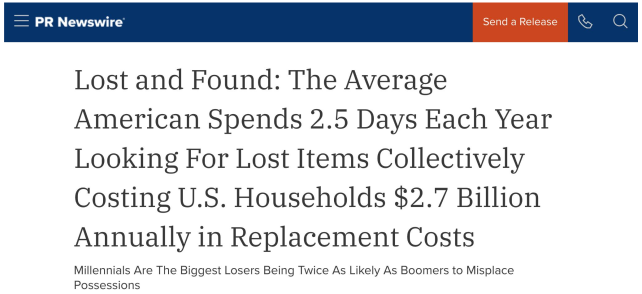
Use Case

Device:

Localized object location tracker

Application:

Developing a POC for a camera-based object tracking system for indoor environments to locate personal items such as keys, wallets and remotes.



Lost and Found: The Average American Spends 2.5 Days Each Year Looking For Lost Items Collectively Costing U.S. Households $2.7 Billion Annually in Replacement Costs

Millennials Are The Biggest Losers Being Twice As Likely As Boomers to Misplace Possessions

Problem:

Issue of frequent loss and misplacement of personal items

Audience:

Students, Parents

Alzheimer’s and Dementia patients

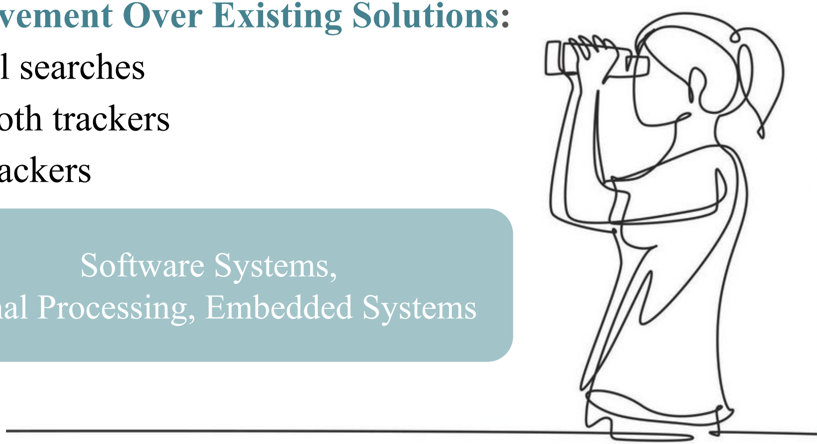
Improvement Over Existing Solutions:

Manual searches

Bluetooth trackers

GPS trackers

Software Systems,
Signal Processing, Embedded Systems



Pic Cred: Google Images

Use Case Requirements

Any more and false positives will waste too much of the user's time

Object Detection

- 80% accuracy (>iPhone, well-lit 10x10ft room)
- Detection and tracking process completed within 5 minutes.

Latency for processing, querying and llm response

Nooks and corners

Camera Setup

- 80% floor visibility to minimize blind spots

Personal information kept safe with encryption

Privacy

- Authorized access on authenticated local devices only

Cost Constraints

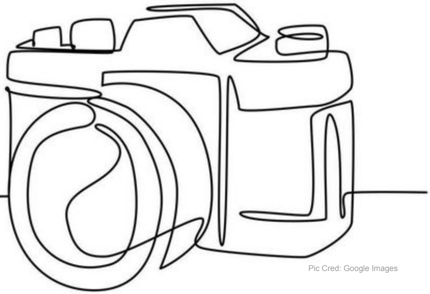
- Hardware cost <$300.
- Cloud costs <$1000 per user/month for continuous usage.

Low scale production

User satisfaction and POC

Query Response

- Users receive item location within 20 seconds of querying, based on data at most 30 seconds before time of query



Pic Cred: Google Images

Use Case Unknowns

1. **Power:** Power consumption for edge devices (Rpi/ Jetson)
2. **Network:** Bandwidth for data synchronization between cameras is unknown
3. **Storage:** Data storage locally versus on the cloud
4. **Compute:** Cloud machine requirements depend on usage, impacting service pricing
5. **API Queries:** Costs vary based on API call frequency, affecting pricing and viability

Technical Challenges

1. Object detection/Audio translation accuracy

- Dependent on the accuracy and reliability of ML object detection models. Further training on new datasets maybe required.
- There will likely have to be pre-processing on audio in noisy environments.

2. Power delivery

- Sufficient power must be delivered to the edge compute in a way that must be convenient for the end user.

3. System latency

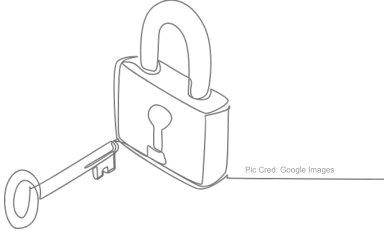
- Time between when a user request is made and fulfilled must be reasonable
  - Database query,
  - Serve an image,
  - Generate any LLM results.
- Latency between taking an image, and its processed results being present in the database.

4. Privacy

- System must ensure user images remain private, secure from others

Pic Cred: Google Images

Technical Challenges: Risk Mitigation



Pic Cred: Google Images

1. Object detection accuracy/precision

- If a “latest ___ recorded” is not always accurate, we can store a history, and make it easy for a user to see it.
- Alternative of user-text input

2. Power delivery

- Many possible choices
  - One of them will hopefully work
- Battery powered/ wall outlet

3. System latency

- Query time:
  - Bottlenecks are clear in our case, and can be made optional/turned off
- Analysis Time
  - Likely throughput issue, add more machines

4. Privacy

- Incorporate authentication before access database
- Local data processing (Jetson)

Risk Mitigation

Early testing

Fail fast and change course

Continuous component-level regression testing

Model accuracy, system throughput, maintain quality of system

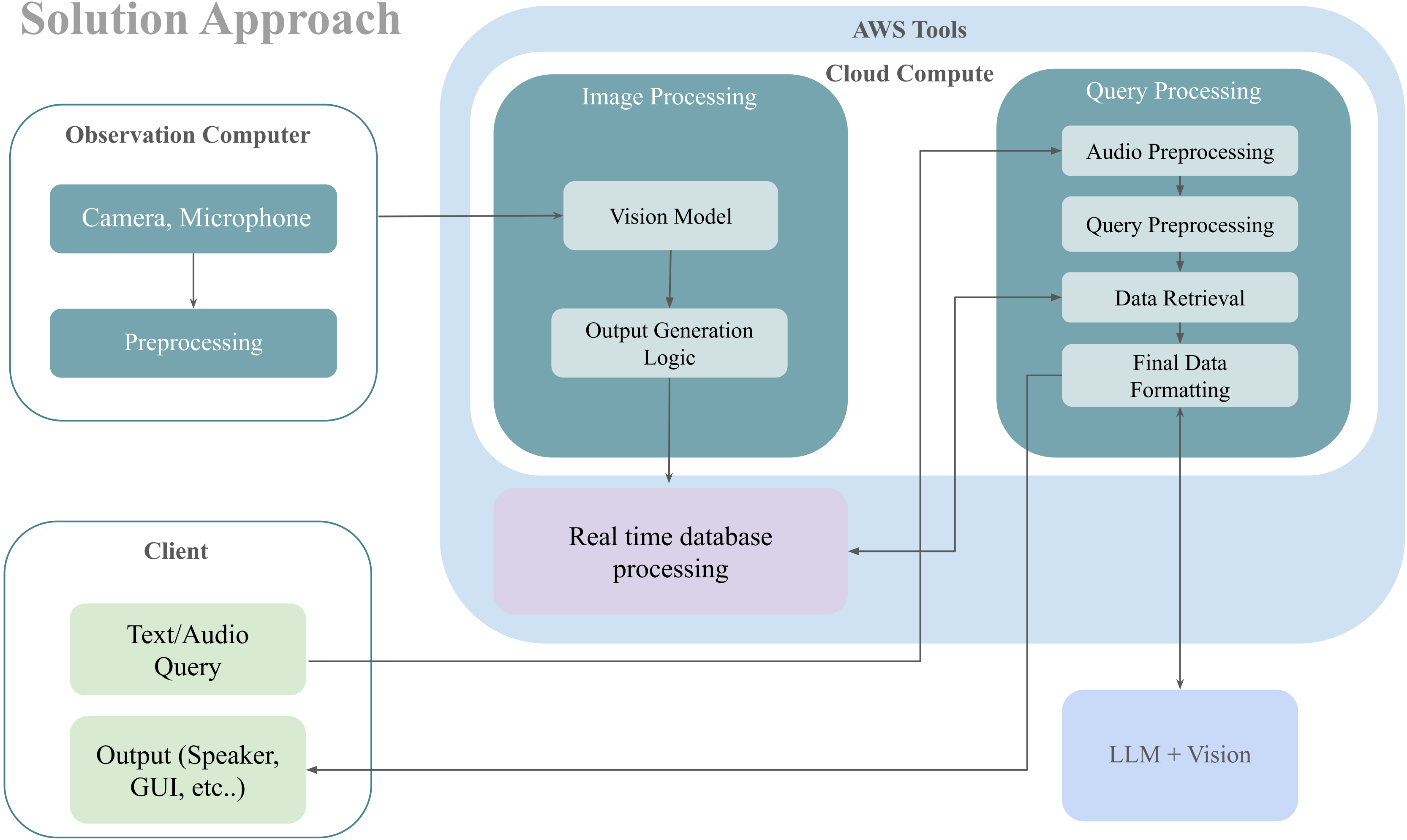
Fine-tuning

Regularly evaluate alternative options on modular and system level

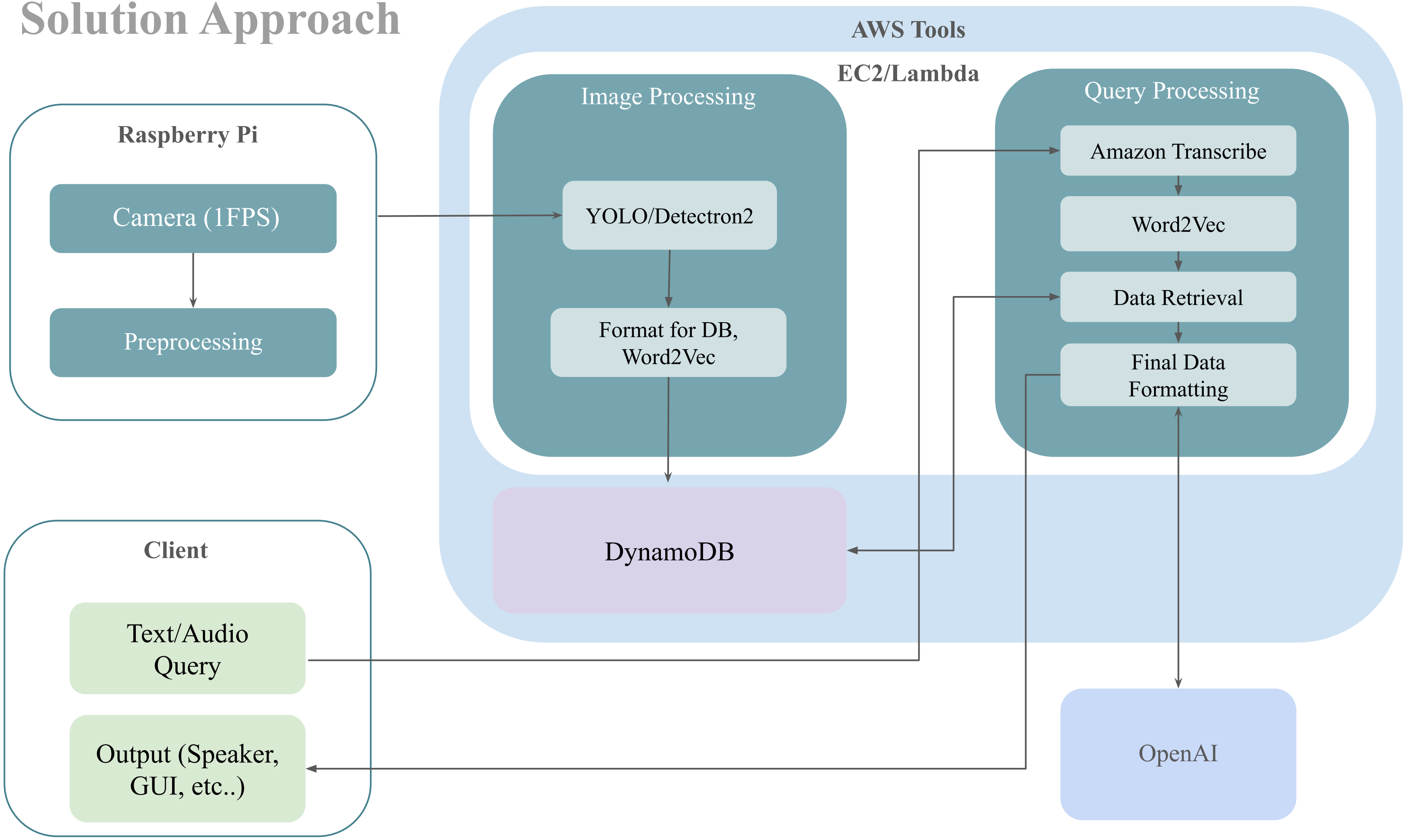
Backup options

If no solution within current system's constraints, have options that are faster/ more accurate

Solution Approach



Solution Approach



Raspberry Pi

Camera (1FPS)

Preprocessing

AWS Tools

EC2/Lambda

Image Processing

YOLO/Detectron2

Format for DB, Word2Vec

Query Processing

Amazon Transcribe

Word2Vec

Data Retrieval

Final Data Formatting

DynamoDB

Client

Text/Audio Query

Output (Speaker, GUI, etc..)

OpenAI

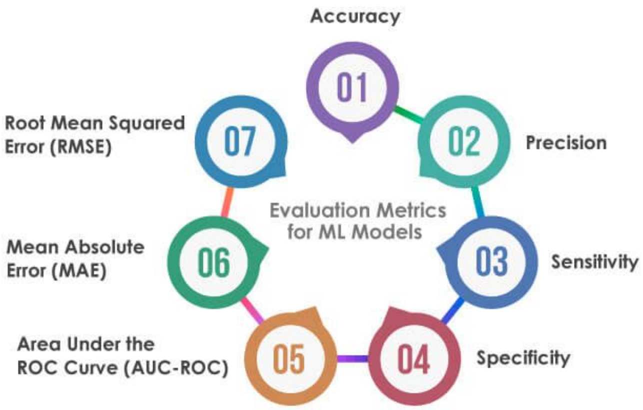
Testing Verification and Metrics

1. Accuracy Testing

- Testing data with different rooms/lighting
- 80% accuracy for objects larger than an iPhone.
- Track up to 10 objects within the predefined space.

2. Latency Testing

- Measure time to process object detection on the edge (<5 minutes per image).
- Validate that users receive location queries within 30 seconds of request and that the photo being analyzed is at most 30 seconds old.
- Monitor edge processing efficiency to ensure it can handle at least 4 data points/sec.


Accuracy
01
Precision
02
Sensitivity
03
Specificity
04
Area Under the ROC Curve (AUC-ROC)
05
Mean Absolute Error (MAE)
06
Root Mean Squared Error (RMSE)
07
Evaluation Metrics for ML Models

https://www.testingxperts.com/blog/ml-testing

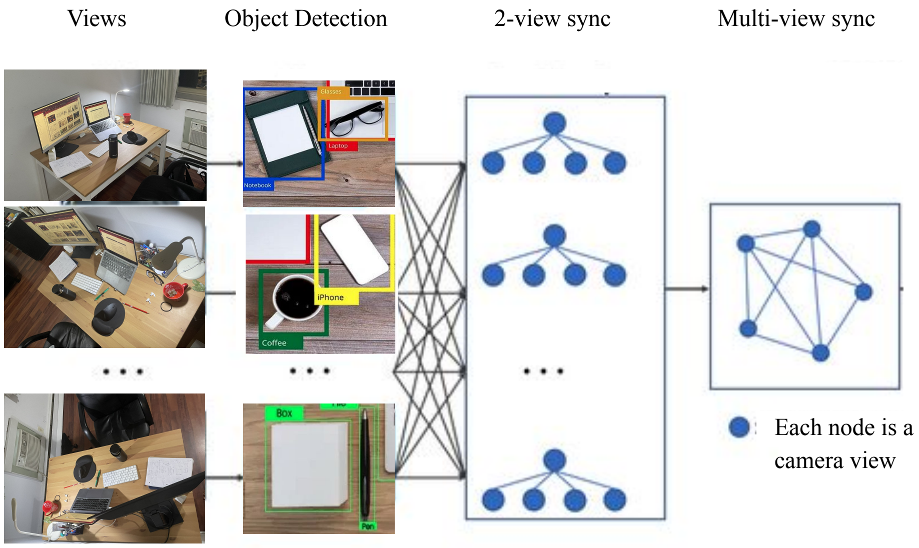
Testing Verification and Metrics

3. User Satisfaction Testing

- Conduct usability tests with target users to assess ease of use, response time, and overall satisfaction.
- Measure user feedback to evaluate whether the product meets customer needs.

4. Multiple Camera Synchronization

- Test time-based synchronization between multiple cameras to verify continuous tracking with minimal blind spots



Views

Object Detection

2-view sync

Multi-view sync

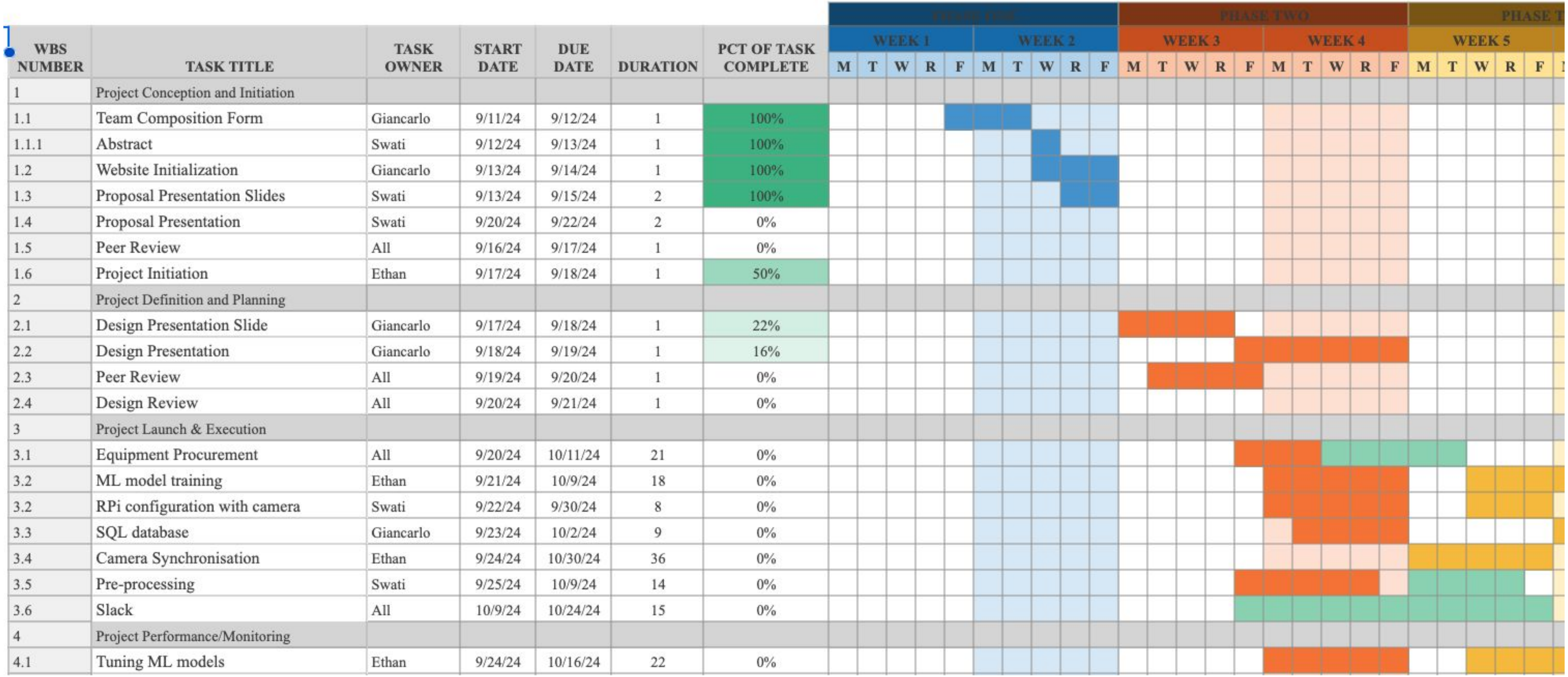
● Each node is a camera view

https://medium.com/analytics-vidhya/beginners-guide-to-object-detection-algorithms-6620fb31c375
https://www.mdpi.com/1424-8220/21/7/2464
https://medium.com/analytics-vidhya/beginners-guide-to-object-detection-algorithms-6620fb31c375

Task Division

Giancarlo	Ethan	Swati	All
Set up cloud infrastructure (AWS EC2, S3)	Develop and train object detection models (YOLO, Detectron)	Set up Raspberry Pi and integrate with cameras	Testing & verification (latency, accuracy, power, synchronization)
Develop and implement REST API for communication between cameras, cloud, and user interface	Optimize model performance for accuracy and speed	Synchronize multiple camera feeds and handle data transmission	Usability testing and gathering feedback
Design and develop web-based user interface for querying object locations	Evaluate model performance in different lighting and room setups	Test and optimize power consumption for portable use	Final system integration and debugging

Gantt Chart



WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE	PHASE ONE										PHASE TWO										PHASE T				
							WEEK 1					WEEK 2					WEEK 3					WEEK 4					WEEK 5				
							M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F
1	Project Conception and Initiation																														
1.1	Team Composition Form	Giancarlo	9/11/24	9/12/24	1	100%					🟦	🟦	🟦																		
1.1.1	Abstract	Swati	9/12/24	9/13/24	1	100%								🟦																	
1.2	Website Initialization	Giancarlo	9/13/24	9/14/24	1	100%								🟦	🟦	🟦															
1.3	Proposal Presentation Slides	Swati	9/13/24	9/15/24	2	100%									🟦	🟦															
1.4	Proposal Presentation	Swati	9/20/24	9/22/24	2	0%																									
1.5	Peer Review	All	9/16/24	9/17/24	1	0%																									
1.6	Project Initiation	Ethan	9/17/24	9/18/24	1	50%																									
2	Project Definition and Planning																														
2.1	Design Presentation Slide	Giancarlo	9/17/24	9/18/24	1	22%											🟧	🟧	🟧	🟧											
2.2	Design Presentation	Giancarlo	9/18/24	9/19/24	1	16%															🟧	🟧	🟧	🟧	🟧	🟧					
2.3	Peer Review	All	9/19/24	9/20/24	1	0%												🟧	🟧	🟧	🟧										
2.4	Design Review	All	9/20/24	9/21/24	1	0%																									
3	Project Launch & Execution																														
3.1	Equipment Procurement	All	9/20/24	10/11/24	21	0%															🟧	🟧	🟧	🟩	🟩	🟩	🟩	🟩			
3.2	ML model training	Ethan	9/21/24	10/9/24	18	0%																🟧	🟧	🟧	🟧	🟧			🟨	🟨	🟨
3.2	RPi configuration with camera	Swati	9/22/24	9/30/24	8	0%																🟧	🟧	🟧	🟧	🟧			🟨	🟨	🟨
3.3	SQL database	Giancarlo	9/23/24	10/2/24	9	0%																	🟧	🟧	🟧	🟧					
3.4	Camera Synchronisation	Ethan	9/24/24	10/30/24	36	0%																					🟨	🟨	🟨	🟨	🟨
3.5	Pre-processing	Swati	9/25/24	10/9/24	14	0%															🟧	🟧	🟧	🟧	🟧		🟩	🟩	🟩	🟩	
3.6	Slack	All	10/9/24	10/24/24	15	0%															🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
4	Project Performance/Monitoring																														
4.1	Tuning ML models	Ethan	9/24/24	10/16/24	22	0%																🟧	🟧	🟧	🟧	🟧			🟨	🟨	🟨