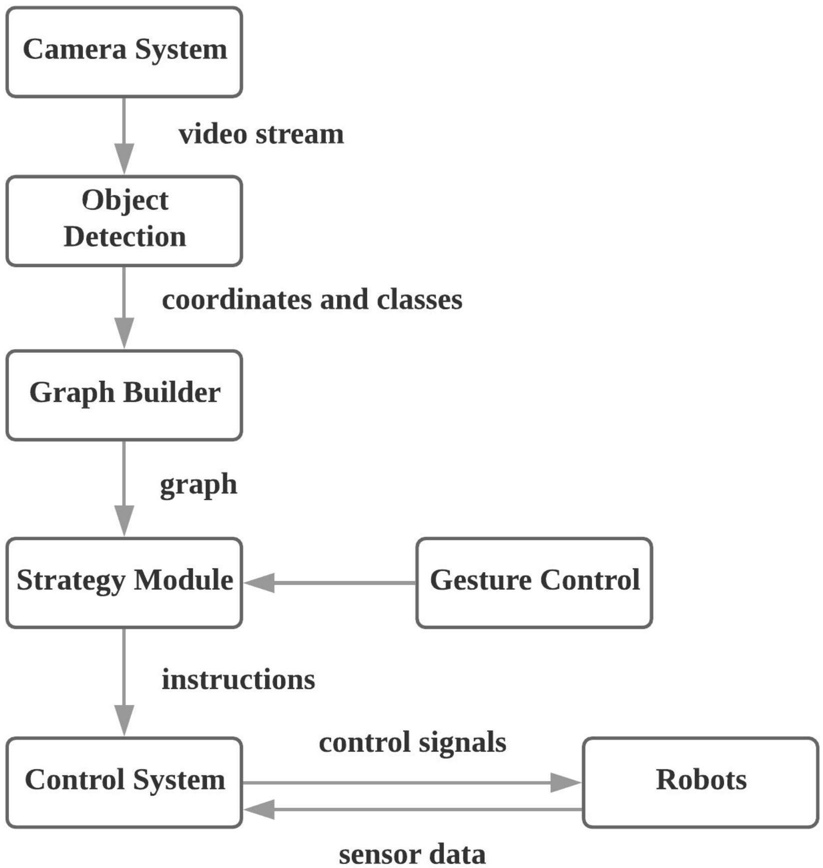
1. Github link

https://github.com/ramk94/Thief_Policemen

2. Project Overview

Design a system for controlling the motion of 2 Vikingbot cars (police) and 1 Hexapod (thief). A camera system which provides a video stream of the gaming board and through the help of our object detection algorithm(Yolo Object Detection), we identify what the object is. The object detector gives us initial coordinates of the center of the gaming board and also the object currently occupying the space. The coordinates and the objects goes into the graph builder which is responsible for giving the graph of the gaming board and also the current center information.Graph is then fed into the strategy module which is responsible for giving instruction to the control system. Gestures are also a part of the strategy module. Based on what gestures you provide, it gives a signal to move the Hexapod. In our game, the Hexapod (thief) moves first so the first signal is received from the Gesture control module. Once the thief moves, car(s) moves accordingly towards the thief.



Note: “Overview: Workflow and Module Design” shows the basic software structure of the project, and please read this file first to have a high-level abstraction. Then, follow the arrows’ directions to read documents in current folder. For example, “Object Detection” is related to Wubin_object_detector.pdf, and so as other modules.

3. Challenges

1. Power was the biggest challenges for this project. We had a 6 volts power supply for the hexapod which was giving power to the hexapod servos. After running for around 10-15 minutes, the battery dies.
2. One of the hexapod servos was not functional and it took us a while to realize that.
3. Magnetometer did not serve the need as required for the project. There was a need for calibration all the time in the initial phase.

4. High Performance computer needed to perform the training and running the actual model which all the members of the team did not have besides Lu and Wubin
5. Car was rotating very fast which is really bad for our algorithm
6. Router connection was very slow.

We had some difficulty initially picking up the project as there was no real explanation we could locate for how things worked or what did what. Worse even was that some parts were missing, like batteries. Leading us to not realize for some time that the hexapod had 2 power sources and not just the one attached to it. The hexapod itself also had a dead servo in the middle connecting servo of leg 4 which was not initially picked up on and had to be replaced. Couple that with the fact that further documented information on the project was only mentioned in the project report from Fall term where we were initially led to believe the project reports from Winter term were all we had to look at.
Power was a major factor on the hexapod and later the viking bots. At first it needed to much until we were provided with a battery that met the power needs just barely but runs out of power in about 20m. We hooked the hexapod up to a power supply to see how much current it actually wanted when it moved and was operating and found that it idled around 2.2 amps just to remain standing and spiked up to 3.9 amps when it was walking. The battery we used provided a reliable about 3 amps. Finding better batteries requires some voltage dividers to fry the Raspberry Pi and Servo Hat that both could only handle 5-6 volts. Since the aim of this class was more on the machine learning and programming side we opted for the simpler approach as some of us weren't as acquainted with the physical aspects of building a robot.
This is why we suggest getting a bigger battery or lightening the hexapod by giving it a lighter but sturdy frame so it can have adequate power for the 18 servos that make up its 6 legs.

Another issue also related to power was the viking bots. The charging cable for the power supply to recharge their batteries had gone missing and the lab tech could not locate it. So unable to charge the batteries for one of the viking bots we opted to cobble together a solution by taping together 8 double A batteries and linking them up as each one was 1.5V so together they made 12V close to the 11.1V battery we couldn't charge provided. This was definitely sufficient for powering the viking bot during the demo but the power was now drawing more current than it was tested with and since the turning for the viking bot functioned as a duration of time. The wheels now having more power caused that viking bot

to overshoot its turns despite the visual software Wubin designed trying to correct the robots movements it just didn't have fine enough motor control to stop the robot from over shooting.
In the future it would be idea to either tune the wheels to more precisely to the power they are getting or allow more finer turns. There is a variety of ways this could be corrected but we were out of time. If you had a more powerful computer than our laptop you could compensate by taking smaller movements more rapidly since the visual software wouldn't have as big of a processing delay between robot moves (it took about 8-10 seconds power round). We thought of piping the data over the internet to the robots to compensate for this but doing to rules imposed by the Computer Action Team (CAT) we can't setup and run a VPN on their network so as long as the devices relied on PSU's wifi for internet it would not be viable without a special exception from the CAT. Otherwise a different way to connect to the internet such as Comcast's Xfinity wifi network if a member was a subscriber might be an option to keep the testing locations more open than just your own home.

While the visual software had some processing delays it was still a late term compensation for the magnetometers that didn't work and took a long time to finally arrive. At first they were being shipped from China but this was around the trade disputes between China and the Trump administration resulting in tariffs that might have impeded things because the order ultimately went undelivered with the selling company claiming it had gone out of stock. So the parts had to be bought again from another seller, this time in America for more money but still took a week to show up. This was expected to be okay as they were expected to work out of the box due to already having code to work with them. But after getting them and testing them it was found they weren't very reliable (we did buy cheap ones) and would be very inconsistent. So that even with initialization software that ran multiple times on the sensor calibrating its data it still came out different at bootup and over time. Making it nearly impossible to use for reliable navigation where robots had to be stopped inside specifically sized triangles on the gaming board. A higher quality sensor that could combine its magnetic field data with a gyroscope and other sensors might have been more effective and it's unclear if the last team really tested this but ultimately the magnatomers seem less than reliable for precision control. Improvements to the visual software Wubin designed or hardware upgrades for the hardware its running on would be recommended. Or incorporating an entirely new sensor of some kind that could handle orientation.

One last issue would be general robot maintenance between terms or by teams. If our robots had been properly maintained and stored using them at the start of the term would have been much easier since we wouldn't have had to diagnose their problems and repair them first. Better documentation concerning the robots final design and how to use it would be helpful to.

4. Future Improvements

1. Maybe replace the hexapod robot with something else maybe a car
2. Replacing the hexapod with maybe a regular car would solve the power issue
3. Changing out the metal frame of the hexapod for something lighter.
4. Better router
5. Faster Computer for training the Yolo Network and processing the data