ECET 2025 Capstone Project
sentryplusgroup@gmail.com
AK, MG

Memorandum

To: Kimberly Lemieux
From: Amaziah Kinavuidi, Mark Gladkevitch
CC: Capstone Faculty, Camosun College
Date: October 16, 2025
Subject: SENTRY Progress Report

Summary

The sentry team is making great progress on their device that is going to detect, track and deter deer. While missing the window for the PCB order, we have managed to get our design in on the next order which was in Week 7. We are currently working on finishing our enclosure design for our hardware by the end of Week 8 and setting up serial communication between the Raspberry Pi 5 and Arduino Uno microcontroller. With having our PCB design now submitted, computer vision code functional, and an enclosure design 70% done, we are sure we will have a working device ready for the capstone symposium on December 13th.

Background

Wildlife intrusions on properties have been an ongoing issue in Victoria BC [3]. Animals like deer damage people's orchards and gardens. Unfortunately, it isn't easy to deter such guests as it takes time and energy to do so. Often it is all out dangerous.

Figure 1, SUBLETHAL home defense turret

People have used several scare tactics, spray repellents and deer resistant plants [1], but these methods aren't always as effective. A company in South Africa is implementing non-lethal pole mounted turret systems called the SUBLETHAL system for property defense [2]. While the idea of sentry turrets is not new, very few have managed to implement and innovate this idea for affordability and accessibility to the public.

Our Product

We plan to create a sentry turret that will be accessible to everyone. Our turret will consist of a small computer that's programmed to detect, track and deter deer. It will aim to help homeowners or renters, and farmers have a system that autonomously watches for deer using a camera and deters them when detected. This will save a lot of time and energy for a lot of people who deal with deer intrusions repeatedly.

Progress

We've made excellent progress with the hardware for our sentry device and have working software. We have our printed circuit board design submitted for fabrication, enclosure under design process and a working program that is detecting and tracking deer.

Hardware

The hardware, which is the physical component of our project, will mainly consist of the Raspberry Pi 5 for computer vision, a PCB for power distribution, stepper motors that will be moving our laser to deter the deer and our enclosure which will hold every component together.

As of now, the hardware component is slightly behind schedule due to a missed PCB order window in Week 5, and a longer than expected development time for the turret 3D model. All necessary parts are bought, including all components for the PCB. In addition, we received a donation; an upgraded camera that works from a USB port, which we hope will increase the performance of the software and reduce latency. We have already completed and submitted our PCB design for manufacturing and are currently working on 3D design for our enclosure in Fusion 360. Once this is done, we hope to begin to assemble the turret.

Printed Circuit Board (PCB) features:

Our board will mainly function as a power distribution system for different hardware components.

- All power cords from devices merged into one wall outlet cord for ease of use.
- Screw terminals for free wire connection, this way we can power our devices at any distance from the PCB
- Special mount for Arduino microcontroller
- Extra male attachment pins to power new optional devices
- Extra USB-A ports to power more optional equipment such as speakers or lights.

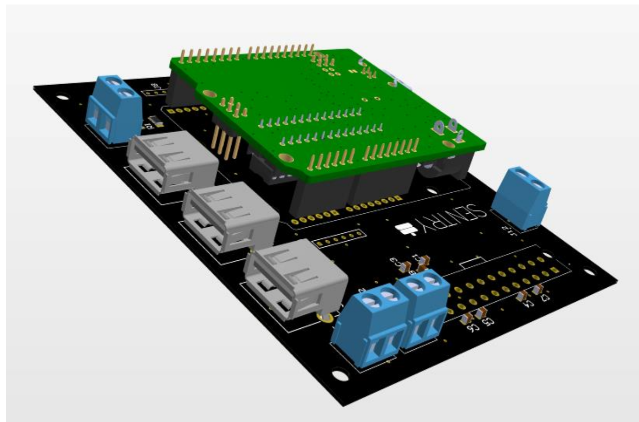


Figure 2, 3D PCB model (not fully assembled)

Laser:

Our 5mW green laser has arrived in good condition, and we have confirmed its operation. It shines much brighter than expected which will be good for higher visibility.

We chose a green laser because they are known to be more visible during high-light conditions which will increase the chance of the target noticing it when it is being tracked.

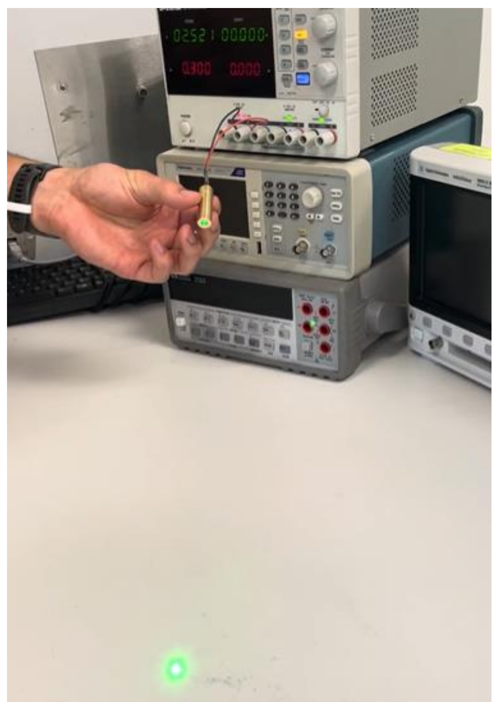


Figure 3, Laser operational

Motors + Motor Controller:

We have verified that the NEMA 17 motors are operational by conducting a brief test with an Arduino that is hooked up to a HW-095 motor controller. There were some worries that the motor controller would not be compatible with the stepper motors because they are made for a different type of motor, but the motors run flawlessly. Also, the controllers are available in the back room which makes it easier to source.

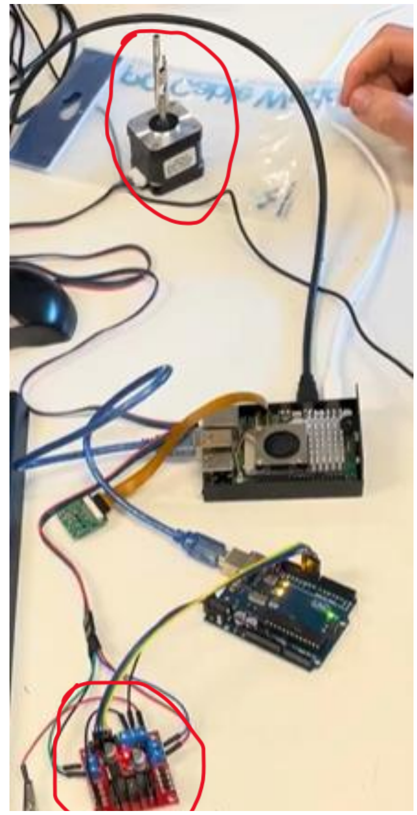


Figure 4, Motor connected to Motor Controller and Arduino

Camera:

The camera has a field of view of 100 degrees and a resolution of 1280x720 pixels. The field of view is the most important part because the wider you can see, the further you can track. It is easy to operate, and it plugs into the USB port of our Raspberry Pi which adds versatility to future camera upgrades.

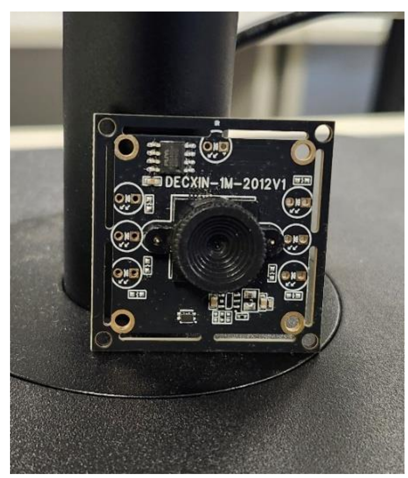


Figure 5, Upgraded camera

Arduino:

We have decided to use this microcontroller to operate our motors and deterrence devices due to its ease of use and availability. It will be mounted onto our circuit board as shown in figure 2.

Figure 6, Arduino uno board

Raspberry Pi:

This will be our main brain due to its high processing power. It will handle all software tracking processes and send signals to the Arduino controller.

Figure 7, Raspberry Pi 5i

Power Supply:

We found that a computer power supply was our best option due to its power capabilities and because there are plenty of old supplies lying around for free. This PSU supports various voltages and currents which are perfect for any future expansion.

Figure 8, Basic computer power supply

Challenges

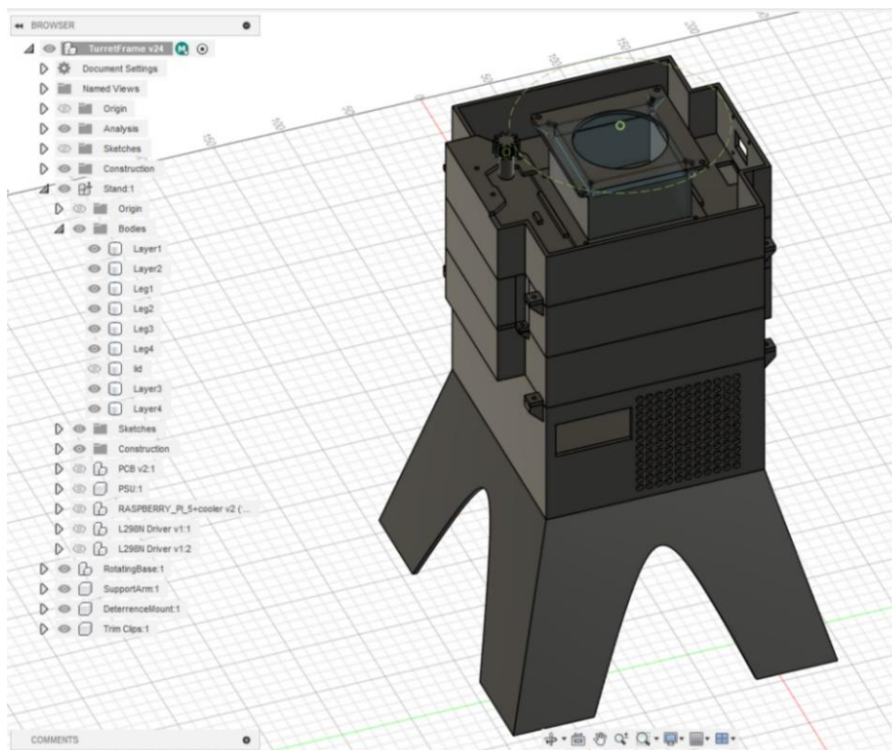
As we progress through the project, new challenges and complexities arise. But with the help of our peers and instructors there has been no challenge that we could not overcome.

Altium PCB Design:

Due to design flaws, there were many delays and reconstructions that had to be made and verified by the designated instructor. Furthermore, due to Mark's sickness he missed the week five deadline for PCB submission. Thankfully it turned out to be a good thing after noticing some critical mistakes in some of the pinout designations.

Fusion 3D model:

Due to lack of experience in 3D card modeling the design process has taken longer than projected. The design had to be re-thought and re-designed many times, but we are steadily moving along.



Data Collected

We have received lots of valuable feedback and data. This ranged from general knowledge about PCB layouts to helpful ideas for strengthening our turret frame and optimized parts selection. We hope to use this knowledge to increase the 3D design efficiency and speed for future turret components.

Goals

The goal is to complete and print the 3D model of our turret's enclosure before the end of week 8 while we are expecting to receive our PCB around week 9. We plan to have a working prototype by week 10.

Software

We have a computer program that is going to be running on our hardware and sending instructions to detect and track deer, then move the motors to the corresponding location.

Different datasets containing annotated and labelled deer images are collected from Roboflow, a computer vision tool for developers. The images are split into a train, valid, and test set. Having 3 different sets, as we see below, ensures the models are properly trained on the train images only, then validated on the valid images and then tested on the test images. This will help us have a complete and tested model.

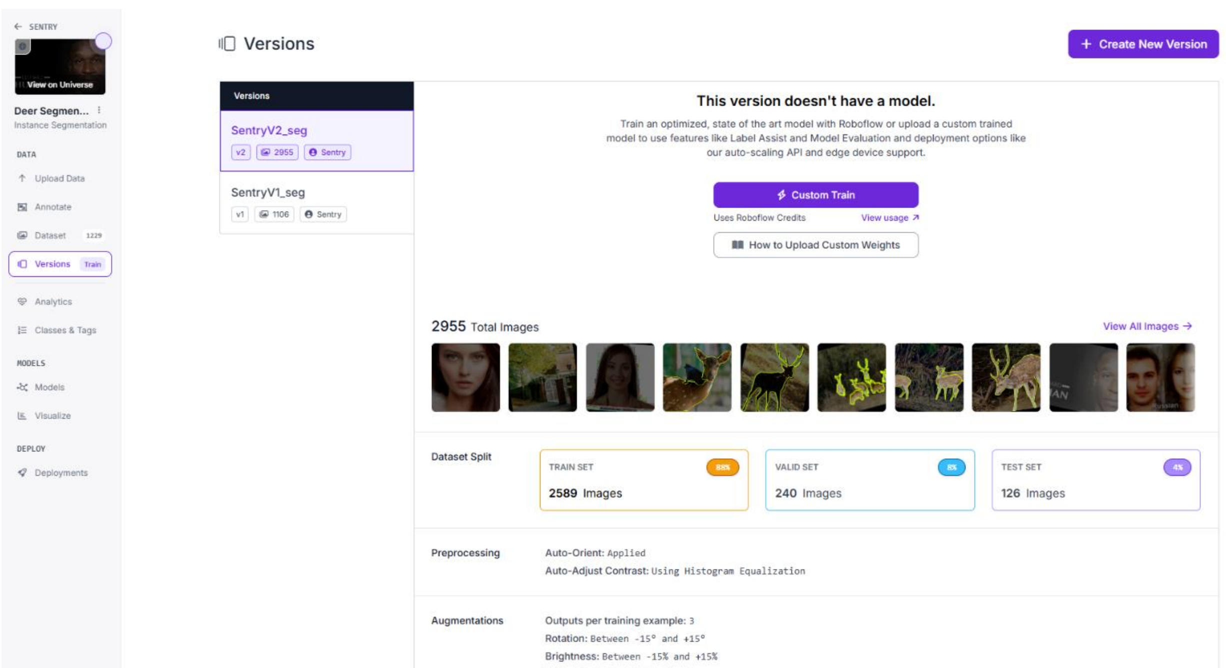


Figure 10, Dataset training website

The set of images collected will determine how accurately we'll detect and track the deer. We have a few different custom computer vision models (YOLO) on which we've used pre-trained models to train them specifically on deer images in various environments. The model is then used in the code for detecting deer (See **Appendix D - *Figure.16*** for tracking code).

So far, we have organized images, trained YOLO (computer vision) models for pinpointing our targets and programmed the Raspberry Pi 5 computer to use a model for detection and tracking of the deer.

Challenges

One of the challenges faced was finding a good dataset with good annotations and labels. We ended up personally annotating each image. Another problem encountered was that at times the model would have false positives, meaning it would detect a human or another and regard them as deer. However, we did more research on different methods of improving accuracy. We used image segmentation which annotates the exact pixels of a deer, which makes it easier for a computer vision model to detect the deer.

Data Collected

We've collected data on the several custom trained models (models trained on different sets of deer images only), such as the confidence levels and precision the models have on various images. We can see an example in fig 11. The data collected tells us how the model works on detecting deer in various environments. We analyzed the data and checked if any improvements need to be made.

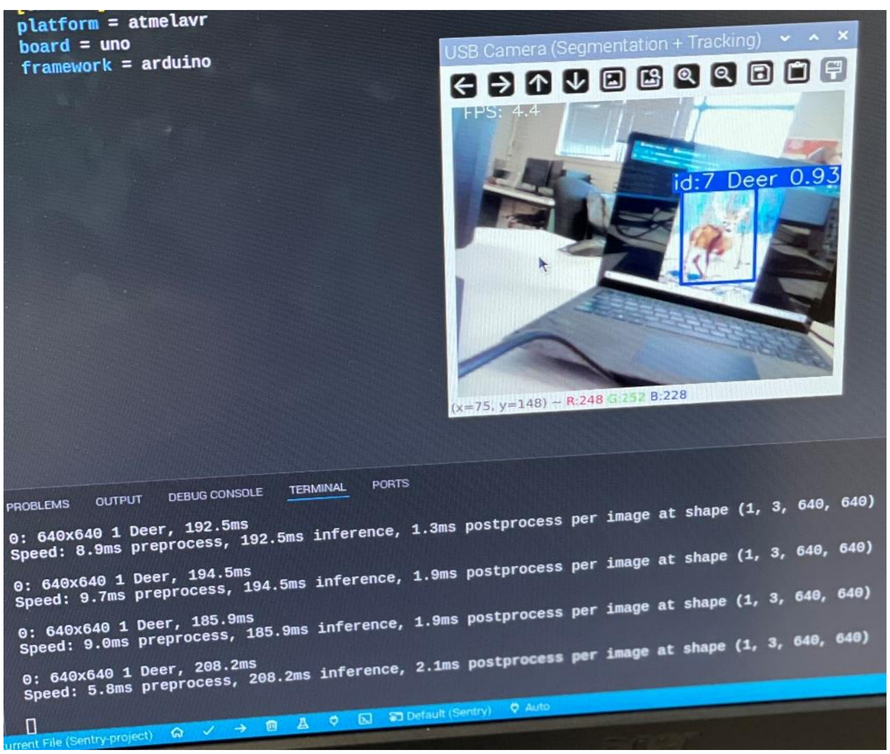


Figure 11, Camera feed with live deer detection

Goals

A goal we have is to have a working dataset with a deer, person and non-deer class by the end of Week 7. The model should have a high confidence level and be able to tell between classes accurately. On Week 8, we plan to have an optimized computer vision program.

Firmware

The firmware is the software built into the hardware which allows the software and the hardware to communicate. We have good communication with the software and the camera; this was achieved through a downloadable camera library and a few lines of code in our main software script. Although communication between software and motors is still being established, we are still optimizing the stability of the detection software and setting up serial communication between the raspberry pi5 and Arduino uno. The live stream video has been put on pause as we are working on the defining features of the project for now.

Challenges

The challenge was getting a smooth-running video for tracking, because the camera was unoptimized for such work and the file formals for the deer tracking models were larger. The software ran at about two frames per second, which is not even close enough for a smooth target tracking experience as the turret will make drastic movements every half a second. But we have resolved the issue by using optimized file formats for software and upgraded our camera.

Data Collected

We have learned about the correct selection use of trained model file formats and discovered the best resolution to run our program in for maximal software speed and target recognition accuracy.

Goals

We plan to polish off the camera to software part of the firmware along with the software since they are very closely tied together. For the motor to software communication the projected completion time will be near the end of week 8.

Finance

We are well within our financial budget of $500 and are on track to stay that way. We have spent around $20 on PCB parts since the start of project, and it is expected to be the last major purchase for our project. SpectrAI donated a camera for our sentry device, which has saved us money. The donation allows us to have extra funds for unforeseen expenses.

Our Status

Overall, we're not too far behind. With our PCB complete and submitted, enclosure being designed, motors and laser operational, and some software detecting/tracking, we are still able to have our device done and working before capstone symposium.

References

[1] District of Oak Bay, "Urban Deer - District of Oak Bay," District of Oak Bay, Jul. 04, 2025. http://oakbay.ca/community-culture/pets-wildlife/urban-deer/

[2] "Sublethal remote guns." Available: https://www.sublethal.co.za/what-is-a-remote-gun.php

[3] CBC, "Urban deer in Greater Victoria community to be put on birth control," CBC, Aug. 24, 2021. Available: https://www.cbc.ca/news/canada/british-columbia/urban-deer-birth-control-1.6147200

[4] Product Design Online, "Create custom 3D printable gears in Fusion 360 | Practical Prints #1," YouTube. Jan. 11, 2020. Available: https://www.youtube.com/watch?v=B8A_11o7QZ0

[5] Tim, "Object and Animal Recognition with Raspberry Pi and OpenCV," Core Electronics, Sep. 30, 2024. https://core-electronics.com.au/guides/object-identify-raspberry-pi/ (accessed Oct. 17, 2025).

Appendix A – Finance

Actual Expenses																	
Labour		$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$0/h	
Capital Expenses																	
Facilities/Equipment		$6,405.00			$ 7.00												
Supplies				$ 60.25		$ 19.42											
Weekly		$6,405.00	$ -	$ 60.25	$ 7.00	$ 19.42	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -		
Cumulative		$6,405.00	$ 6,405.00	$6,465.25	$6,472.25	$6,491.67	$6,491.67	$6,491.67	$6,491.67	$6,491.67	$6,491.67	$6,491.67	$6,491.67	$6,491.67	$6,491.67		$6,491.67

Figure 12, Total current expenses sheet (Tuition included)

Qty	Unit	Category	Description	Cost	Extended		Note
1	pieces	Hardware	Raspberry pi 5	$ 219.00	$ 219.00		Ordered from Amazon
1	pieces	Hardware	Arduino uno	$ 30.00	$ 30.00		General Camosun inventory
1	pieces	Hardware	Green Lazer	$ 15.89	$ 15.89		Ordered from Amazon
1	pieces	Hardware	PCB	$ 20.00	$ 20.00		General Camosun inventory
2	pieces	Hardware	2X Nema17 stepper motor	$ 17.99	$ 17.99		Ordered from Amazon
1	pieces	Hardware	10X Radial bearing (Y-axis)	$ 8.99	$ 8.99		Ordered from Amazon
2	pieces	Hardware	2xLazy susan bearing (X-axis)	$ 9.48	$ 9.48		Ordered from Amazon
7	pieces	Hardware	Capacitor 10µF +/-20% 25V 0805	$ 0.45	$ 3.15		General Camosun inventory
3	pieces	Hardware	USB Connectors USB 2.0 Interface, Type A	$ 2.21	$ 6.63		From digi key
1	pieces	Hardware	M1ScrewTerminal	$ 1.41	$ 1.41		From digi key
1	pieces	Hardware	M2ScrewTerminal	$ 2.48	$ 2.48		From digi key
1	pieces	Hardware	LazerScrewTerminal	$ 1.41	$ 1.41		From digi key
1	pieces	Hardware	MOSFET (N-Channel)	$ 0.67	$ 0.67		General Camosun inventory
1	pieces	Hardware	Resistor 10k +/-1% 1206 250 mW	$ 0.19	$ 0.19		From digi key

Figure 13, Current bill of materials

sentryplusgroup@gmail.com
AK, MG

Appendix B – Risk/Requirements

Table 1, SENTRY risk table

Description	Probability	Impact	Risk	Response	Outcome
Power Outage	High	Very High	High-Very High	Accept - Device won't receive power	Device stops operating
Bearings oxidization	Medium	Medium	Medium	Mitigate - Make it more weatherproof	Bearings last longer
False positive	Low	High	Medium	Mitigate - Train the software better	Software detects deer more confidently
Shortings	Very low	High	Low-Medium	Avoid - Testing and spacing out wires before assembly	Probability of shortage becomes even lower
Prolonged exposure	Very low	Medium	Low	Mitigate - Improve software accuracy	Decreases the chance of eye damage
Camera obstruction	Medium	Medium	Medium	Transfer - Product owner must get rid of obstruction	Impact becomes zero
Wi-Fi connection limits	Medium	Low	Medium-Low	Transfer - Owner must place product closer to the modem	Impact becomes very low

Table 2, SENTRY requirements table

Ref	Requirements	Test	Pass/Fail Criteria
R01	System can run autonomously without external technology or supervision	We will let the system become standalone with no laptop connected or person controlling it and we will observe if the system operates as expected.	If the system stays ON and operates according to group expectations then it's a pass, anything else is a fail.
R02	Camera can recognize the set targets	Check the video feed from the camera to see if it is recording and observe if the software will recognize its target.	The camera must be able to tell between targets and non-targets.
R03	Camera can track its target	Observe if the software will track the target while its stationary or when it moves at different speeds.	The software must follow its target with precision, and with low latency.
R04	The motors can move the laser to track its target in accordance with the camera	Power on the system and test if the motors move the laser pointer in accordance with software.	If the motor guided laser is aligned with the software's targeting, then it's a pass.
R05	The system should have a kill switch that works	Test the kill switch if it works	We must be able to turn the system ON and OFF

R06	It must be safe for consumer use	Test the product ourselves to see if it's safe	If the system isn't lethal and doesn't harm anyone, then it should be a pass
R07	The system must have an automatic and manual deterrence mode	Turn on the system and see if it automatically sets the deterrence mode and test if it can be switched to manual mode	The system must fend for itself without external influence in auto mode, and if can be fully controlled by an operator in manual mode.
R08	The aiming system is accurate	Turn on the system and observe where the software is aiming in comparison to the physical laser	If the laser pointer is aligned with the software's targeting and it is guided with low latency, low jitter, and high precision of about 10 MOA then it's a pass
R09	The final cost should be less than $ 150 CAD	Go back and check the total cost of the parts	If the total cost of the parts isn't more than $150, then it's a pass
R10	Must be able to stream the camera video feed to another device.	Check on our cellphones if we're receiving live video feed	Live video feed with decent resolution and low latency

Appendix C – Printed Circuit Board

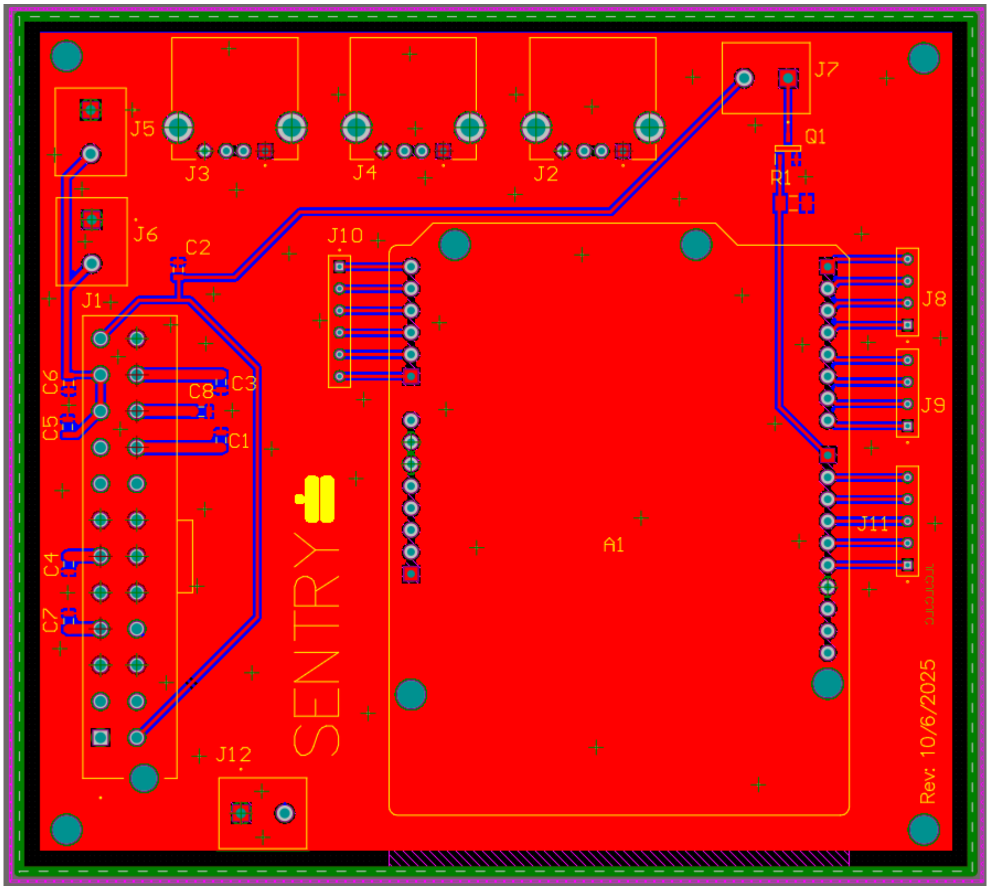


Figure 14, PCB 2d board diagram

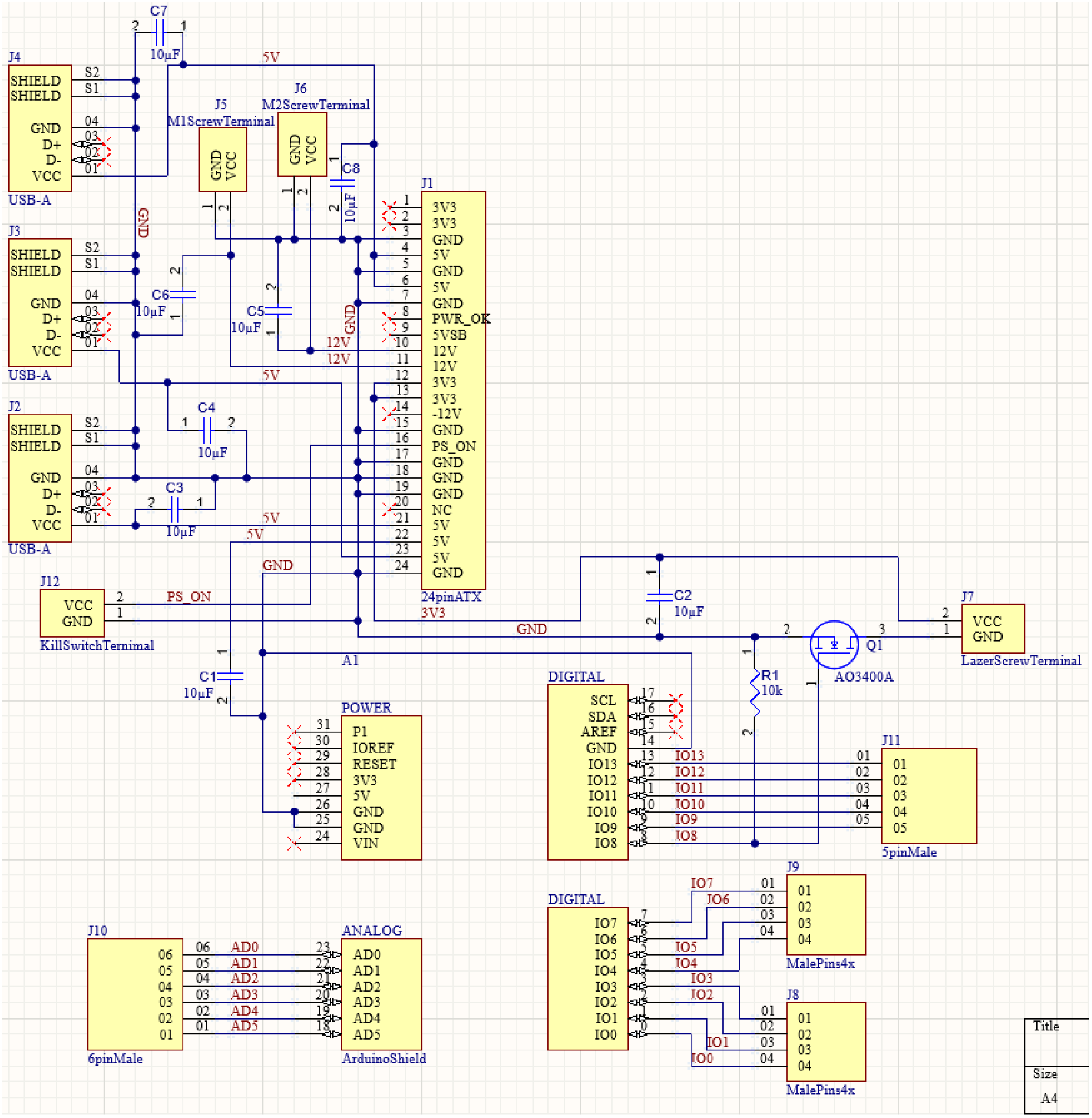


Figure 15, PCB schematic diagram

Appendix D – Software

```python
import cv2
from ultralytics import YOLO
import time

# --- Initialize USB camera ---
cap = cv2.VideoCapture(0)  # 0 = default USB camera
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 320)   # Can be adjusted for performance
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 240)

# --- Load YOLOv8 model ---
model = YOLO("deerdetect.pt")

while True:
    start_time = time.time()

    # Capture frame-by-frame
    ret, frame = cap.read()
    if not ret:
        print("❌ No frame captured.")
        break

    # Run YOLO detection
    results = model.track(frame, persist=True, tracker='botsort.yaml', conf=0.60, iou=0.05)
    annotated_frame = results[0].plot()

    # Calculate FPS
    end_time = time.time()
    fps = 1 / (end_time - start_time)
    text = f'FPS: {fps:.1f}'

    # Draw FPS text
    font = cv2.FONT_HERSHEY_SIMPLEX
    text_size = cv2.getTextSize(text, font, 1, 2)[0]
    text_x = annotated_frame.shape[1] - text_size[0] - 10
    text_y = text_size[1] + 10
    cv2.putText(annotated_frame, text, (text_x, text_y), font, 1, (255, 255, 255), 2, cv2.LINE_AA)

    # Display the resulting frame
    cv2.imshow("USB Camera", annotated_frame)
```

Figure 16, initial code for detecting and tracking deer

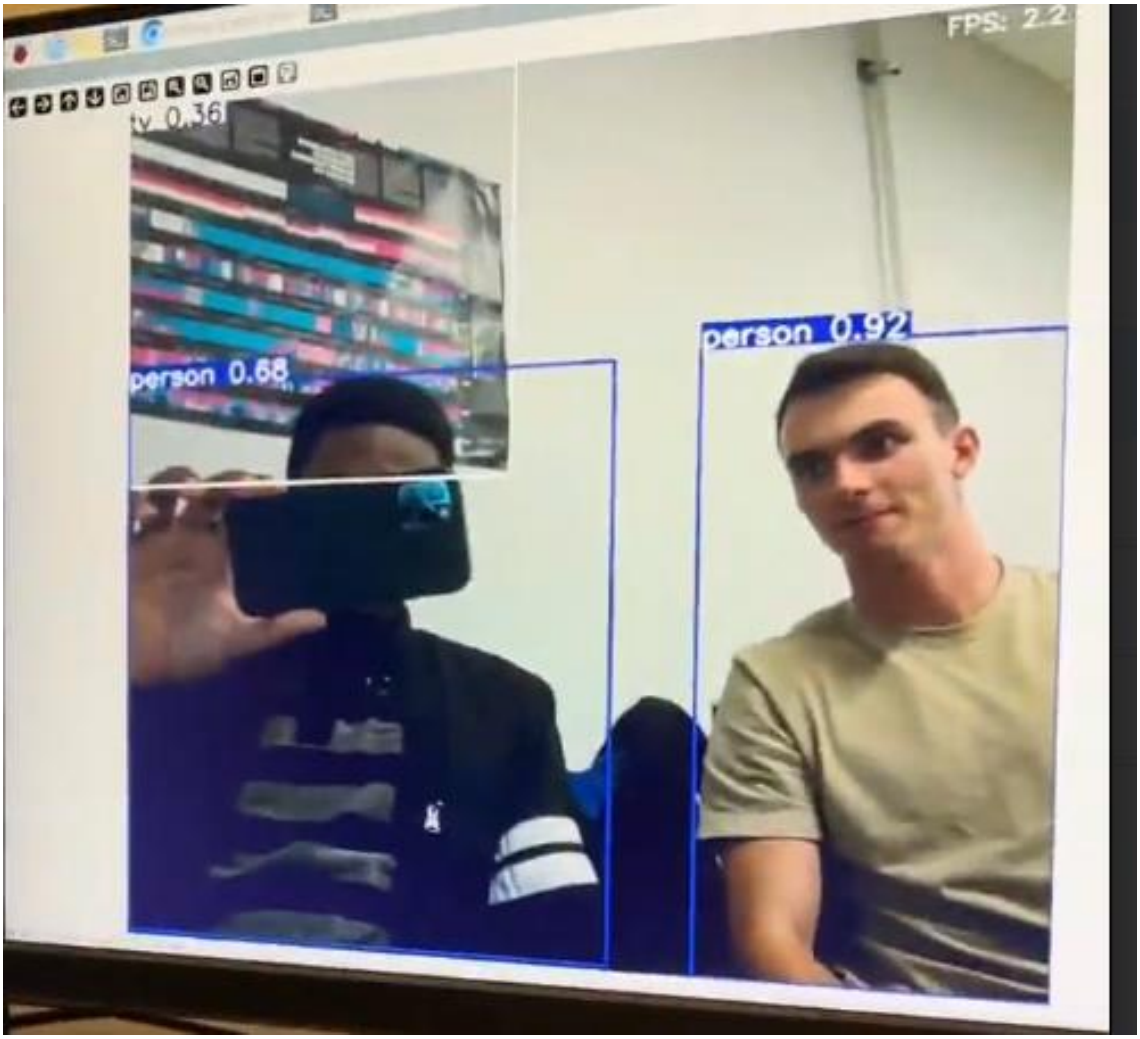


Figure 17, Target tracking video before optimization

Appendix E – Fusion Model

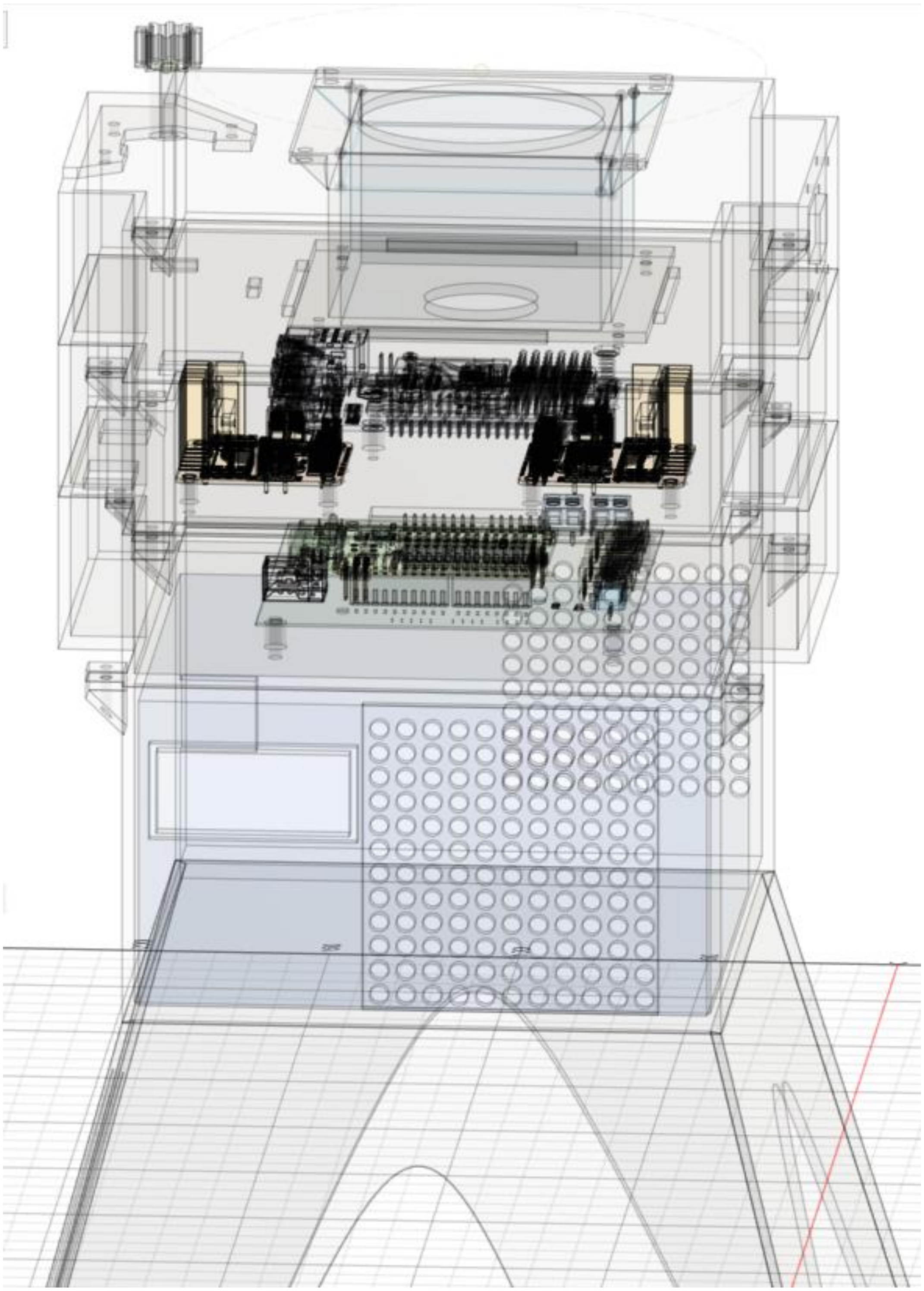


Figure 18, Current 3d model internal component view