CS5330: Project 3
Real-time 2-D Object Recognition

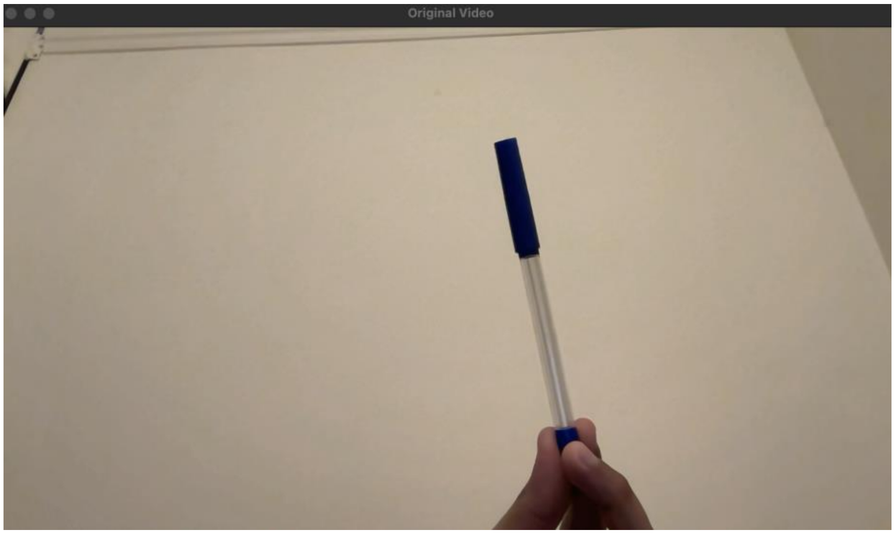
Team Member:
Vishaq Jayakumar (002737793)

Description:

The program includes functionalities for image segmentation, object classification, and database management. Objects are represented by their features and labels, stored in a database that can be saved to and loaded from a binary file. Image processing techniques such as Gaussian blur, thresholding, and morphological operations are applied to the video frames. The program operates in two modes: classification and training. In classification mode, the largest connected region in the frame is identified and classified based on features compared to a pre-existing object database. The classification is based on a scaled Euclidean distance metric, and the system dynamically adjusts a distance threshold for accurate classification. The code provides visual feedback on the identified region, displaying its bounding box and labeling it with classification information. In training mode, the user can label and add new objects to the database. The user can save the object database and exit the program using designated keys. The code includes functions for displaying regions with bounding boxes and labels, as well as computing features and calculating distances for classification. Overall, the program provides a real-time interactive environment for object classification and database management through webcam input.

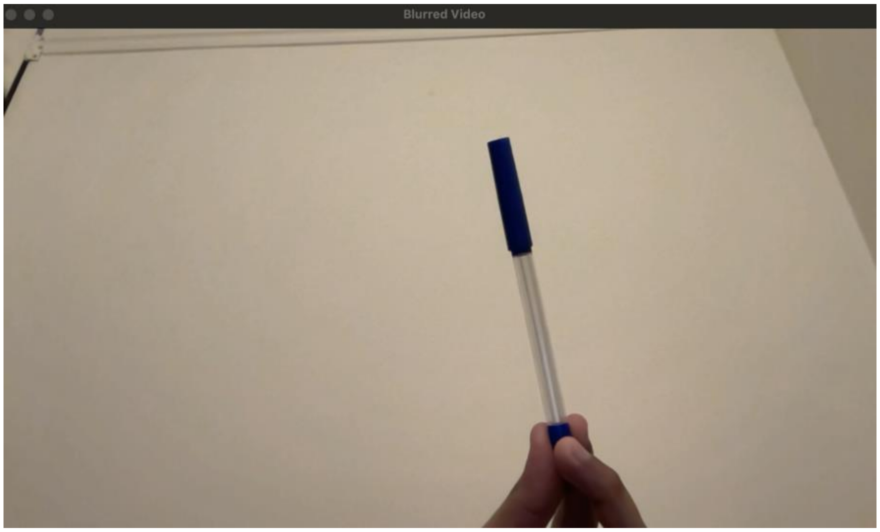
Output Images:

1.



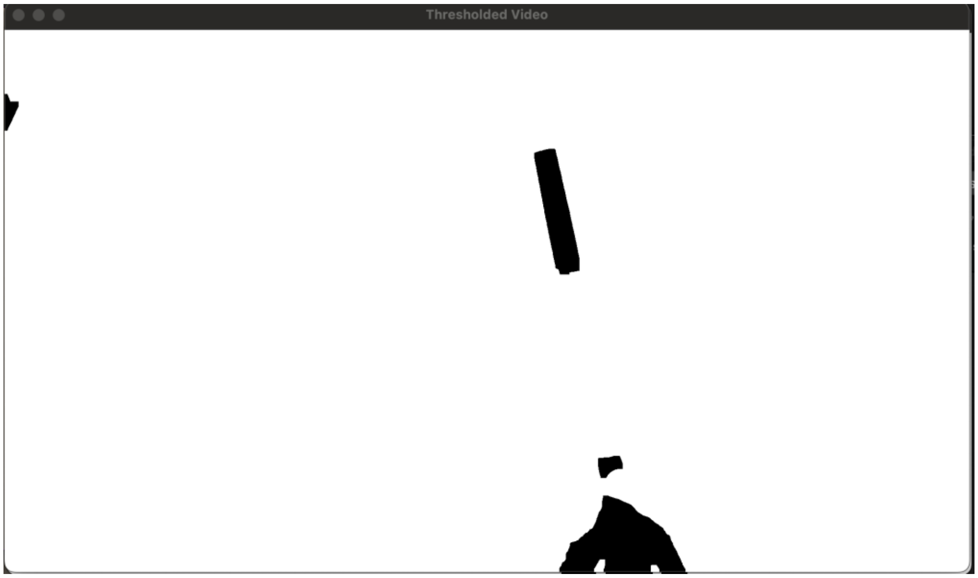
This is a sample frame. Here there is a pen it is the reference object.

2.



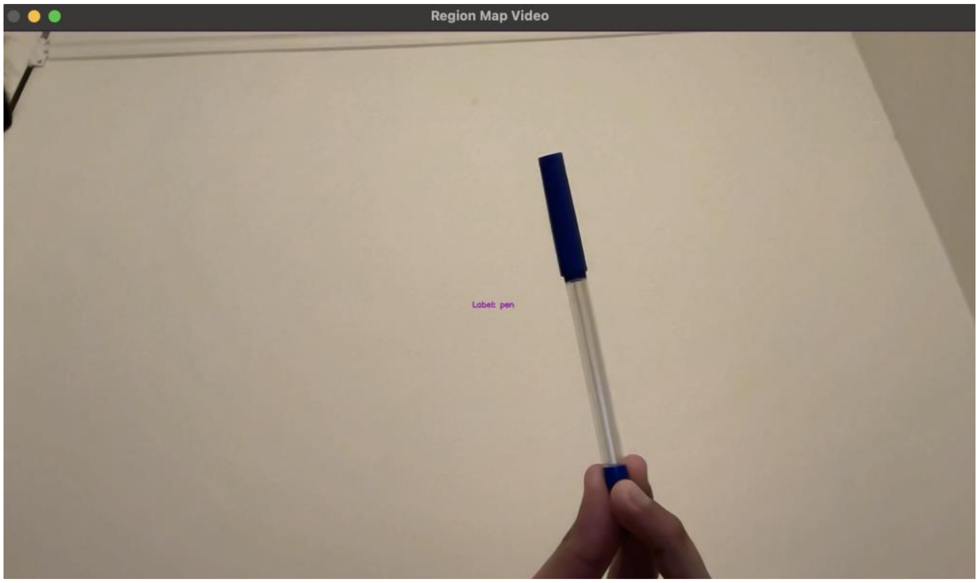
The image is blurred to smoothen and remove any noise from it.

3.



This is the image after thresholding.

4.



Label: pen

This is the output after the classifying is done.

Reflection:

I've applied object-oriented programming principles, using structures and functions to manage object data, handle file I/O operations, and structure the code modularly. The inclusion of database handling functionality reflects my ability to save and load object databases, essential for real-world applications.

User interaction is seamlessly integrated, with keyboard commands for mode switching, saving databases, and program termination. Error handling is considered, demonstrated by informative messages for issues like webcam unavailability or file handling problems. The real-time application design, processing webcam input for dynamic feedback, highlights my understanding of the challenges in developing such computer vision applications.

Overall, this learning experience has equipped me with a comprehensive skill set for practical computer vision applications using C++ and OpenCV.

Acknowledgement:

- Pattern Recognition and Computer Vision course material.
- Stack Overflow community
- https://medium.com/codex/image-processing-using-opencv-and-c-5dfaa7f49be