Final Project

Objective: At the end of the final project, you will have successfully implemented a YOLO object detection model on a mobile robot that controls via Raspberry Pi.
Goal: You are a member of the special police in the NYC. Successfully navigate the robot through the maze to identify the bad guys.

Training Steps

- Segment the relevant videos and tag them to their respective classes. Construct your Train set, Validation set, and Test set. I prefer this simple program, but there may be better options.
- Train a YOLO by Ultralytics on the tagged images.
- Extract the weights in an appropriate format.

Implementation Steps

- Construct the GoPiGo robot and attach the Raspberry Pi. Connect the camera module to the Raspberry Pi.
- You can burn a base Ubuntu image on the RPi, connect the RPi to the internet, install ROS2 through this, and GoPiGo controls through this. **The alternative way would give you control over the libraries you install, hence it would help you debugging if needed**.
- Install YOLO through Ultrlytics.
- Copy your trained weights to the RPi for inference.
- Write a Python script to read the serial images from the camera. Broadcast them via a ROS2 topic / service.
- Write another Python script to implement the YOLO inference. That script should also subscribe to the ROS2 topic / service to receive the images.
- Improve your Python scripts to recognize AprilTags and calculate the distance between the robot and the Tag.
- A third Python script performs GoPiGo controls depending on the results provided by the inference procedure.

Progress Tracking

- You have the option to work either by yourself or in groups of two. This is the signup link.
- You should have a GitHub repository to host your code. Your progress and contributions are tracked via github. If you decide to keep your repository private, you should grant me (chandima-ccsu) access to your repository.

Grading (subject to adjustments)

Total available points: 50.

- Successfully Train a YOLO model (10pts).
- Implement the YOLO model on the RPi and correctly perform inference on test images (10pts).
- Successfully implement ROS2 Nodes for image communication and message passing on RPi (10pts).

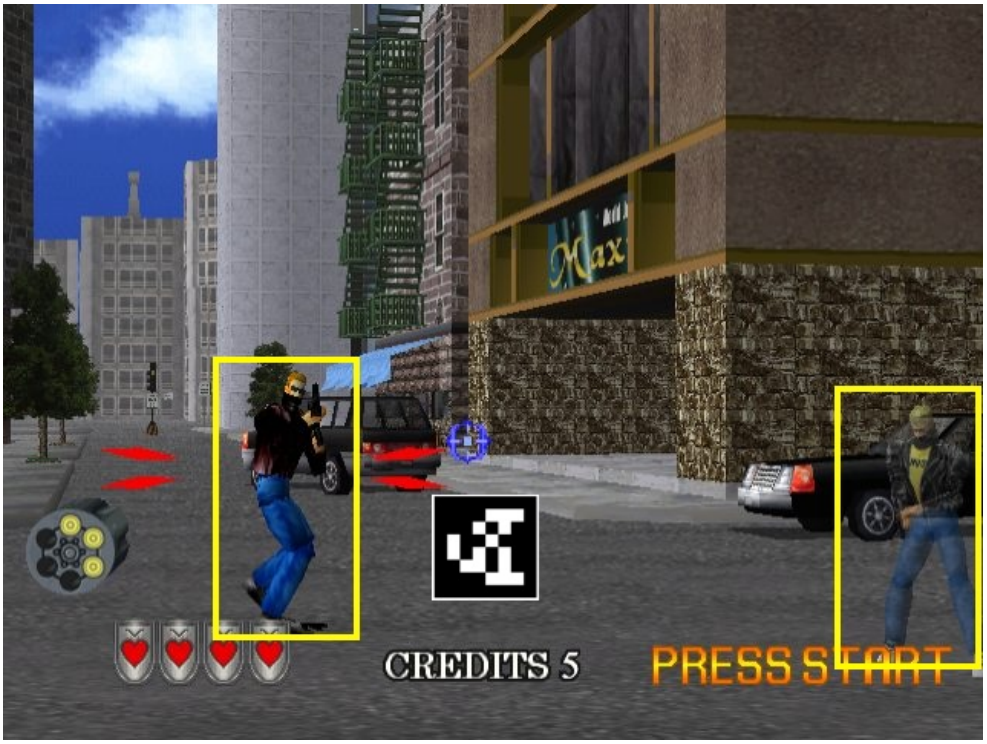

Figure 4: Inference time image with an AprilTag.

- Successfully control the GoPiGo robot through the RPi (5pts).
- Successfully reading the AprilTags (5pts).
- A GitHub repository with a reasonable commit history, a README.md file with results of each of the above steps, and instructions to operate the robot (5pts).
- Working demonstration video (5pts).