

Final Project

1 Introduction:

The purpose of this lab is to practice embedded application development. The E9V3 board provides various I/O interfaces, and you are required to utilize some of these interfaces to develop an application. The basic requirement is to receive input, process the data, and generate an output that serves a predefined purpose.

For example, the system may take a video signal, perform processing, and execute operations based on the processed result.

In this project, you will implement a system that performs object detection.

Deadline: Dec. 18th, 2025

2 Requirements:

2.1 Real-Time Object Recognition

- 2.1.1 The system shall capture live video from a camera **connected to the E9V3 board**.
- 2.1.2 The captured video must be displayed on an external monitor (HDMI or LCD is acceptable).
- 2.1.3 Detecting 5 distinct categories of objects, using publicly available object-detection algorithms.
- 2.1.4 During evaluation, **5 object classes** will be randomly selected from a predefined set of **8 classes**, and the system shall attempt to detect them in real time.
- 2.1.5 The 8 possible classes are: **spoon, banana, keyboard, cell phone, book, scissors, bottle, and cup**.

2.2 Photo Object Recognition

- 2.2.1 Detects multiple object categories present in the hidden image using any suitable public-domain object detection algorithm.
- 2.2.2 Two sample photos will be provided to each group. Based on the objects present in these images, you may collect or prepare a dataset to train your detection model.

3 Grading Criteria:

3.1 Real-Time Object Recognition (55%)

3.1.1 Scoring is based on the number of correct object categories detected **within the allowed time**:

Recognized objects	Score
1	25 points
2	35 points
3	43 points
4	50 points
5	55 points

3.1.2 Scoring for recognition speed is defined as follows:

- Recognition completed within **1 second** → **100%** of the time-score
- Recognition completed within **3 seconds** → **90%** of the time-score
- Recognition completed within **5 seconds** → **70%** of the time-score
- Recognition exceeding **5 seconds** → **0%** (no credit)

3.1.3 The recognition speed is measured as the time required for the model to successfully display the correct label on the image after the object or the camera has been moved.

3.1.4 During the demo, the system must complete the object recognition task **within 1 minute**.

3.1.5 The output display must show correct labels simultaneously on the screen, and this condition must be maintained for **at least three consecutive frames**.

3.2 Photo Object Recognition (35%)

3.2.1 Each group's object-count accuracy on the two test photos will be evaluated.

3.2.2 **Groups will be ranked based on accuracy**, and points will be assigned according to ranking.

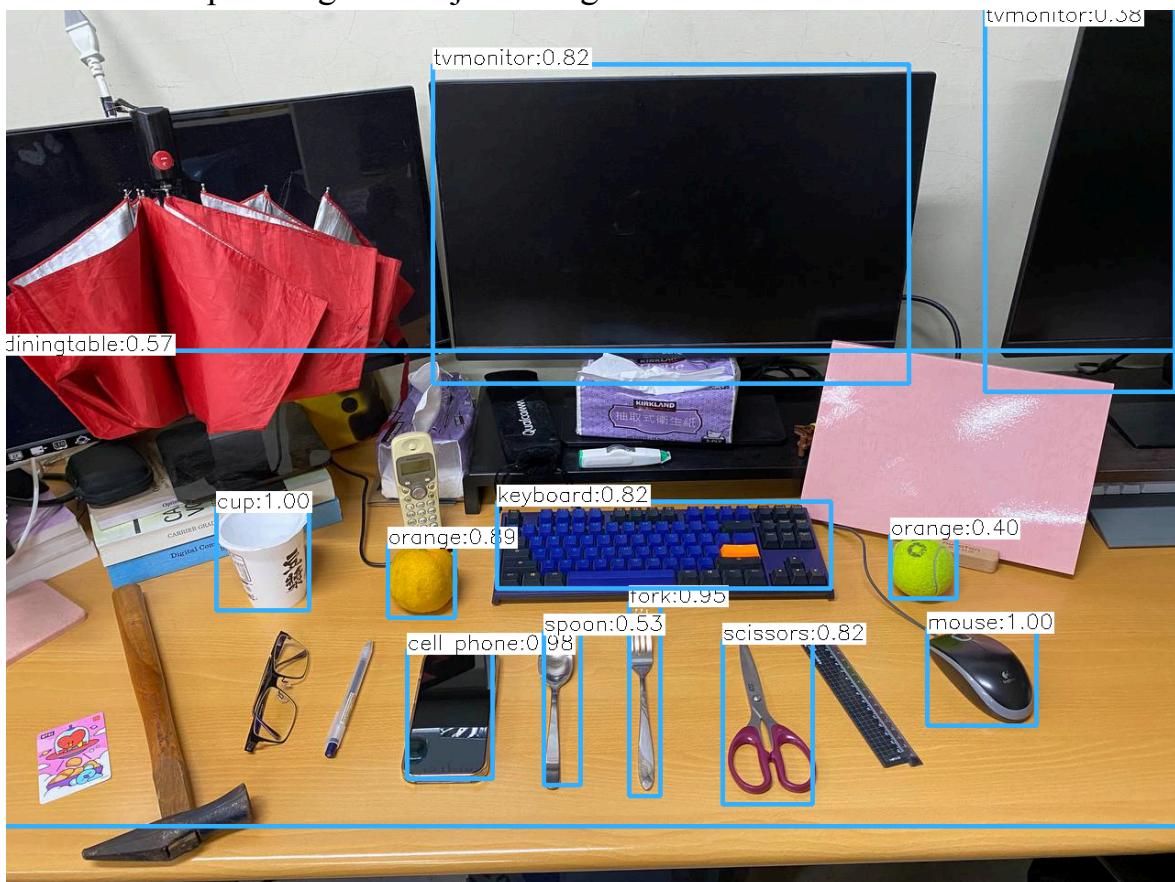
3.2.3 Accuracy is calculated by **summing the total number of correctly detected objects across the two test images**, and all groups will be ranked according to this combined score.

3.2.4 Your system must clearly draw bounding boxes around the objects and label each object with its corresponding category. You may annotate as many objects as possible, as long as the detected objects are correct.

Here is a sample image for object recognition.

3.2.5 The output display must show five correct labels simultaneously on the screen, and this condition must be maintained for at least three consecutive frames.

3.2.6 Sample image for object recognition.



3.2.7 Only objects with correctly drawn bounding boxes and correct labels will be counted for scoring.

3.2.8 Time Limit: Recognition must complete within 5 minutes.

3.2.9 The TAs will provide two label examples. The labels produced by the model on these photos must match the label examples provided by the TAs; otherwise, they will not be counted as correct.

3.2.10 Each label will appear at least once during the demo; however, the specific object or image associated with that label is not guaranteed to be the same.

3.3 Report. (10%)

You also need to hand in a report (either in English or Chinese) that describes your implementation for both 2.1 and 2.2, including how you implement object detection. It does not need many pages but should be concise and easy to understand by readers.

3.4 Upload your source code to E3.

4 Notes:

- 4.1 We will provide an image for you to test your program. It is also recommended to save the resulting image in the SD card.
- 4.2 You need to display the result on the monitor (HDMI or LCD).
- 4.3 You can prepare different programs for each requirement.