



# Thermal Threat Detector

## Critical Design Review



Photo Science Capstone 2023-24  
April 2nd, 2024



# Overview



- Design and develop a **thermal imaging system** that can **detect a suspected threat in a room** with no lighting.
- The device must be able to enter a room and return images to the operator.

# Requirements

**Size:** The device should be around **14 in (d) x 4 in (h)**.

**Sound:** The device must be able to operate with **minimal sound output**.

**Mobility:** The system must be on the **ground**.

**Operation:** The device must be operative in **manual mode**, but if possible, it should be able to switch between manual and autonomous modes.

**Power:** The device should last **at least 1 hour** on a single charge.

**Camera:** The imaging system should have multiple thermal cameras able to detect **a range of human body temperatures**.

# Requirements

**Field of View:** The imaging system should achieve **180 degrees**.

**Users View:** **Wireless live view**, from a safe distance.

**GUI:** The GUI must have a button to allow the user to **capture a picture**.

**Resolution:** The system should be able to resolve **a human being**.

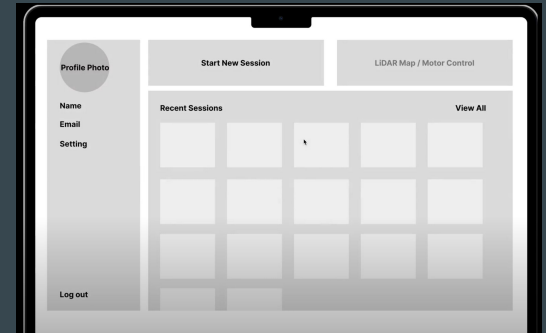
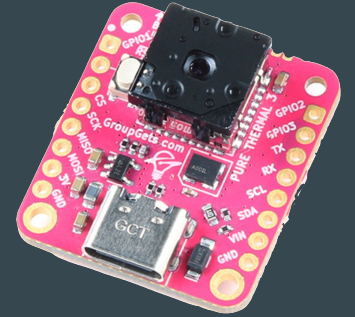
**Lens:** The lens should be **a fixed focus lens**.

**Budget:** The budget is **\$3,000**.

**Completion date:** The completion date is **April 27, 2024** at Imagine RIT.

# Where we left off

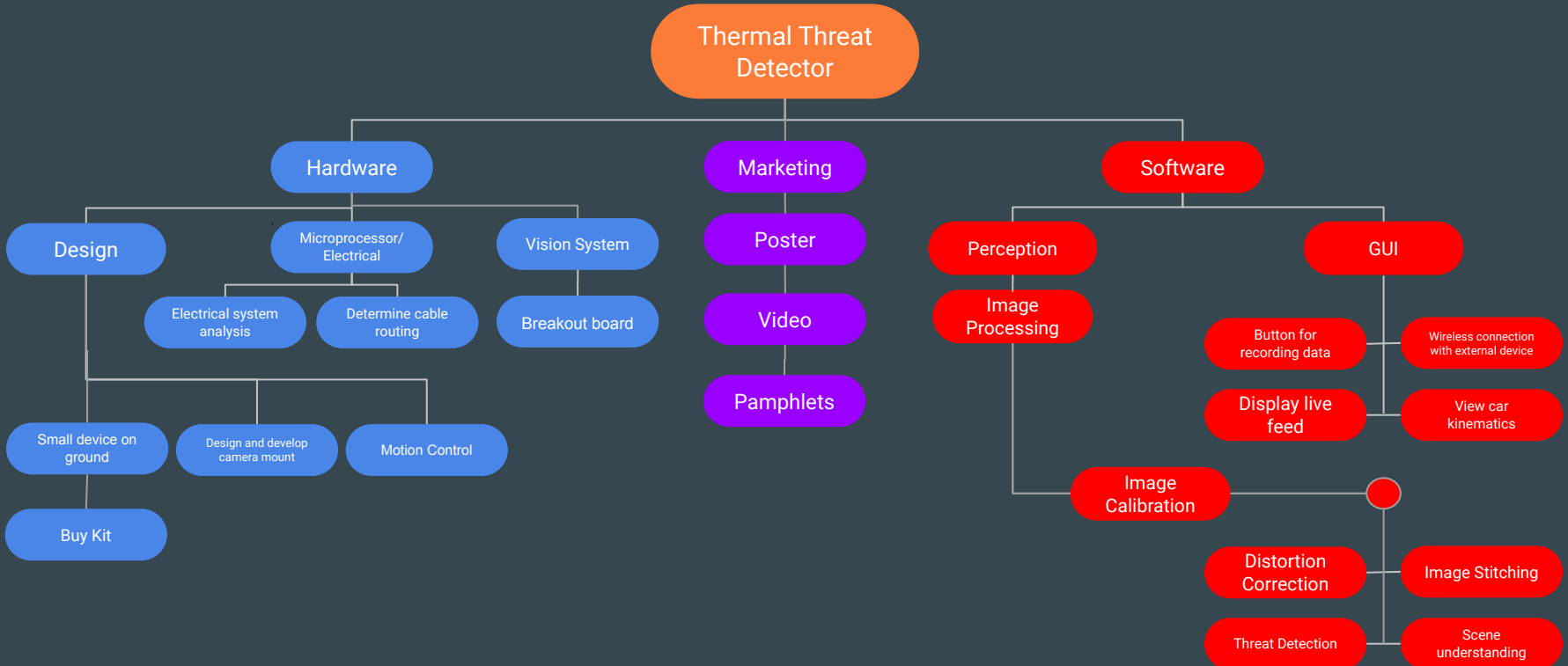
- Connecting to the Camera
- Programming GUI and Object Detection
- Designing camera mount and printing prototypes



## What we are working on

- Fine tuning camera mount
- Camera Calibration
- Image Stitching
- Vehicle control
- Finalizing GUI

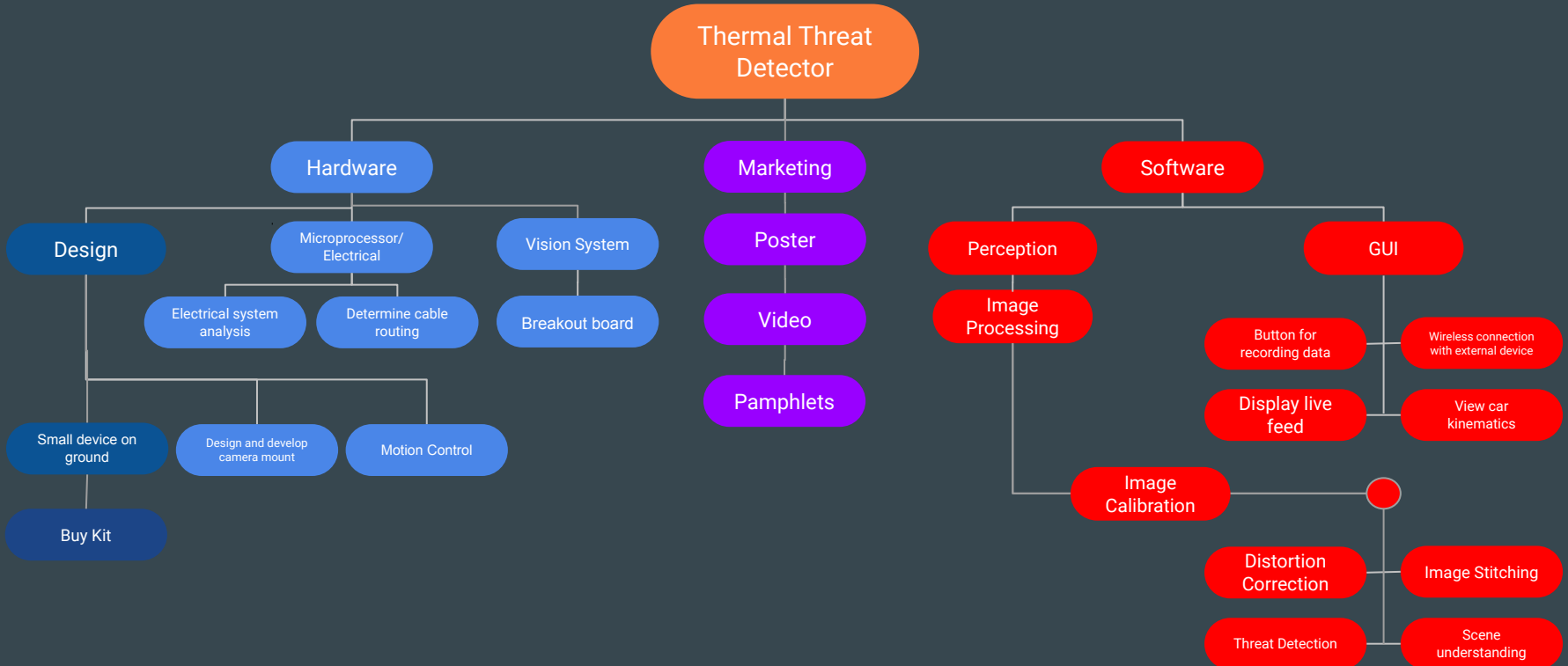
# Work Breakdown Structure



# Hardware Updates



# Work Breakdown Structure: Vehicle



# Vehicle

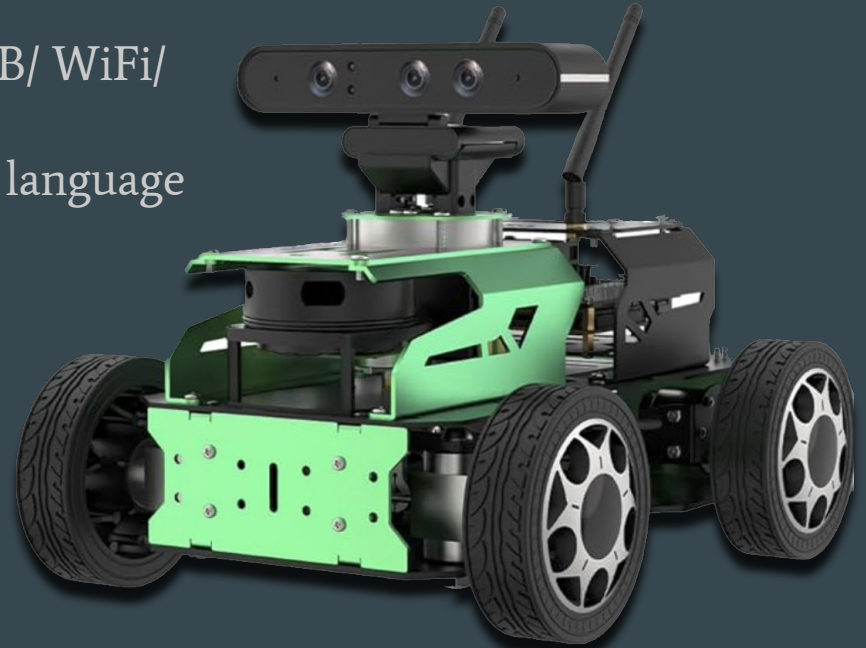
The vehicle chosen was the JetAcker with Jetson Nano.

Included features:

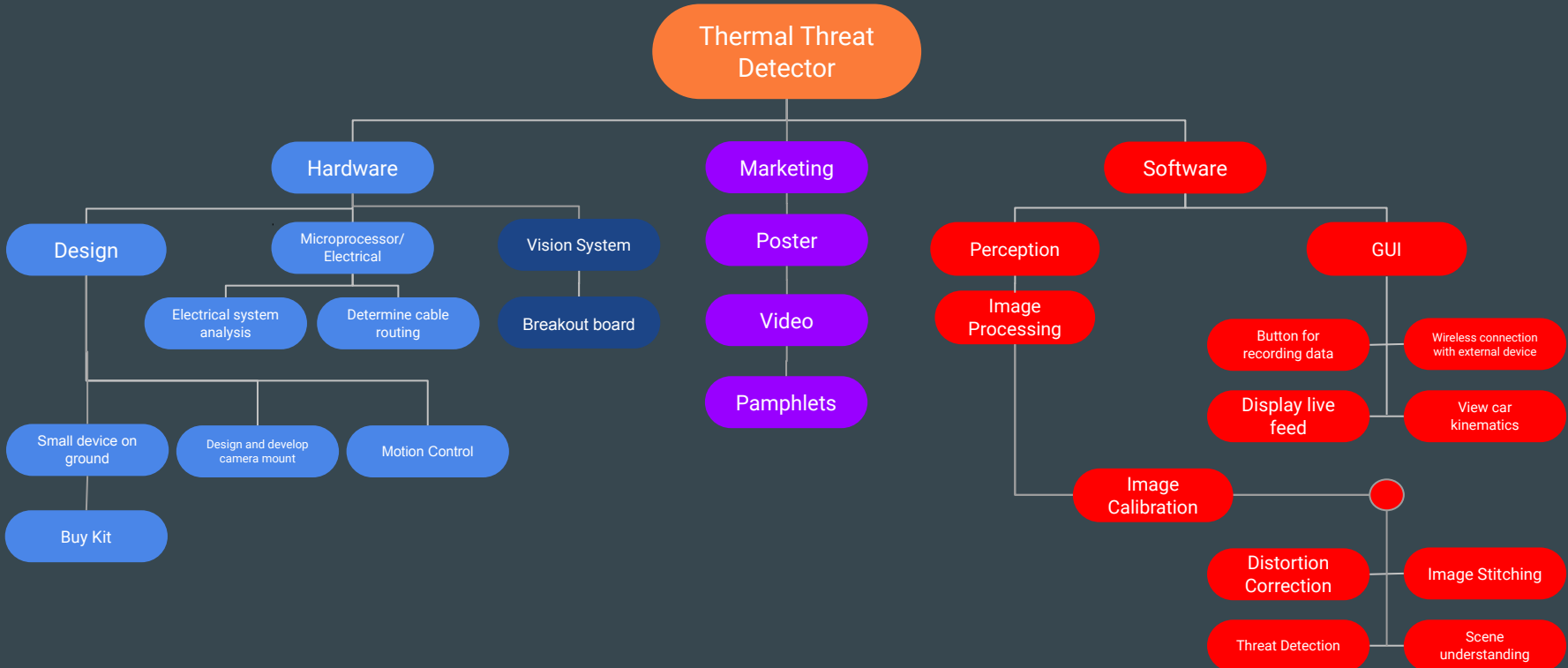
- LIDAR Mapping
- Multiple communication Methods (USB/ WiFi/ Ethernet)
- Compatible with python programming language

A series of test have been run:

- Sound output
- Manual control
- Trained Lidar
- Battery Testing



# Work Breakdown Structure: Vision System



# Pugh Analysis - Camera

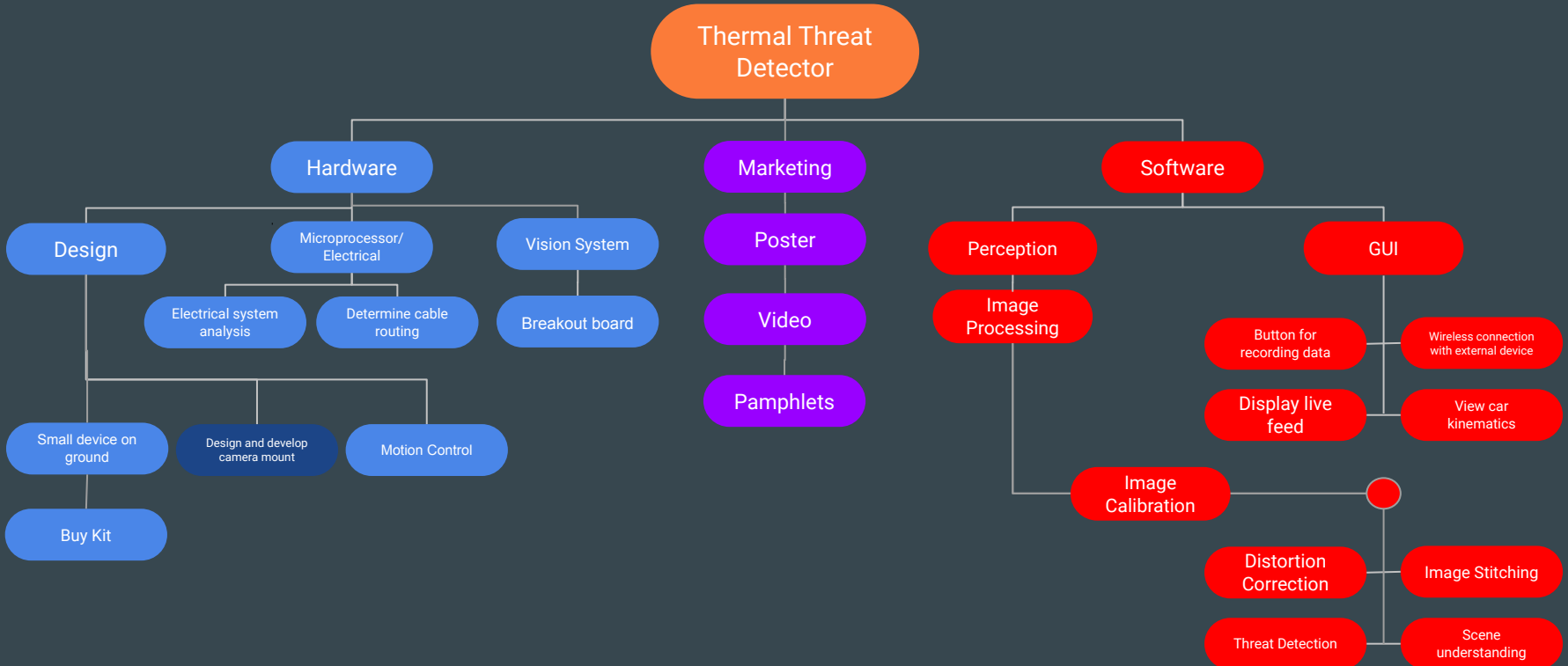
Baseline	Criteria	Weight	Camera					
			Lepton 3.1R	FLIR K1	FLIR C3-X	SN-TPC4201VT-F (III)	Klein Tools Pro	HIK Vision
\$2000 total	Price	5	Better	Same	Worse	Worse	Same	Worse
160x120 px	Resolution	5	Same	Same	Worse	Better	Better	Same
97.6-99.6	Temperature Range	5	Better	Better	Better	Better	Better	Better
50-60	Field of View	5	Better	Same	Same	Worse	Same	Worse
1 hour	Power	3	Better	Better	Better	Worse	Worse	Worse
Per user manual	Easy-to-setup	1	Worse	Better	Same	Worse	Same	Same
Yes/No	Rotate	2	Worse	Worse	Worse	Worse	Worse	Better
Per website and description	Durability	1	Same	Better	Better	Same	Better	Same
Based on year released	Reasources/online info	2	Worse	Same	Same	Same	Same	Same
Would it be easy to create a mount	Camera size/shape	3	Same	Worse	Better	Better	Better	Same
Yes/No	Stream on it's own	2	Worse	Worse	Worse	Better	Worse	Worse
	Better		4	4	4	4	4	2
	Same		3	4	3	2	4	5
	Worse		4	3	4	5	3	4
	Weighted Better		18	10	12	15	14	7
	Weighted Same		0	0	0	0	0	0
	Weighted Worse		-7	-7	-14	-16	-7	-15
	<b>Overall Score</b>		11	3	-2	-1	7	-8
	<b>Best Decision</b>	<b>Lepton 3.1R</b>						

# Cameras - What we are using

- Lepton 3.1R
  - 95° HFOV & 71° VFOV allowing for 190° total HFOV and 156° VFOV
  - 160x120px frame size
  - \$142 per camera x 3 cameras
    - \$426 total
    - Price of all three Lepton cameras is less than the price of one camera for any of the other brands



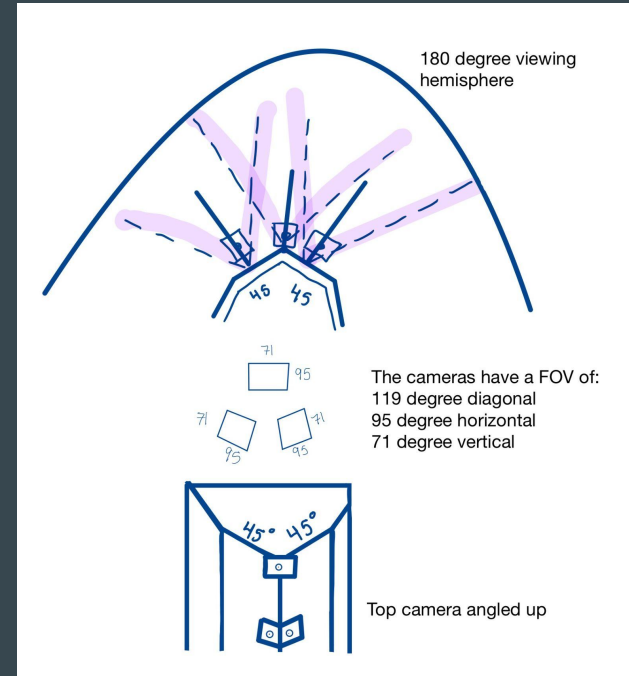
# Work Breakdown Structure: Camera Mount



# Initial Camera Mount Design

- Each Lepton Camera had:
  - 119 degree diagonal
  - 95 degree horizontal
  - 71 degree vertical

-The mount was designed to achieve a 180\* horizontal field of view and a 156\* vertical field of view

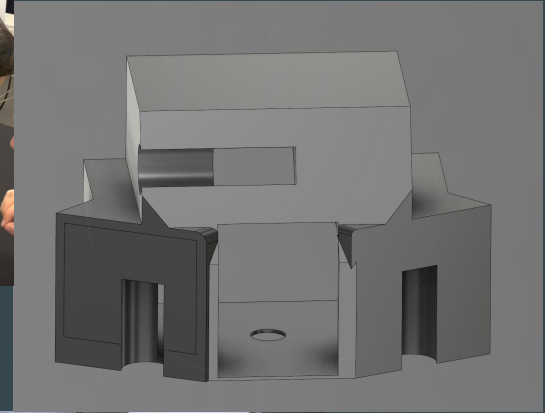


# Camera Mount Testing

-When testing FOV, placements were marked on the floor where the edge of the frame was for each camera.

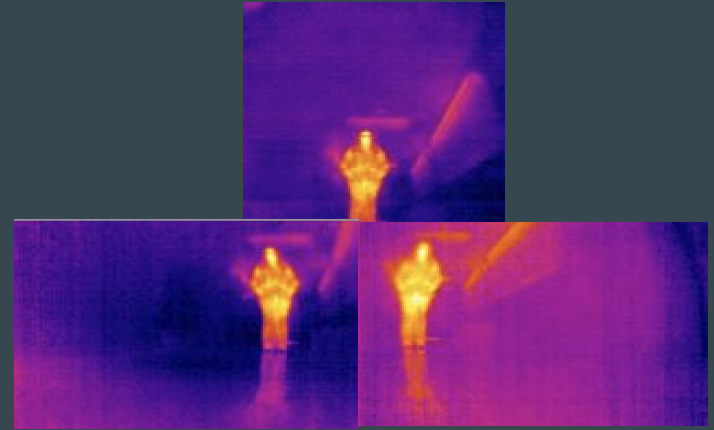
-The angle was measured from the outside marks

-Camera mount is being finalized with adjustments calculated from testing.





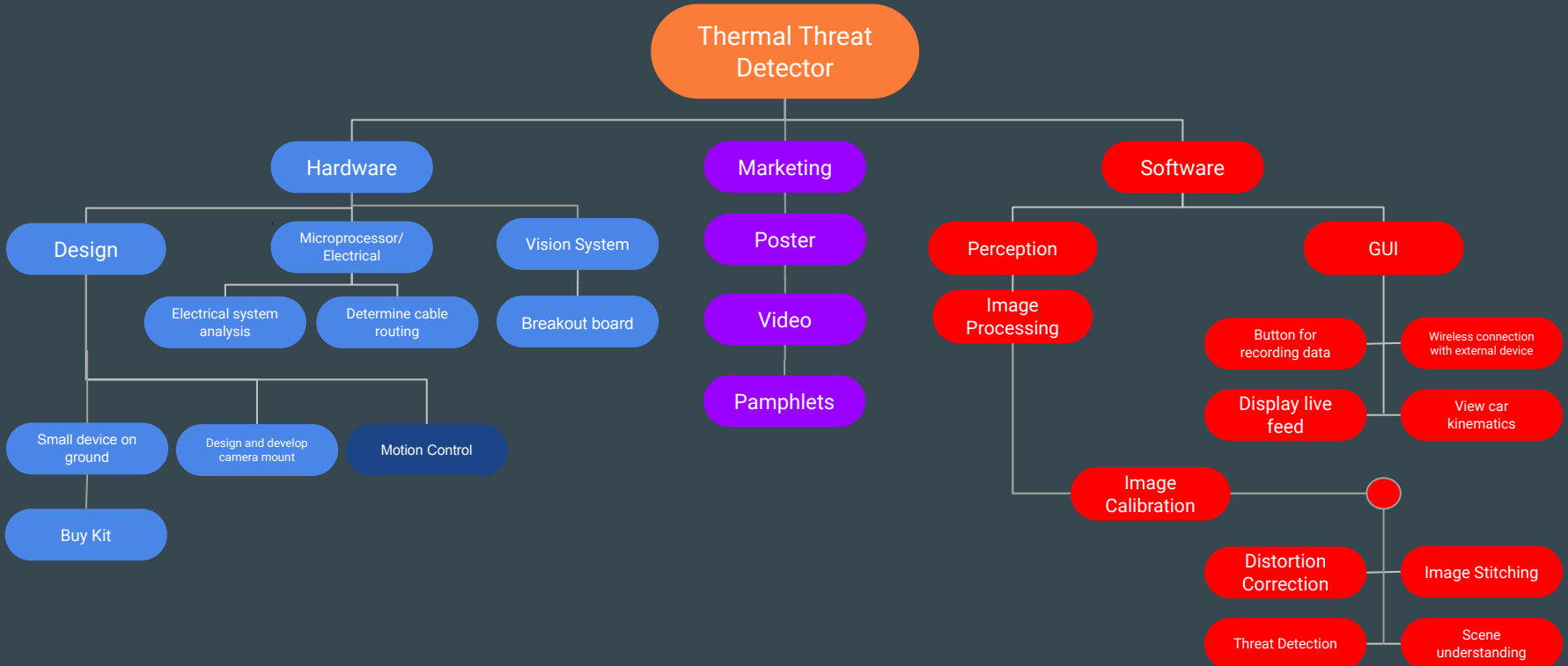
# Current Mount FOV and Overlap



-Due to complications with original image stitching, the overlap on our reprint was overcompensated.

-With our new mount, the angle degree of outside cameras will be increased, decreasing overlap and focusing on obtaining our 180° FOV

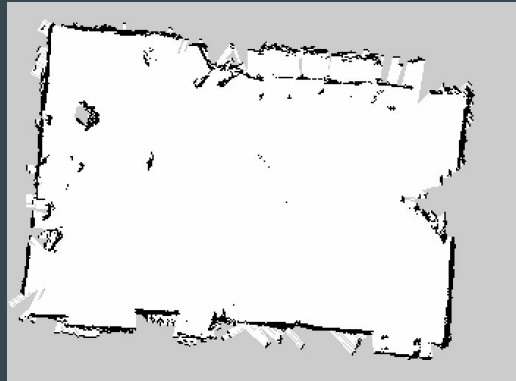
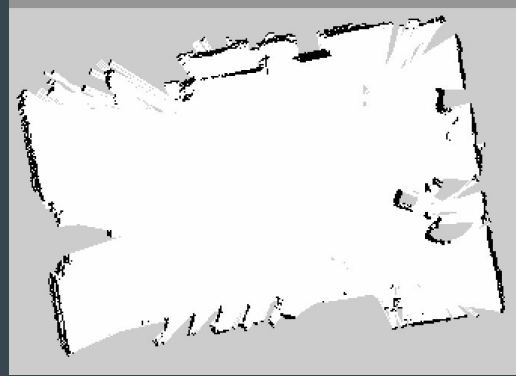
# Work Breakdown Structure: Motion Control



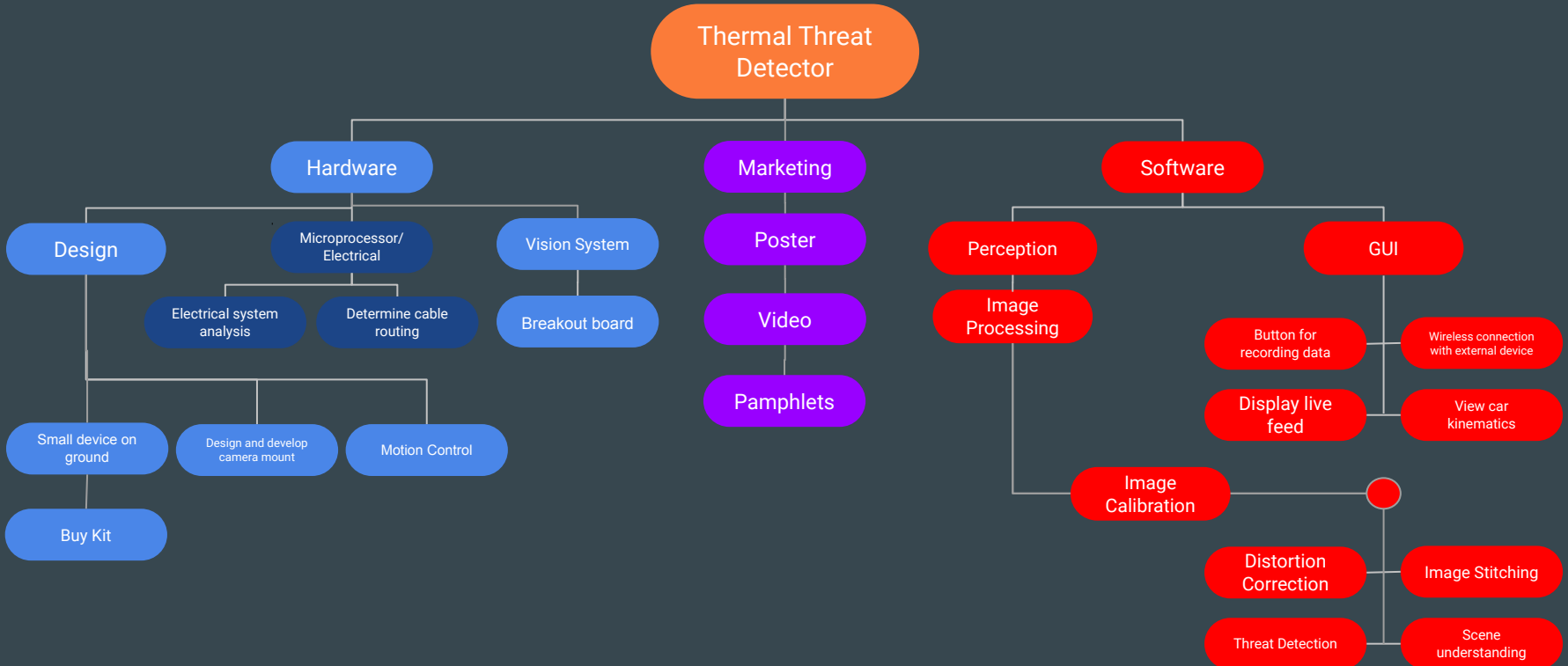
# Motion Control

## Lidar mapping

- Conducted SLAM mapping tests within the Gannett studios. Able to create the map with both manual and autonomous control
- Allowed for autonomous movement, and obstacle avoidance.
- Currently implementing vehicle control into the GUI



# Work Breakdown Structure: Microprocessor



# Microprocessor - Jetson Nano

Complimentary SDK:

- JetPack
- CUDA Toolkit
- cuDNN



Seamlessly integrates with frameworks for computer vision and robotics development like OpenCV and ROS

Because of the processing power, it gives developers a lot of headroom to design and debug while still aligning with the budget.

Topics	Criteria	Weight	Raspberry Pi	Jetson Nano	Jetson Orin Nano	Coral
Processing	GPU	4	Worse	Same	Better	Better
	CPU	1	Same	Same	Better	Same
	RAM	4	Better	Same	Better	Worse
Cost	Processor Cost	4	Better	Same	Worse	Same
Software	SDK	5	Worse	Better	Better	Worse
Hardware	Power Efficiency	4	Better	Worse	Worse	Better
Expandability	USB 3.0 Ports	3	Same	Better	Better	Worse
	Storage	3	Same	Same	Better	Same
Convenience	Vehicle Compatibility	5	Same	Better	Same	Same
	Better		3	3	6	2
	Same		4	5	1	4
	Worse		2	1	2	3
	Weighted Better		12	13	20	8
	Weighted Same		0	0	0	0
	Weighted Worse		-9	-4	-8	-12
	<b>Overall Score</b>		3	9	8	-4

# Charging

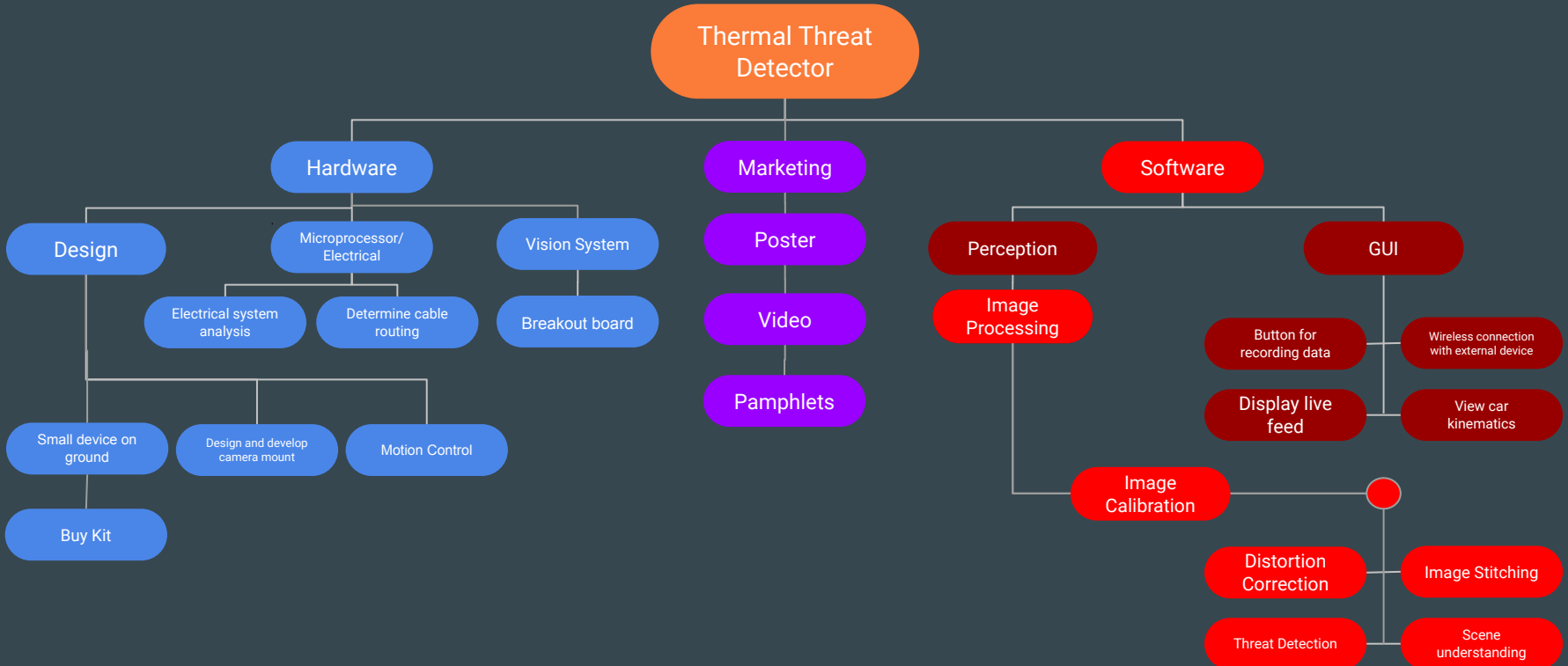
## Initial Testing: 4hr Battery Life

- Ordered two additional batteries to change while the vehicle is operating so the system can last for all of Imagine RIT (8hrs)
- Lastly have to solder the new wire connection and mount the new battery to the vehicle.



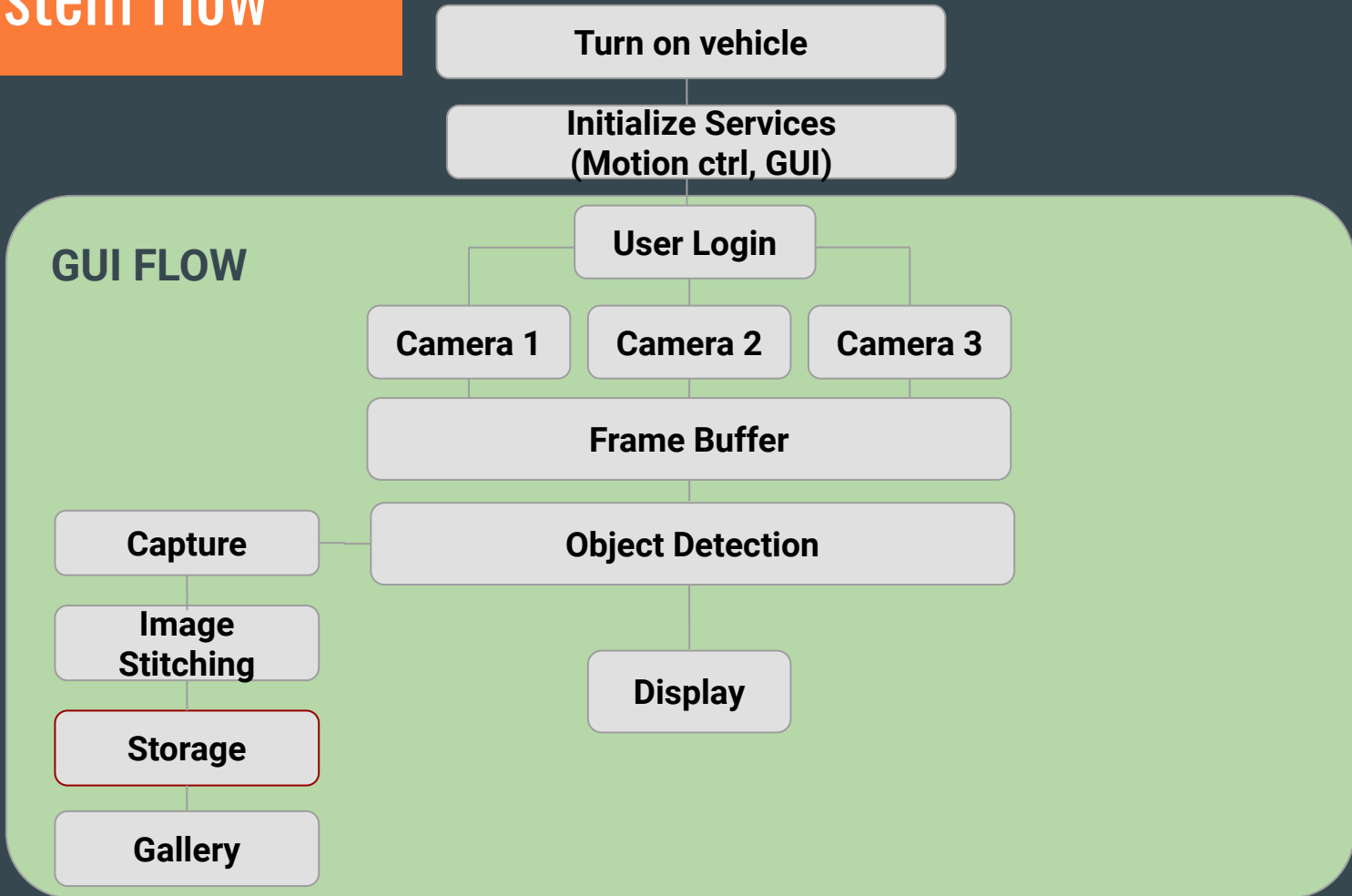
# Software Updates

# Work Breakdown Structure: GUI



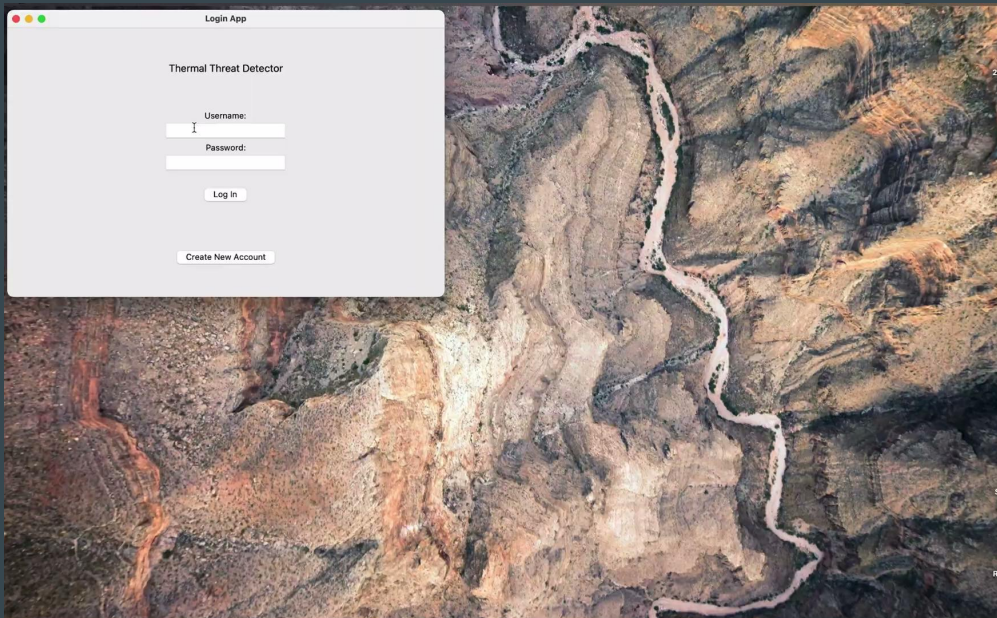


# System Flow



# GUI

- Python script for most functions are completed
- Utilize virtual database for user logins
- Camera's input for live feed
  - Object Detection
- Captured image saved in the Gallery
  - Image Stitching



Working on: syncing 3 cameras together

# GUI

- Using HiWonder to connect with the vehicle
  - Bash script created to start GUI when vehicle turns on
- Vehicle Control
  - Implemented in the GUI
  - User can use keyboard (w, a, s, and d) to control

```
#!/bin/bash

# Open new terminal - enable service
terminator -x bash -c './enable_service.sh'

# Wait for a bit to ensure the first command starts properly (optional, adjust as
needed)
sleep 2

# Open new terminal - start gui
terminator -x bash -c './start_gui.sh'

# Open new terminal - vehicle control
terminator -x bash -c './start_mapping.sh'

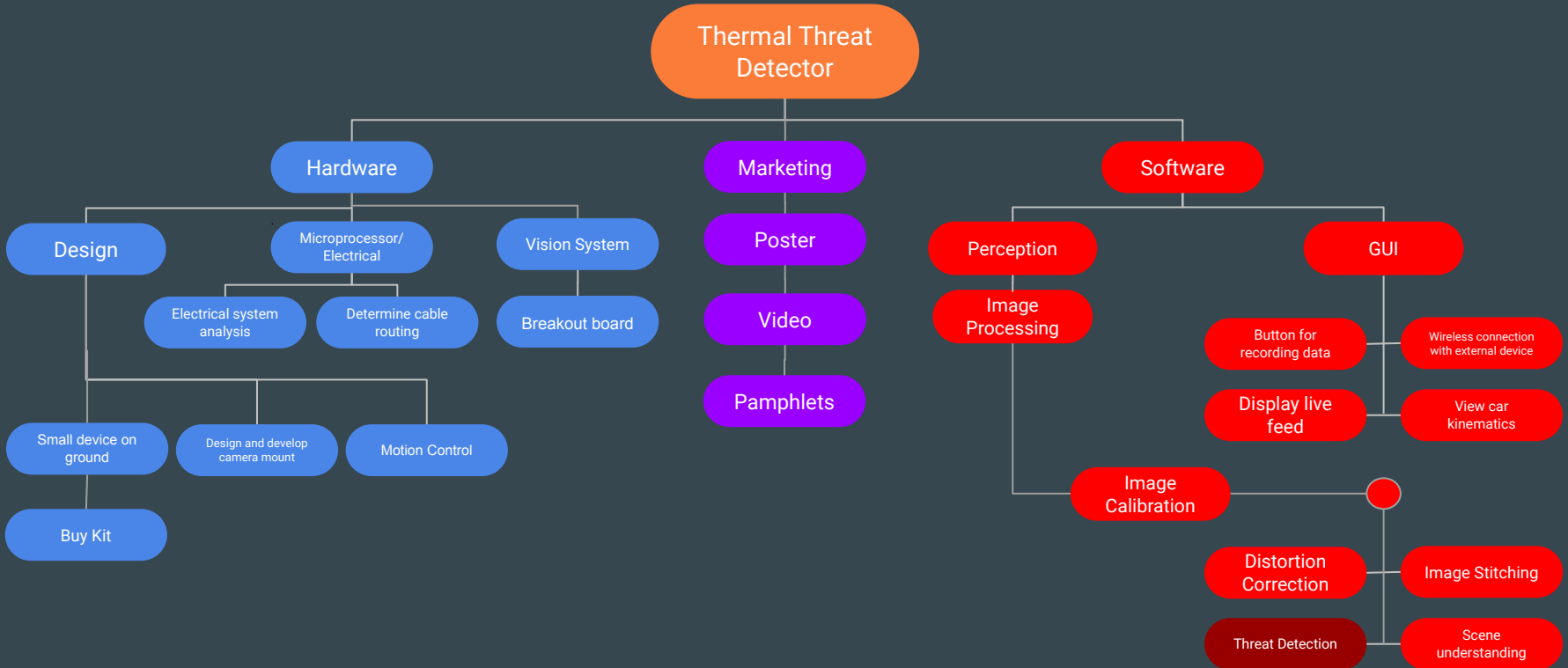
# Open new terminal - save map
terminator -x bash -c './save_map.sh'
```



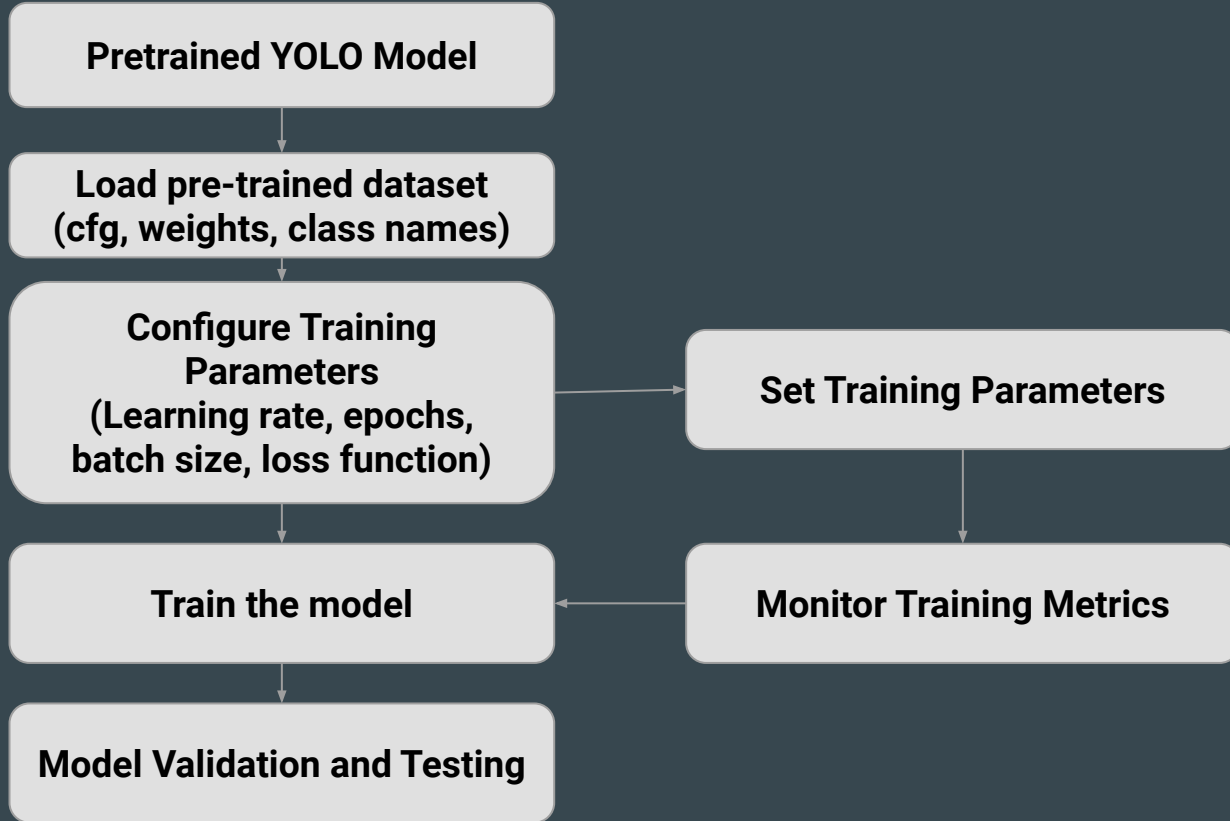
**Moving around:**  
**w**  
**a s d**  
**CTRL-C to quit**

OK

# Work Breakdown Structure: Threat Detection



# Object Detection - Training



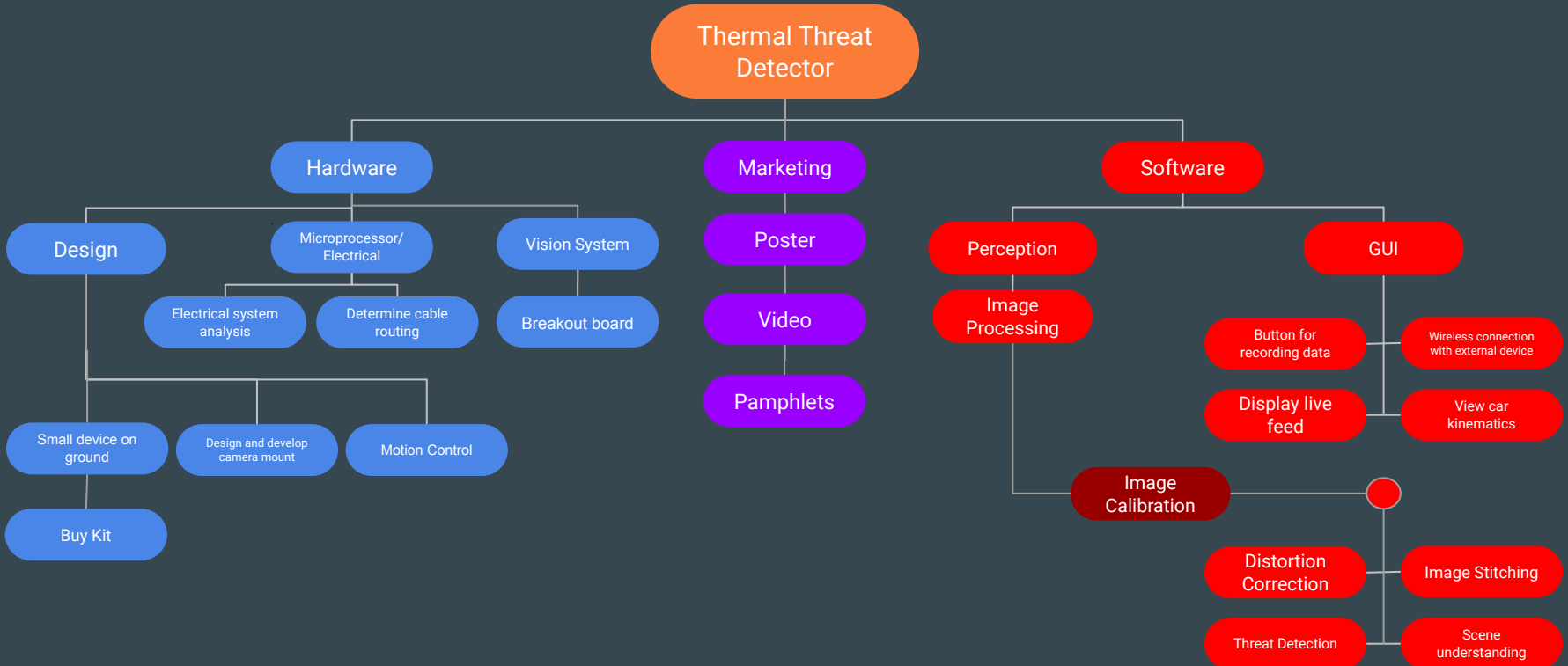
# Object Detection - Results



Used Pre-trained YOLOv3 Model, trained with Flir ADASv2 Dataset

- Capable of detecting people and 14 other distinct objects
- Struggles to resolve people when they are far away from the camera
- Need to retest module with updated calibration

# Work Breakdown Structure: Image Calibration



# Benefits of Calibration

- To know what colors correspond to what temperature in a captured image after applying the color palette
- To know the pixel's response at each temperature
- This can help with image stitching as the three cameras will be able to display the same colors for the same temperatures



# Calibration Process

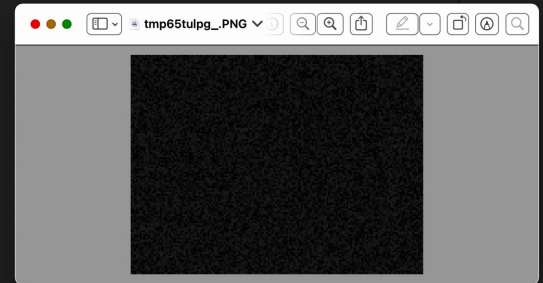
- When you take images pointed at the blackbody radiator, you will know the pixel counts that correspond to that certain temperature (these pixel counts will be different for each camera because they are not calibrated the same)
- Take 10 images and combine them to reduce noise to get “clean” images to work with

# Calibration Results

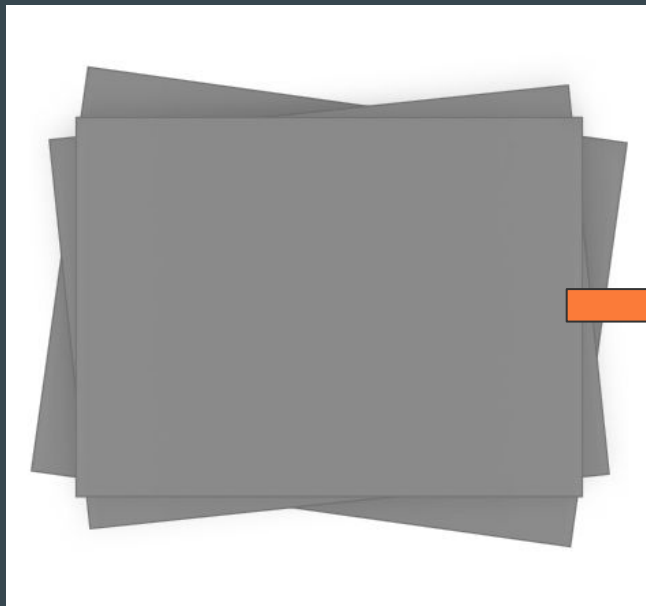
- Gathered data from blackbody radiator, taking 10 images at
  - 10, 20, 30, 40 Celcius
- Step 1: Program successfully combined 10 tiff images into one tiff for each camera and temperature



```
1 import numpy as np
2 from PIL import Image
3
4 # Open the 10 images
5 image_files = ['/Users/gabriellafatigati/Desktop/1_P5_Capstone/PhotoScienceCal/0c/LeptonCap_A_0_*.tiff' for i in range(1, 11)]
6
7 images = [np.array(Image.open(image)) for image in image_files]
8
9 # Combine the images along the fourth axis to keep the color channels
10 combined_image = np.mean(images, axis=0, dtype=np.uint8)
11
12 # Create a PIL Image from the combined NumPy array
13 combined_image = Image.fromarray(combined_image)
14
15 # Save the combined image
16 combined_image.save('combined_image_color.jpeg')
17
18 # Optionally, display the combined image
19 combined_image.show()
20
21
```



# Calibration - Step 1 Results



# Calibration Final Steps

- Extract pixel values
- Calibrate the pixel values to the temperature we said it was
  
- Create a linear regression model showing what pixel values correspond to what temperatures
- Apply gain and offset to uncalibrated images
- Display a calibrated image

# Calibration Issues

- Program is not handling tiff files (resizing and reshaping issue with newly collected data)
- Continuing troubleshooting to fix code

```
Users > gabriellafatigati > Desktop > 1_PS Capstone > PhotoScienceCal > linearRegress2 > ...
80
81     result = cv2.imwrite(os.path.join(resultsDir, 'slope.tif'), slope)
82     result = cv2.imwrite(os.path.join(resultsDir, 'yIntercept.tif'), yIntercept)
83     result = cv2.imwrite(os.path.join(resultsDir, 'rValue.tif'), rValue)
84     if result:
85         print('Files saved successfully.')
86     else:
87         print('Error in saving files.')
88
89     return slope, yIntercept, rValue
90
91 if __name__ == "__main__":
92     maxTemp = 50
93     minTemp = 10
94     steps = 2
95     path = '/dirs/data/uas/uas_data/FLIR_calibration/Rec-000006/averagedImages/'
96     resultsDir = '/dirs/data/uas/uas_data/FLIR_calibration/Rec-000006/statistics/'
97     temps = np.flip(np.arange(minTemp, maxTemp + 1, steps))
98     tiff_file = '/Users/gabriellafatigati/Desktop/1_PS Capstone/PhotoScienceCal/Camera C/combined_C_40.tiff'
99     slope, yIntercept, rValue = linearRegress(path, temps, resultsDir, tiff_file)

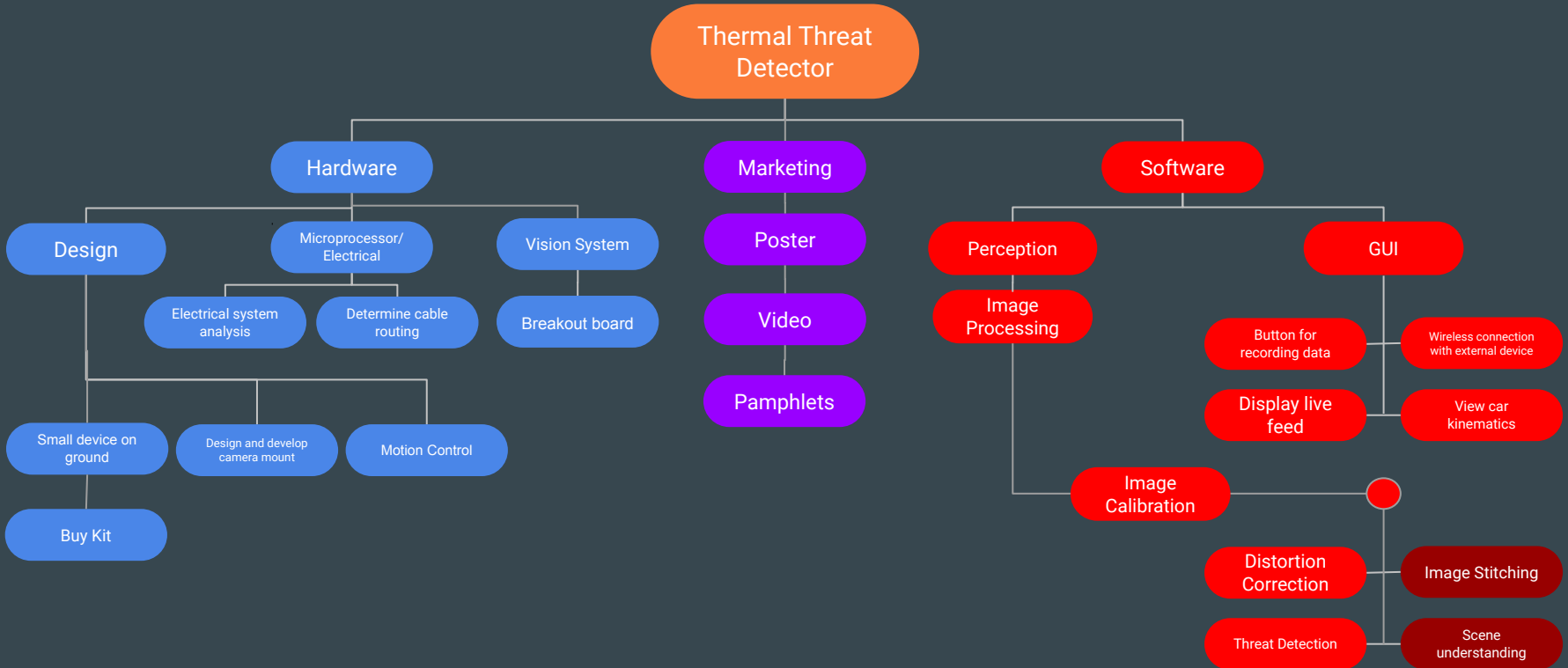
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Shape of c before reshaping: (1,)
Shape of c after reshaping: (1,)
Shape of c before reshaping: (1,)
Shape of c after reshaping: (1,)
Traceback (most recent call last):
  File "/Users/gabriellafatigati/Desktop/1_PS Capstone/PhotoScienceCal/linearRegress2", line 99, in <module>
    slope, yIntercept, rValue = linearRegress(path, temps, resultsDir, tiff_file)
  File "/Users/gabriellafatigati/Desktop/1_PS Capstone/PhotoScienceCal/linearRegress2", line 78, in linearRegress
    os.makedirs(resultsDir)
  File "/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/os.py", line 215, in makedirs
    makedirs(head, exist_ok=exist_ok)
  File "/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/os.py", line 215, in makedirs
    makedirs(head, exist_ok=exist_ok)
  File "/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/os.py", line 215, in makedirs
    makedirs(head, exist_ok=exist_ok)
  [Previous line repeated 3 more times]
  File "/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/os.py", line 225, in makedirs
    mkdir(name, mode)
OSError: [Errno 30] Read-only file system: '/dirs'

[Done] exited with code=1 in 1.12 seconds
```

## Calibration Success in Real Time

- When shooting live, we can apply the gain and offset to each pixel after capturing an image (which is calculated from the program)
- Then the same temperature on all three cameras will show as the same color
- Can check this by selecting the pixel values and comparing

# Work Breakdown Structure: Image Stitching



# Image Stitching - How it works

Program written in python to stitch images together:

## 1. Imports Images

## 2. Resizing Function

- Maximizes the size on each images (1000 pixels)

## 3. Stitch Image

- **Detects key points** in images using SIFT (Scale-Invariant Feature Transform), to match key points.
- Uses ratio test to **determine if points are a good match.**

## 4. Homography Estimation

- Estimates the homography matrix using RANSAC (Random Sample Consensus) algorithm. Uses Brute Force (BFMatcher) to align points. (This uses the data points to determine if there are any outliers).

## 5. Warping

- Warping the second image to first using the matrix.

## 6. Blending

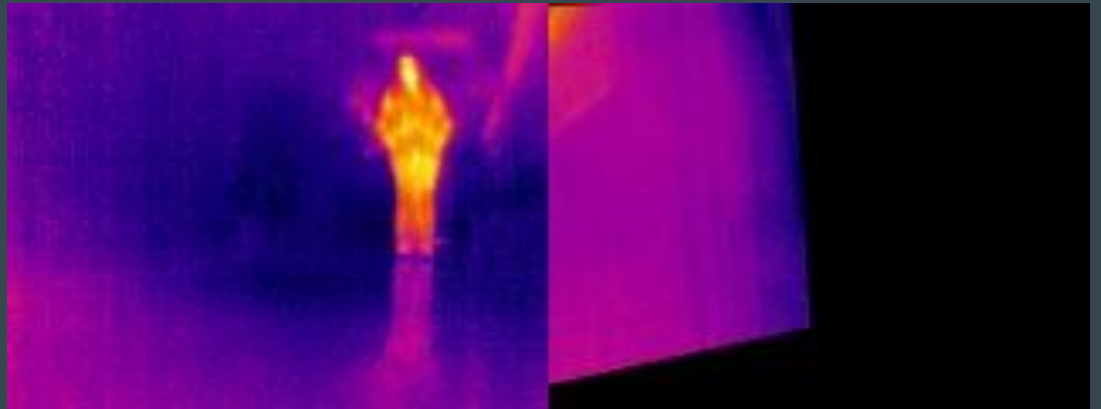
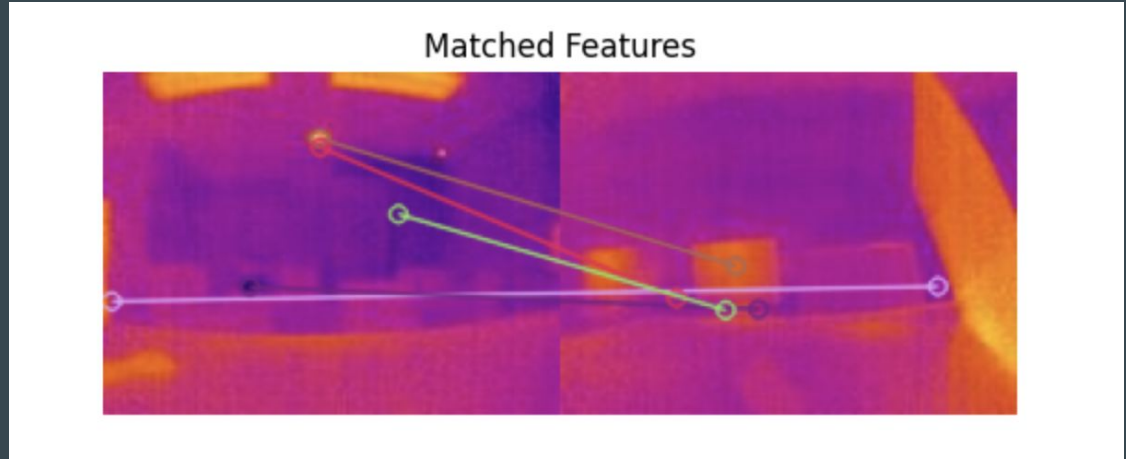
- Seamless integration

## 7. Results/Output



## Image Stitching - Where we left off

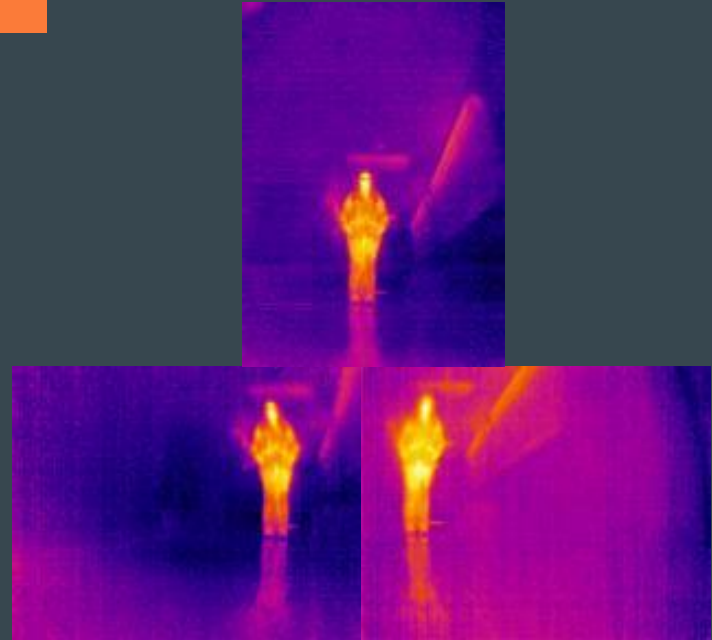
- An error indicated there was not enough matches present to stitch the images.
- To address this issue various image sets were tested. Including black and white images to see if the thermal was impacting the results.
- Same error remained determining more overlap was needed as well camera calibration.



# Imaging Stitching - Moving forward

We know how to complete the images stitching using:

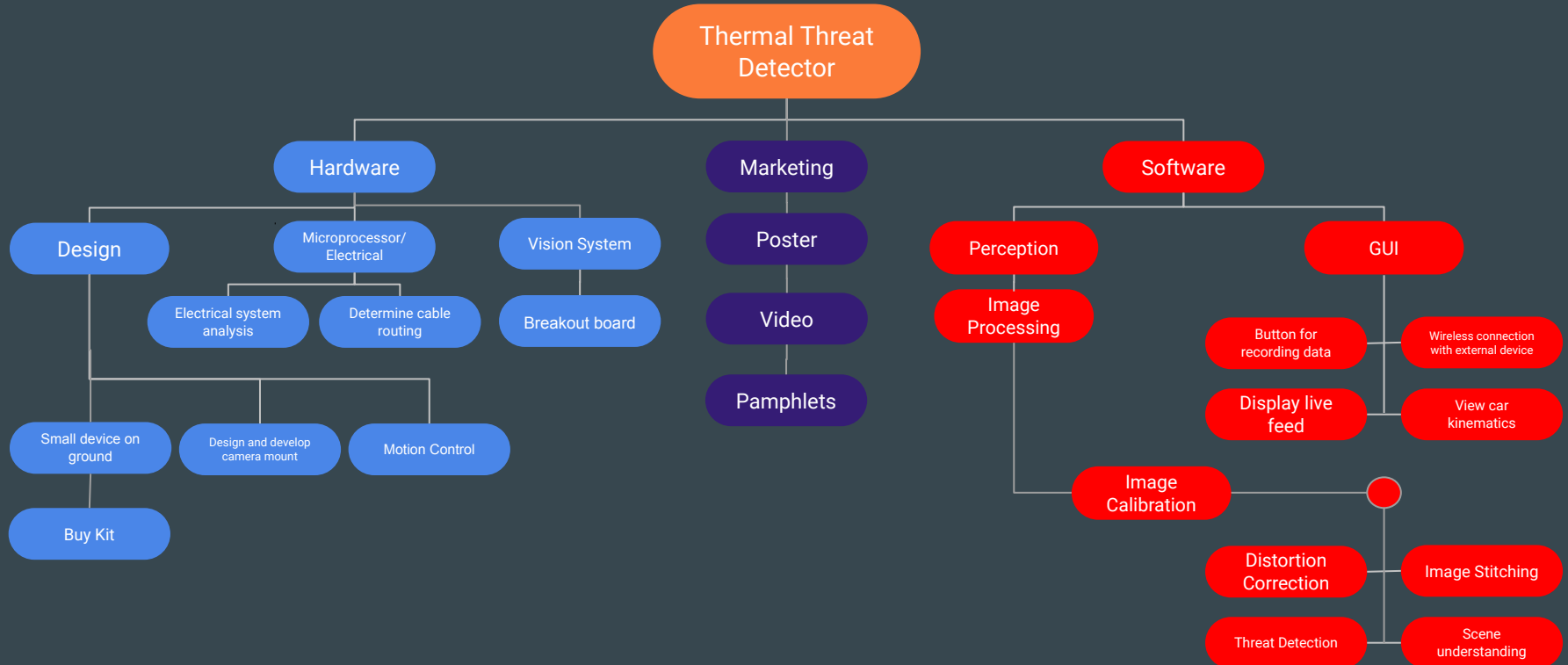
- The new RAW calibrated images.
- Completed matrix will be applied to every stitched image as cameras are in a fixed position.
- Once complete, can be implemented into the image processing pipeline.



Once calibration is complete colors will be standardized allowing the images to be stitched.

# Marketing Updates

# Work Breakdown Structure: Camera Mount



# Marketing/Imagine RIT Updates

- Poster: **Finished**
  - Headshots
  - QR Code
  - Image of Device
- Video: **Filmed & Now Editing**
  - Interviews
  - Overview of Class
  - B Roll
  - Demonstration
- Pamphlets and Brochures: **In Process**
  - 'What is Thermal Imaging?'
  - 'What is Capstone?'
  - Hardware and Software Page

# Marketing/Imagine RIT Updates



## RIT | College of Art and Design Photographic Sciences Senior Capstone Class of 2024's Thermal Threat Detector

### Project Information

Thermal imaging is an imaging techniques used in various ways such as search and rescue, inspections and in the medical field. Photo Science seniors were tasked to create an imaging device that can enter a room, detect a threat, capture and send thermal images back to the user in another room. We have done so by fusing an autonomous vehicle with three thermal cameras and programmed software that allows us to view 180 degrees inside the room.



### ImagineRIT Booth

At our exhibit, you can see our thermal threat detector in action as we will be giving a live demonstration of the vehicle's capabilities. You can also view yourself through the lens of a thermal camera. Come talk to the Photo Science seniors of the class of 2024, and they may even let you take the wheel!

### Project Collaborators



Sam Allen Madeline Dowe Gabriela Fatigati Leanna Herrick



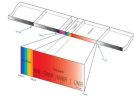
Sue Kim Xyron Neumann Jared Redington Annie Schmitt Ma Taslemos

### Event Information

April 27th, 2024  
10:00 AM - 5:00 PM  
Capstone Work Room  
Gannett Hall 07B - 2271

### What is Thermal Imaging?

Thermal imaging allows us to view infrared radiation emitted from an object or person, allowing the user to see past the "visual spectrum". The camera converts infrared radiation data to create an image with color palettes representing temperature differences. This system is ideal for low-light and dark situations!



With the Lepton 3.1R, thermal infrared lies just past the visible light spectrum with wavelengths ranging between 8 - 14  $\mu\text{m}$ .



### What is Photographic Sciences?

Photographic Sciences is a unique major to RIT, blending interests in both science and the arts! It's a dynamic field with hands-on classes that explore the capture and collection of scientific data. This program combines photography with imaging science, information technology, computing, optics, and biomedical sciences.

### Meet The Team

**Hardware:**  
Madeline Dowe, Leanna Herrick, Ma Taslemos

**Software:**  
Sam Allen, Gabi Fatigati, Sue Kim, Xyron Neumann, Jared Redington, Annie Schmitt



## Photographic Sciences Capstone 2024

### Thermal Threat Detector

RIT College of Art and Design  
Photographic Sciences

### Software

One of the most important responsibilities of the software team was creating the GUI, graphic user interface. The GUI is the python code for the system that is providing our visual for the data that is coming from the cameras, and it implements features such as: live view, vehicle controls, object detection, and image stitching. With the GUI, the user engages with the components of the hardware team to capture a stitched 180-degree thermal image. When the user logs in to their account, they will be presented with a live view of the three thermal cameras. From there, object detection will allow the user to better identify the threat. An image can then be captured and three individual images from each camera will be stitched into a single thermal image.

### Hardware

This team was responsible for selecting, testing, and assembling all hardware aspects of this project. First, each component including the camera, vehicle, and batteries were thoroughly researched for the most fitting products to be purchased for the system. The jackster vehicle was the first main component that the team purchased followed by the Lepton 3.1R thermal cameras and PureThermal breakout boards. Later, additional batteries and materials including velcro for mounting, screws and chargers were chosen.

Another big responsibility of the hardware team was to design and print the mount which attaches the three thermal cameras to the vehicle. During this process, multiple prototypes were created and field of view testing was performed on the positioning of the cameras. The final product was printed to achieve the 180 degree field of view for the system. Autonomous movement testing, power supply testing, and camera data collection include some of the other responsibilities of this team.



### Our Project

Clearing a room can put an enforcement team into a high-risk situation, being able to enter and search for a potential threat with a vehicle is a safer alternative.

The Photographic Science seniors were tasked to create an imaging system that could enter a low-light room, and capture thermal images. Once collected, the images are sent back to the operator in a safe adjacent room upon detection of a potential threat. This task has been accomplished through the integration of an autonomous vehicle with three thermal cameras and the development of software enabling comprehensive 180-degree room visualization.



Initial class photo taken with the Lepton 3.1R thermal camera.



# PROJECT: Thermal Threat Detector

## Photo Science Capstone 2024

Project start date: 2/9/24

Scrolling increment: 5

Legend: On track Low Risk Med Risk Late/Behind Unassigned

Milestone description	Category	Assigned to	Progress	Start	End Goal
-----------------------	----------	-------------	----------	-------	----------

### Hardware Team

Print camera mount	Goal	Leanna	90%	2/1/24	
vehicle testing	Med Risk	Leanna, Maddie, and Mia	50%	2/2/24	
Vehicle Control	High Risk	All software Team	30%	7/16/29	
Battery Life	Low Risk	Leanna, Maddie, and Mia	100%	1/30/24	
Train LiDAR	Med Risk	All software Team	10%	2/13/24	

### Software Team

GUI Design	On Track	Annie and Sue	60%	1/1/24	
Image Stitching	On Track	Sam	70%	1/3/24	
Camera Control	High Risk	Jared, Gabi, Xyron	33%	1/8/24	
Calibration	Med Risk	Gabi	70%	3/1/24	
Object Detection		Jared	50%		
Intergration	Milestone	All software Team	0%		
Finalize	Milestone	All software Team	0%		

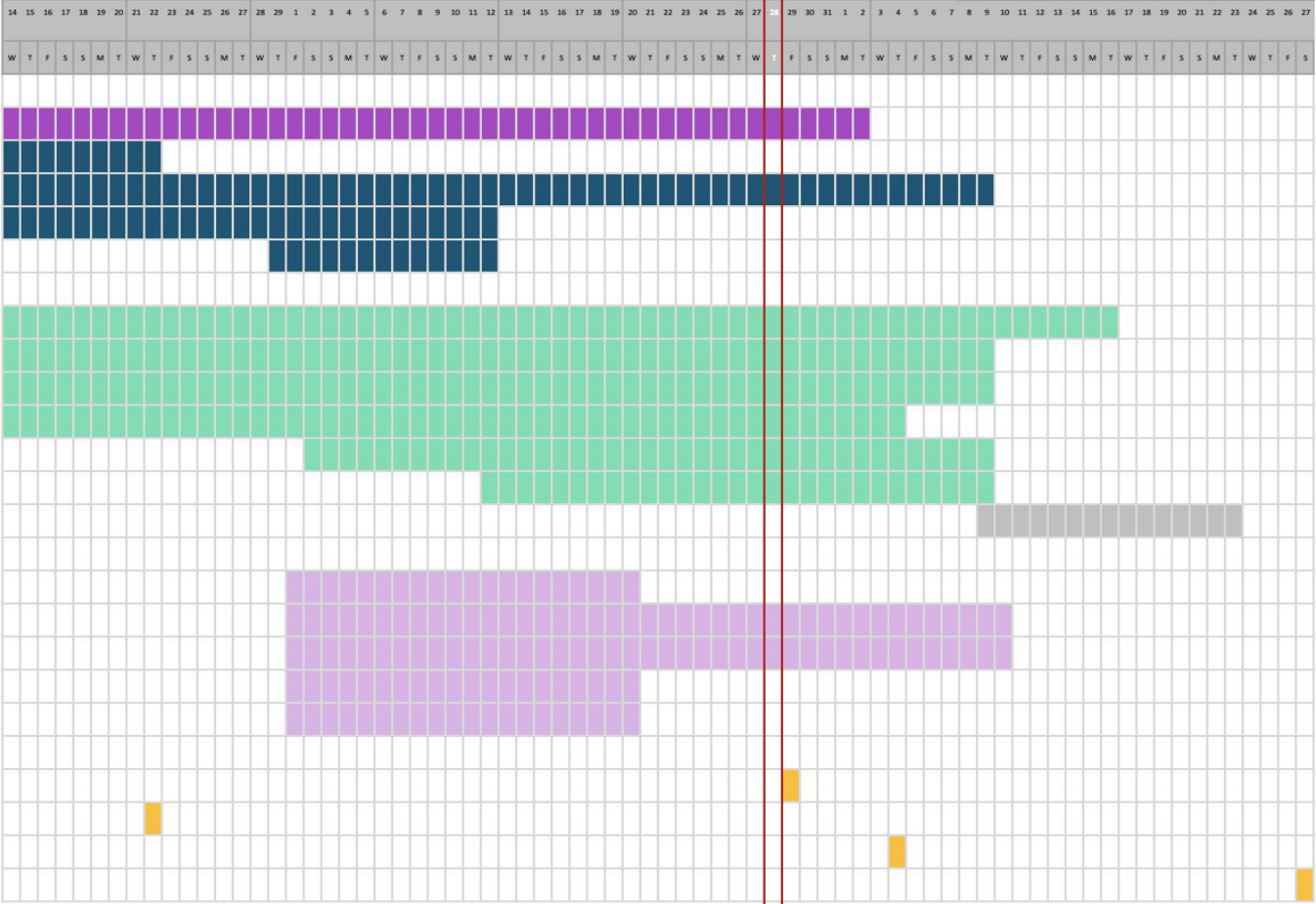
### Marketing/Paper

Imagine RIT Poster		Gabi	50%		
Video		NBD	0%		
Report		Sam And Annie	0%		
Pamphlet		Gabi	100%		
Technical Manual		NBD	0%		

### Presentations

Critical Design Review		Sam, Annie, and Jared		3/29/24	
Faculty Meeting		Maddie, Mia, and Sue		2/22/24	
Critical Design Review		Leanna, Gabi, and Xyron		4/4/24	
Imagine RIT				4/27/24	

### February March April



# Requirements Checklist

Requirement	Requirement Minimum	Thermal Threat Detector
Transportation	Manual	Manual
Power	1 hr	<b>4 hr</b>
Field of View	180*	~177*
Resolution	Capable of resolving a human	Capable of resolving a human in near distance
GUI	Allows user to capture image	Allows user to capture image



# Remaining Work

- Retest image stitching and object detection modules (post calibration)
- Integrate three cameras with GUI
- Documentation
- User manual, technical manual, and research paper
- Marketing for Imagine RIT

# Questions?