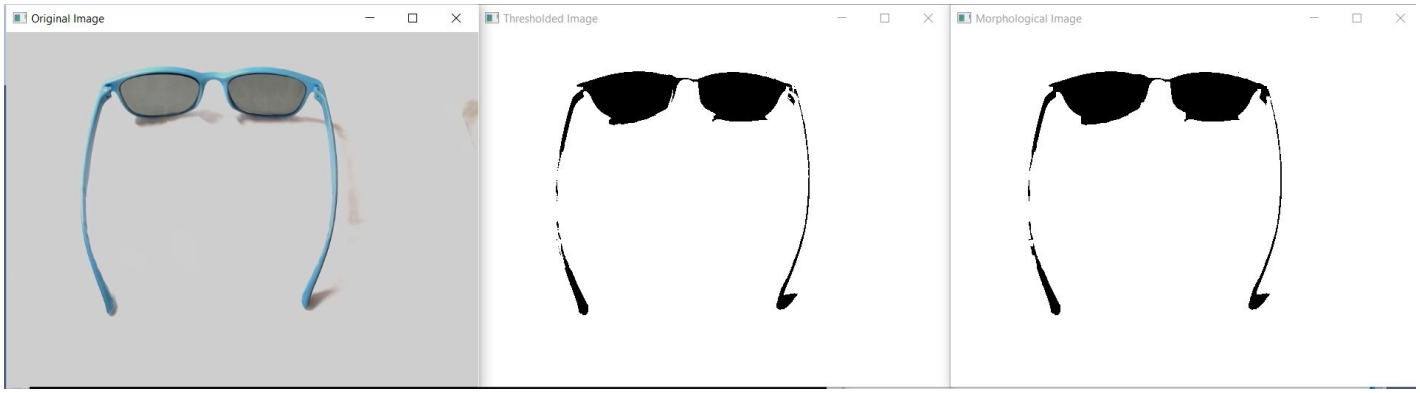
Original video frame image of a hollow ball.

Threshold Output of video frame image of the hollow ball.

Original video frame image of a Mouse.

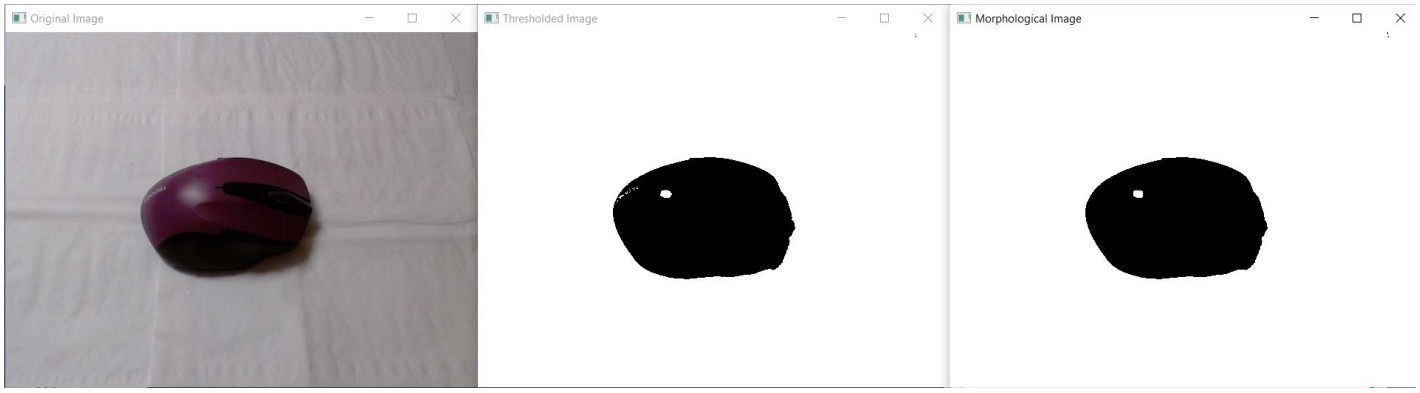
Threshold Output of video frame image of the Mouse.

- **Task 2: Cleaned Images**

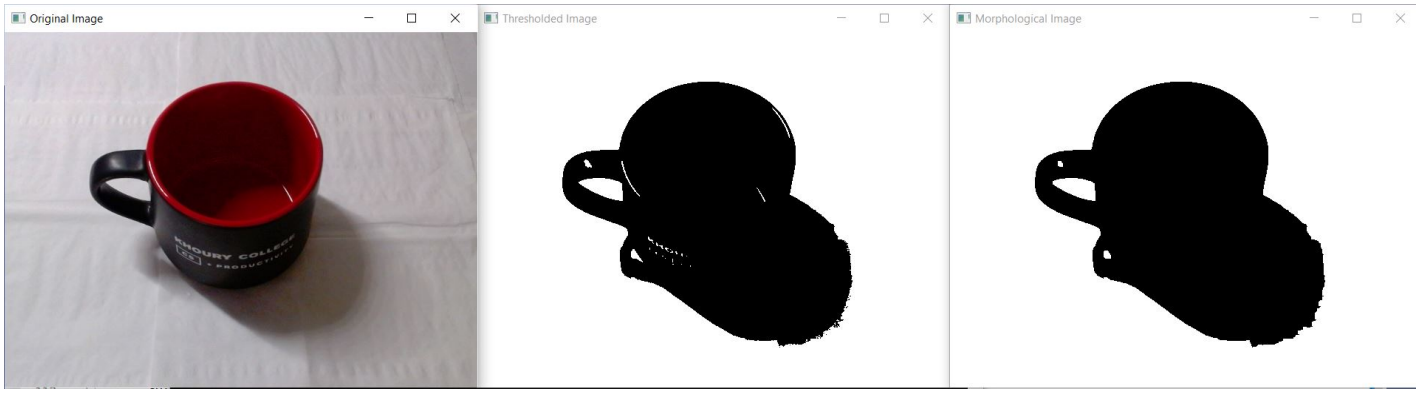


Output display for Original, threshold, and morphological cleanup for a pair of sunglass.

Output display for Original, threshold, and morphological cleanup for a hollow ball.

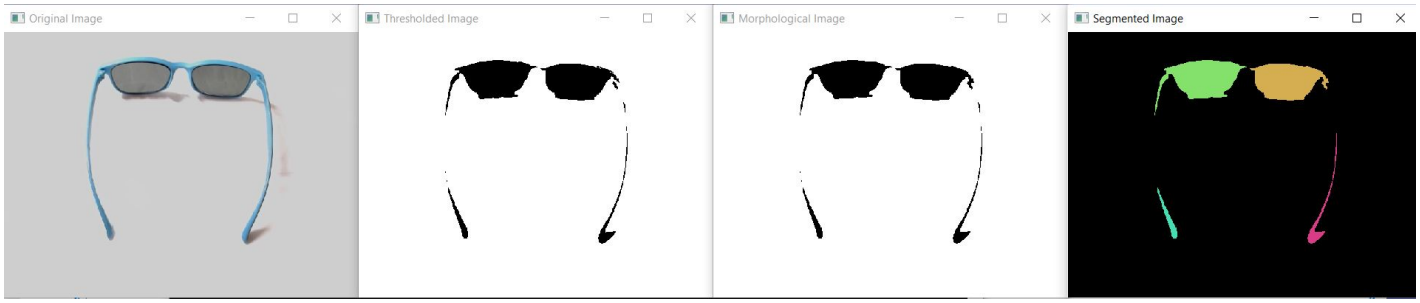


Output display for Original, threshold, and morphological cleanup for a mouse.



Output display for Original, threshold, and morphological cleanup for a mug.

- **Task 3: Region Maps**



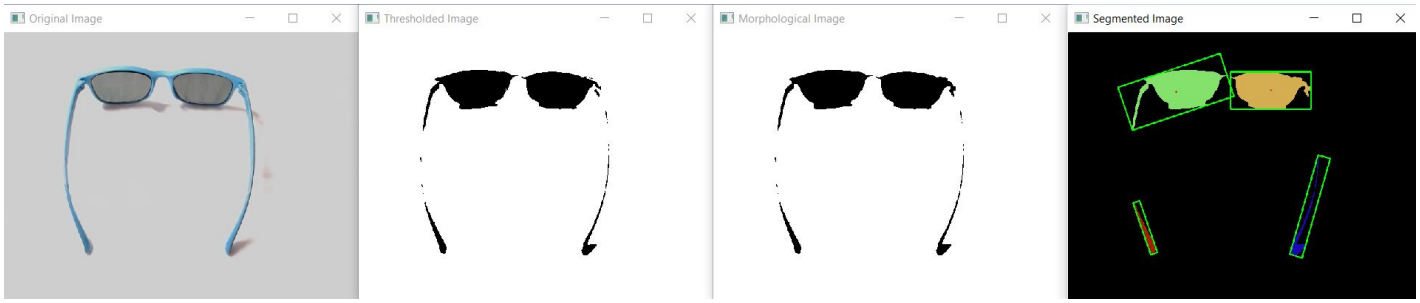
Output display for Original, threshold, morphological cleanup, and segmented region map for a pair of sunglass.

Output display for Original, threshold, morphological cleanup, and segmented region map for a hollow ball.

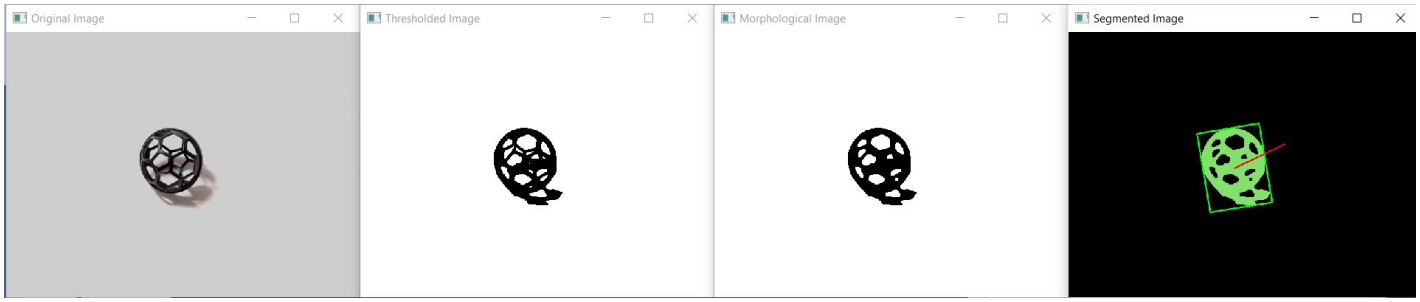
Output display for Original, threshold, morphological cleanup, and segmented region map for a mouse.

Output display for Original, threshold, morphological cleanup, and segmented region map for a mug.

- **Task 4: Feature Computation**



Output display for Original, threshold, morphological cleanup, and segmented region map (with the feature vector overlaid) for a pair of sunglasses.



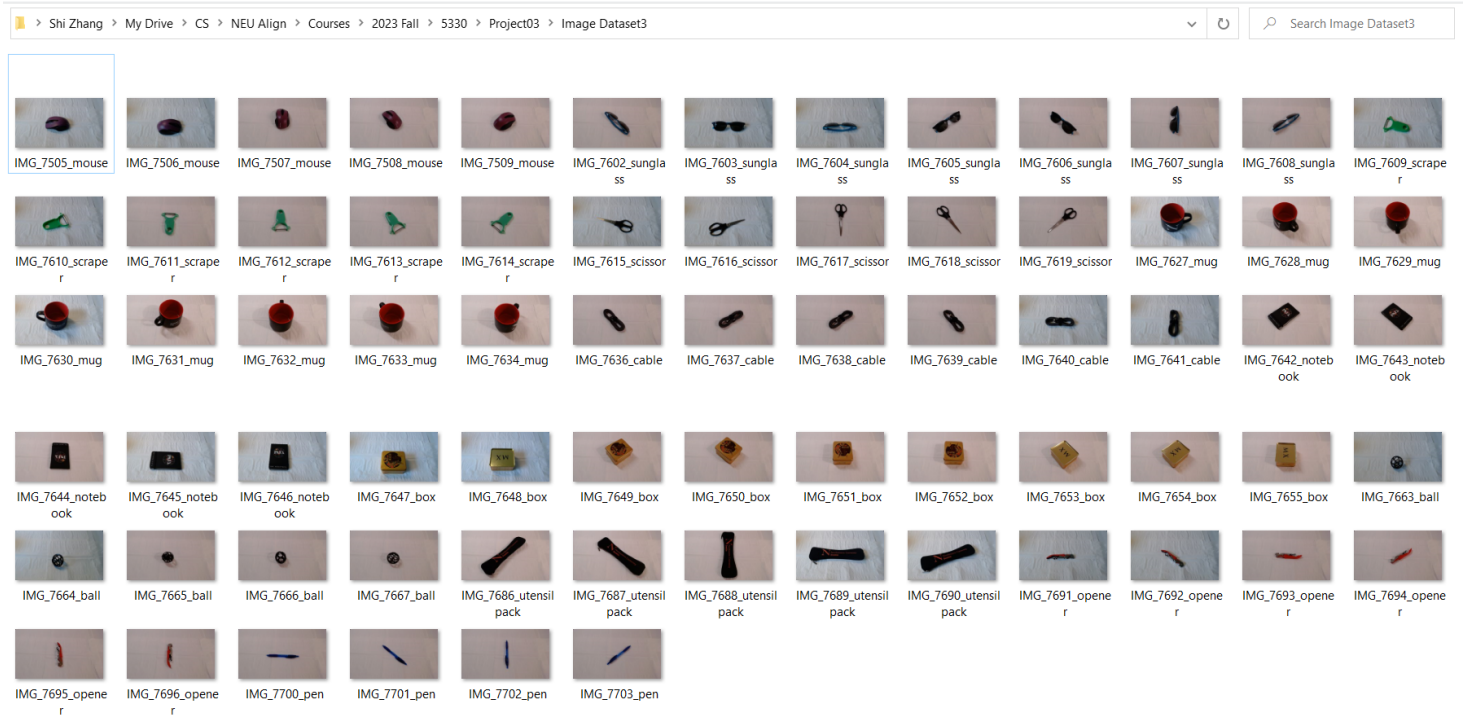
Output display for Original, threshold, morphological cleanup, and segmented region map (with the feature vector overlaid) for a hollow ball.

Output display for Original, threshold, morphological cleanup, and segmented region map (with the feature vector overlaid) for a mouse.

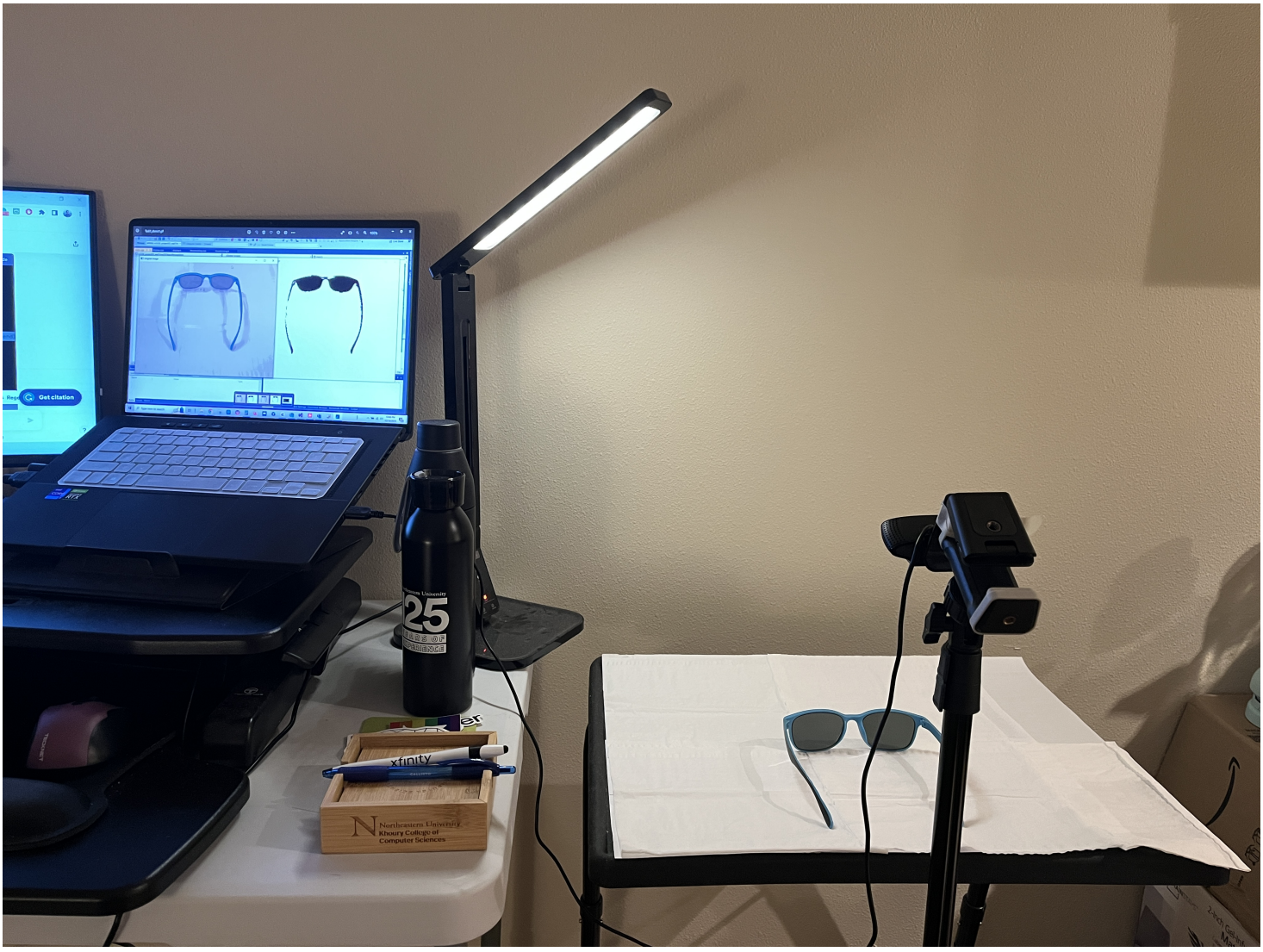
Output display for Original, threshold, morphological cleanup, and segmented region map (with the feature vector overlaid) for a mug.

- **Task 5: Training Data Collection**

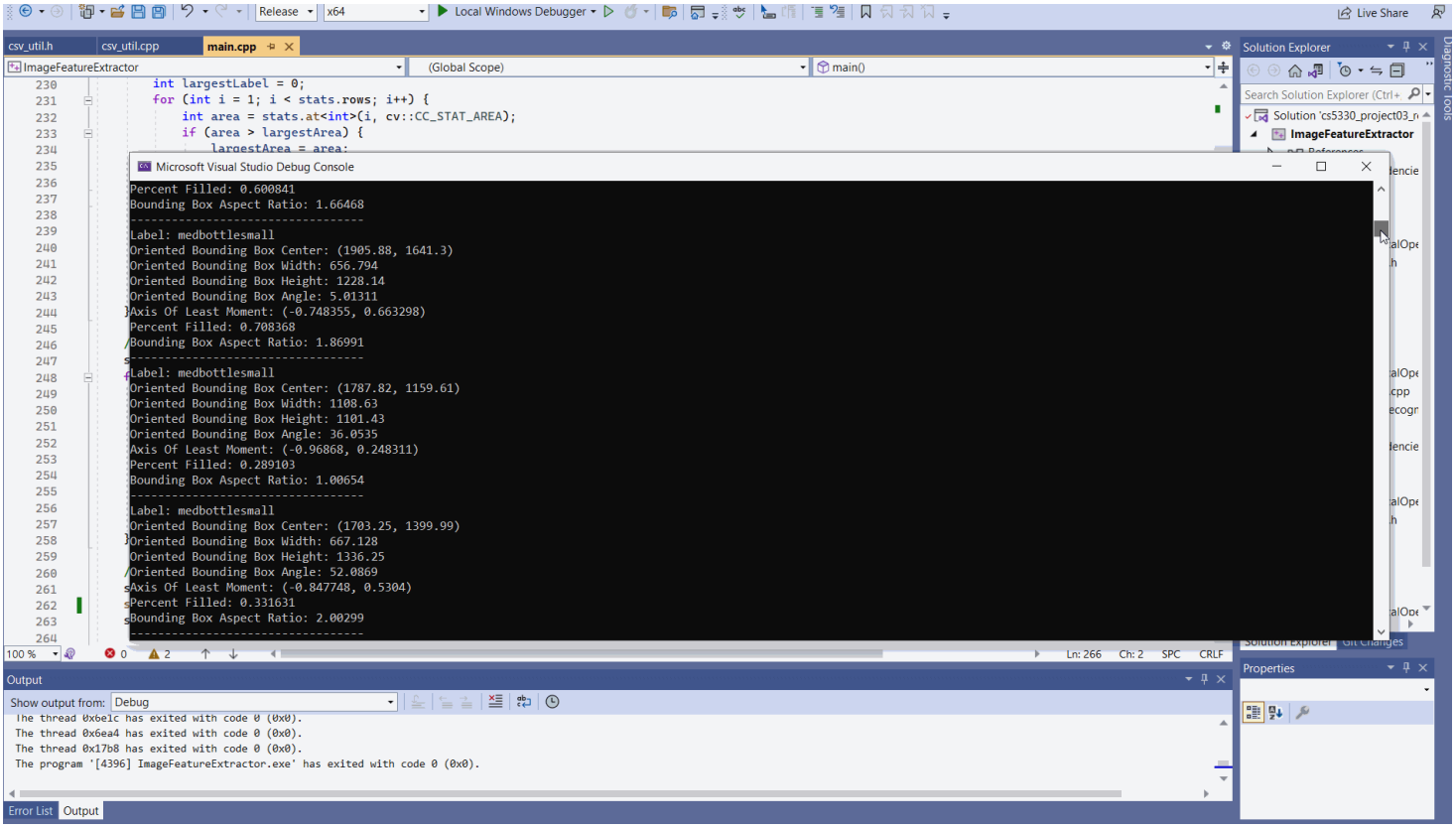
The training data is collected as a local image directory.



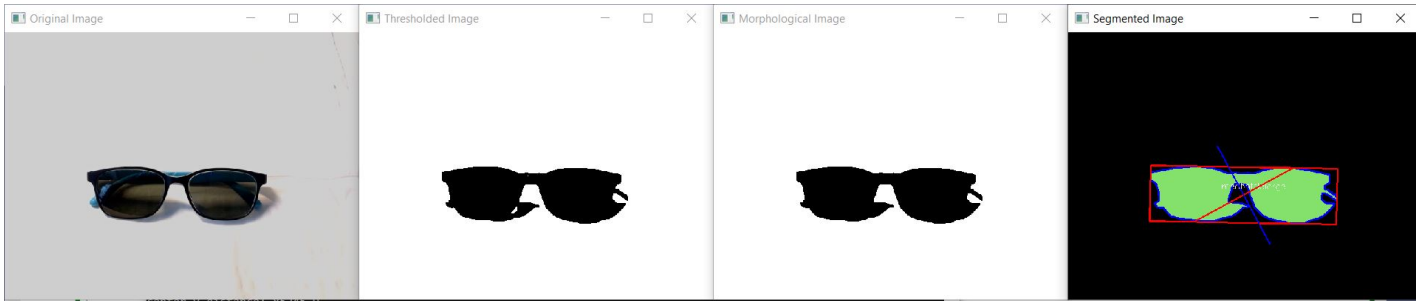
Each image was captured in a workspace environment.



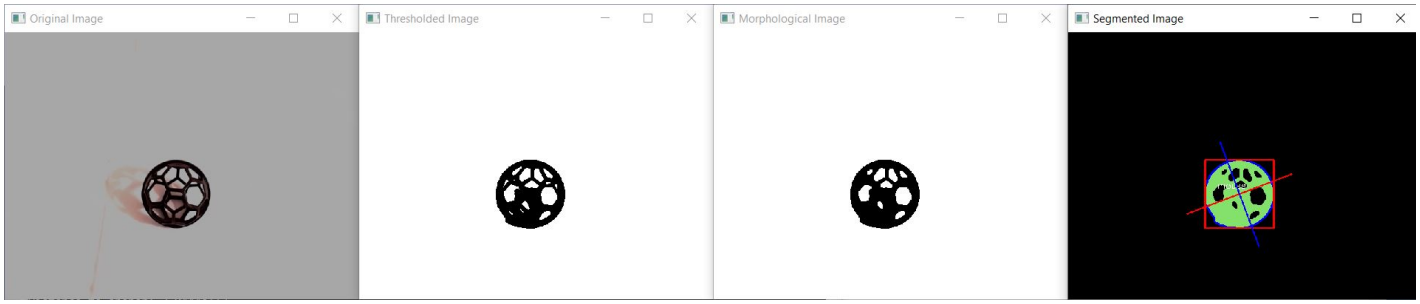
Then running the project ImageFeatureExtractor main program, it will store feature vectors for each image in a local CSV file.



- **Task 6: New Image Classification**

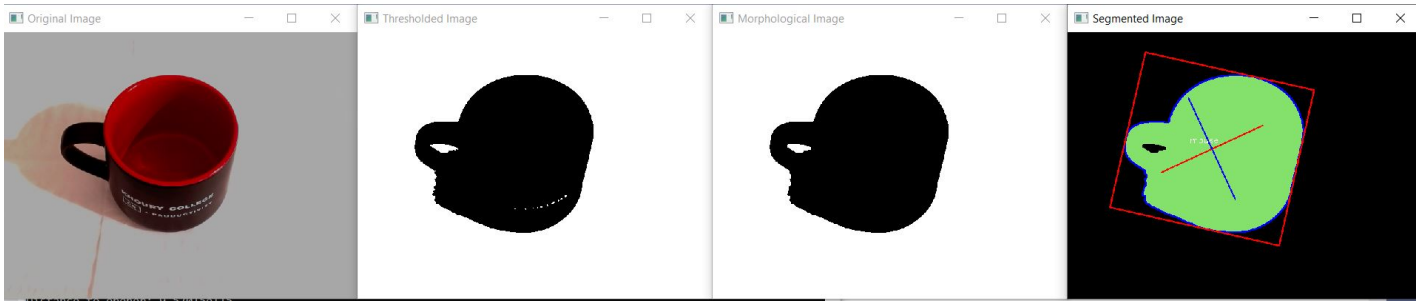


Output display for Original, threshold, morphological cleanup, and segmented region map (with the feature vector and text label overlaid) for a pair of sunglasses.



Output display for Original, threshold, morphological cleanup, and segmented region map (with the feature vector and text label overlaid) for a hollow ball.

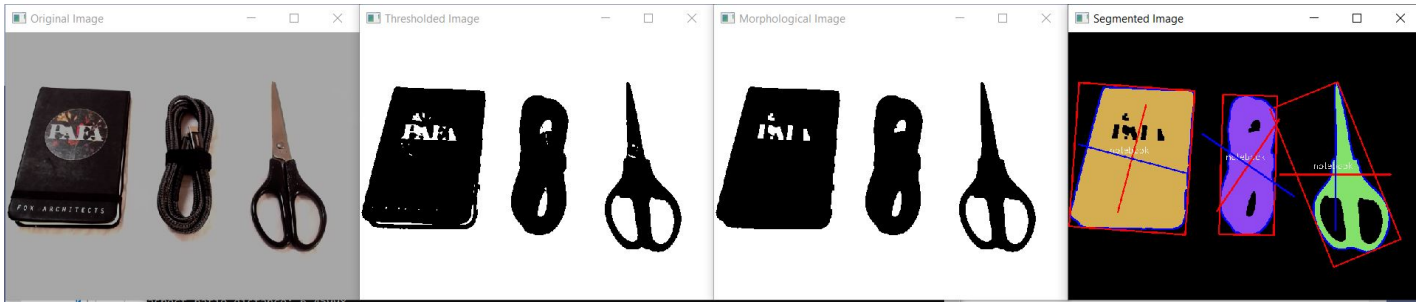
Output display for Original, threshold, morphological cleanup, and segmented region map (with the feature vector and text label overlaid) for a mouse.



Output display for Original, threshold, morphological cleanup, and segmented region map (with the feature vector and text label overlaid) for a mug.

- **Task 7: KNN Classifier**

Testing multiple object recognition using KNN Classifier.



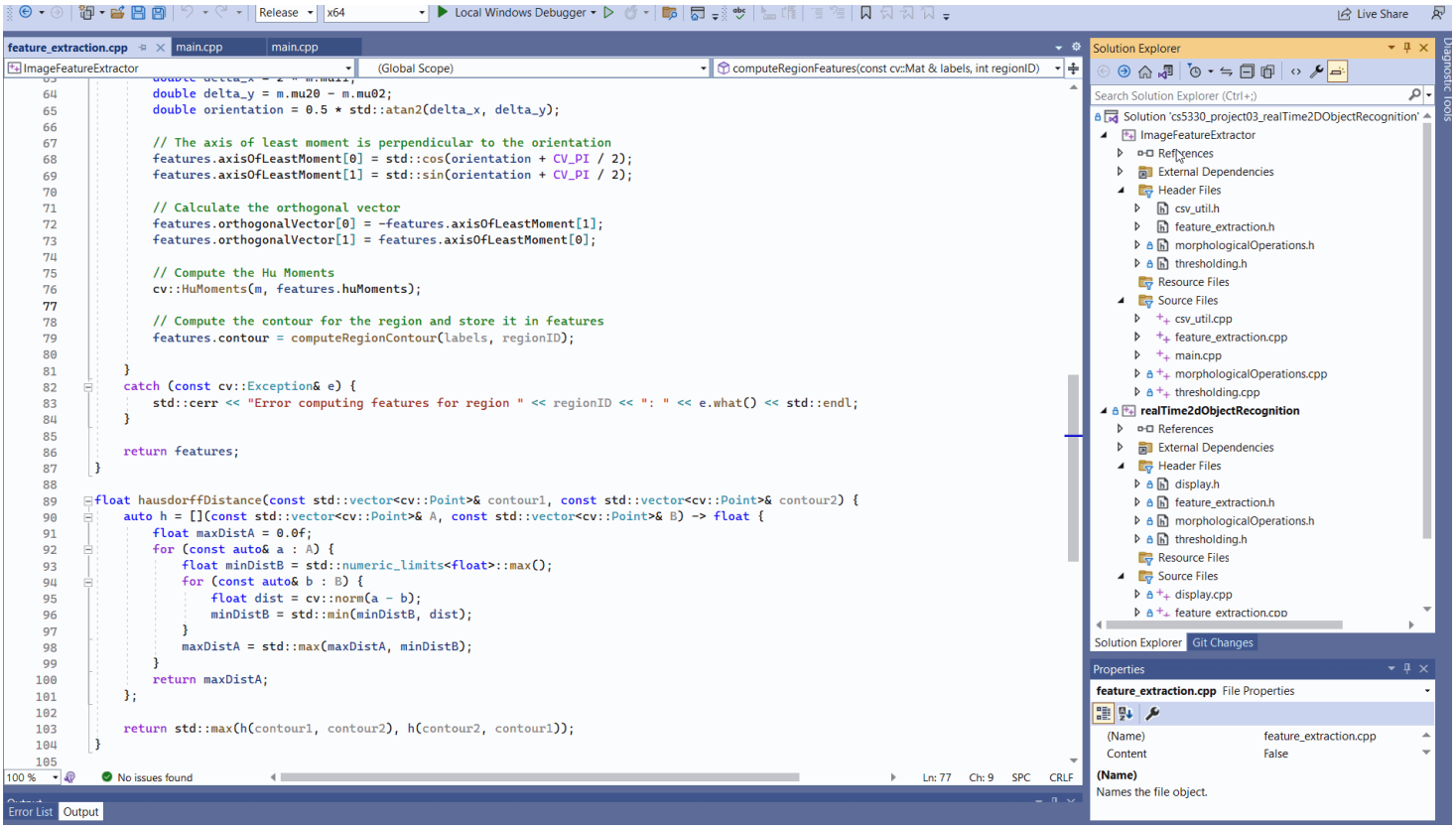
Testing multiple object recognition using KNN Classifier.

- **Task 8: Confusion Matrix for Evaluation**

Truth ->	mouse	sunglass	scraper	scissor	mug	cable	notebook	box	ball	utensil pack	opener	pen
mouse	3	0	1	0	0	0	0	0	0	0	0	0
sunglass	0	5	0	0	0	0	0	0	0	0	0	0
scraper	0	0	7	2	0	0	0	0	0	0	0	0
scissor	0	0	0	4	0	2	0	0	0	0	0	0
mug	0	0	0	0	5	0	0	0	0	0	0	0
cable	0	0	0	2	0	3	0	0	0	0	0	0

Truth ->	mouse	sunglass	scraper	scissor	mug	cable	notebook	box	ball	utensil pack	opener	pen
notebook	5	3	2	1	5	0	10	3	2	0	0	0
box	0	0	0	0	0	0	0	7	0	0	0	0
ball	2	2	0	0	0	5	0	0	8	0	0	0
utensil pack	0	0	0	1	0	0	0	0	0	2	0	0
opener	0	0	0	0	0	0	0	0	0	0	6	4
pen	0	0	0	0	0	0	0	0	0	0	4	6

- **Task 9: System Demo Video**



Video walkthrough of the Program running. {#video-walkthrough-of-the-program-running }

3. Extensions

- **K-Nearest Neighbor Classification**
  In addition to the simple nearest-neighbor recognition, a K-Nearest Neighbor (KNN) matching was implemented. This involves considering the K nearest neighbors in the feature space to make a decision. Please refer to the classifyObjectKNN function under main.cpp under realTime2dObjectRecognition Project.
- **Contour-based Object Recognition**
  A contour-based object recognition technique was explored. By computing contours for regions in the image, the system can differentiate objects based on their shapes. The computed contours offer a concise representation of object shapes and can be

efficiently compared using techniques like the Hausdorff Distance. Please refer to the computeRegionContour function under feature_extraction.cpp under ImageFeatureExtractor Project.

- **Hausdorff Distance for Contour Comparison**
  Hausdorff Distance, a method to measure the "closeness" between two point sets, was employed to compare contours. By considering the maximum distance between a point in one set and the closest point in the other set, this metric offers a robust way to determine the similarity between two contours. Please refer to the hausdorffDistance function under feature_extraction.cpp under ImageFeatureExtractor Project.

4. Reflection

Throughout this project, the intricacies of object recognition using computer vision became apparent. The initial stages of pre-processing and thresholding were crucial to obtain clean binary images. Morphological operations proved to be vital in refining these images. The importance of feature extraction was underscored when it came to distinguishing between objects. The project also highlighted the significance of a well-curated database for training and the choices of classification algorithms. On a broader note, the project provided insights into the vast potential and challenges of computer vision in real-world applications.

5. Acknowledgements

I would like to acknowledge TA Chenjie Wu for the help in explaining the project expectations during office hour.

For many of the work present in this assignment, I referred to the sources below:

- Computer Vision: Algorithms and Applications, 2nd ed. © 2022 Richard Szeliski, The University of Washington
- Visual Recognition Course from University of Toronto, Winter 2012
- Computer Vision Course from University of Toronto, Winter 2013

Project Running Instructions

Development Environment

Operating System: Windows 10

IDE: Visual Studio 2022

OpenCV: 4.6.0

Execution Instructions

OpenCV Setup for windows

I used OpenCV 4.6.0 Version for this project.

Before run the application, you will need to add the Path for environment variables on Windows.

This step can be referred to this tutorial: Setup OpenCV in Visual Studio 2022 for C/C++ Development

Project Setup within Visual Studio 2022

Step 1: Create a Solution

Open Visual Studio.
Go to File -> New -> Project....

In the Create a new project dialog, choose Empty Project under Installed -> Visual C++.
Name the Solution (e.g., realTime2DObjectRecognition) and the two Projects (imageFeatureExtractor and realTime2dObjectRecognition) and choose a location to save it. Click Create.

Step 2: Add the First Project

In Solution Explorer, right-click on the imageFeatureExtractor project.
Choose Add -> New Item....
I created and stored the following files under this project to extract image features from a local image directory and store feature vectors in a CSV file.

- csv_util.h
- csv_util.cpp
- feature_extractor.h
- feature_extractor.cpp
- thresholding.h
- thresholding.cpp
- morphologicalOperations.h
- morphologicalOperations.cpp
- main_feature.cpp

Step 3: Add the Second Project

In Solution Explorer, right-click on the Solution (not the project).
Choose Add -> New Project....
Again choose Empty Project under Installed -> Visual C++.
Name the second project (e.g., realTime2dObjectRecognition) and click Create.
In Solution Explorer, right-click on the FeatureMatching project.
Choose Add -> New Item....
I created and stored the following files under this project to load feature vectors from a local CSV file and compare them with a live camera frame to get a classified result.

- feature_extractor.h
- feature_extractor.cpp
- display.h
- display.cpp
- thresholding.h
- thresholding.cpp
- morphologicalOperations.h
- morphologicalOperations.cpp
- main_recognition.cpp

Step 4: Set Project Dependencies (if any)

If one project depends on the other, right-click on the Solution in Solution Explorer.
Go to Project Dependencies....
Select a project from the Projects dropdown menu.
Check the projects it depends on.

Step 5: Update the file path for your local environment.

Go to main.cpp under ImageFeatureExtractor Project you created.

- std::string directoryPath is for the image library, where you saved the given image database folder in your local drive.
- saveObjectDBToCSV function call under main function directly contains a string to store the output CSV file.

Go to main.cpp under realTime2dObjectRecognition Project you created.

- loadObjectDBFromCSV function call under main function directly contains a string to locate the input CSV file.

Step 6: Build and Run the Projects

To build a specific project, right-click on the project in Solution Explorer and choose Build.
To run a specific project, right-click on the project in Solution Explorer and choose Set as StartUp Project, then press F5 or click on the Start Debugging button.

Step 7: Debugging and Running

You can debug or run each project separately by setting it as the startup project.

Time Travel Days

I request to use 2 travel days in this assignment.