Forget-Me-Not

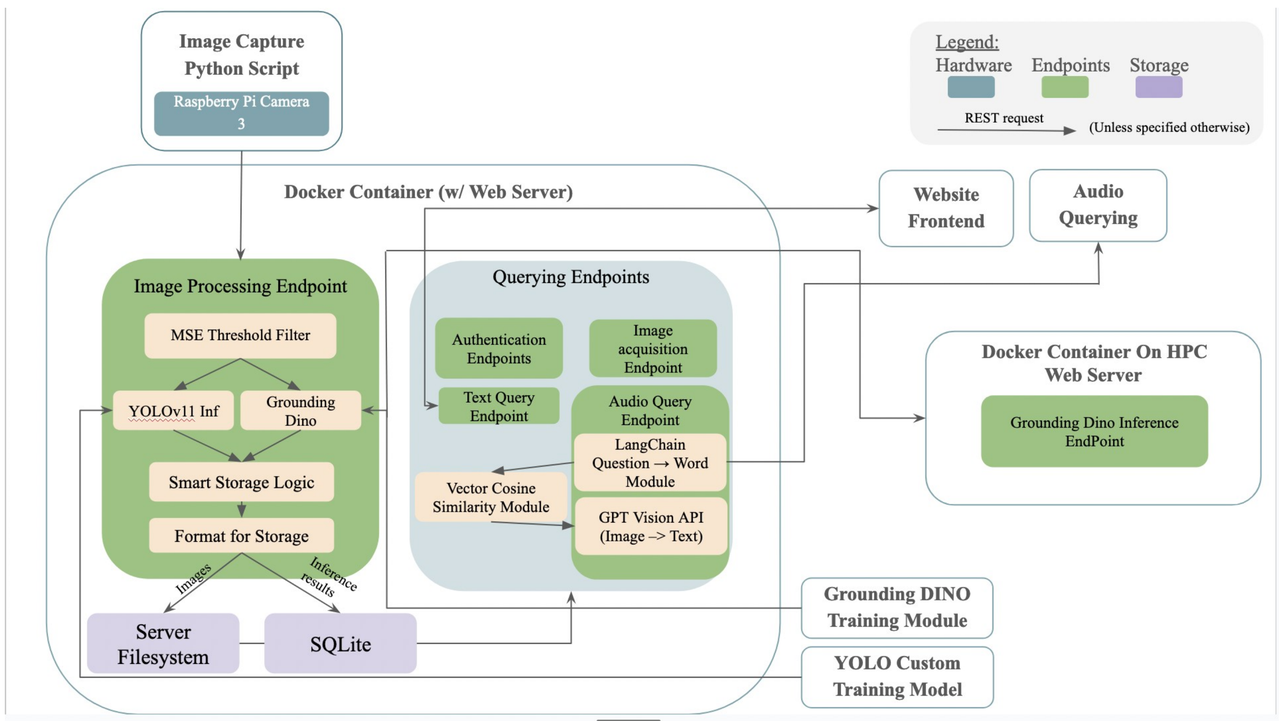
A7: Giancarlo Zaniolo, Ethan Muchnik, Swati Anshu
18-500 Capstone Design, Spring 2024
Electrical and Computer Engineering Department
Carnegie Mellon University

Product Pitch

Forget-Me-Not seeks to help users track and find commonly misplaced items indoors. It accomplishes this using a Raspberry Pi with an attached camera to periodically record pictures of a room, run an ML object detection model on the image, and store the results such that they can be searched using a Web App, or a microphone voice assistant. We sought to achieve 80% object detection accuracy within a 10x10 ft room under well-lit conditions. Our testing showed that we achieved 75% accuracy onboard the Raspberry Pi (YOLO) and an **83% accuracy when running a larger model on the cloud (Grounding DINO) for predefined objects**. We were successfully able to add new objects to the model as well.

System Architecture

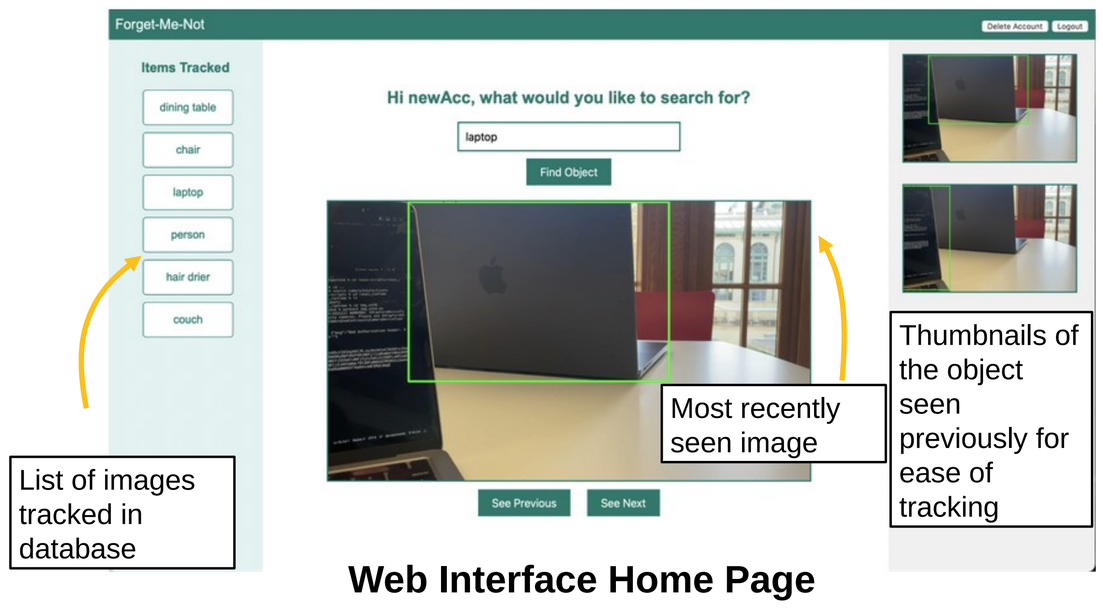
Our high level system architecture utilizes a Raspberry Pi 5 and a camera for real-time image capture. A python script running on the Pi sends images to a Docker container containing processing endpoints. Images are processed with an ML object detection model (YOLOv11 or GroundingDino). and an MSE-based threshold filter to filter duplicate images. To prevent the Pi from running out of storage, by using a heuristic to remove the “least useful image” from the SQLite database once it hits a memory threshold. Our webserver requires authentication for secure access. Users can interact through a web interface or audio commands. Our audio processing uses several processing steps (LangChain, GPT Vision etc.) to return a response with relevant relational information.



System Description

Hardware: A Raspberry Pi 5, a Picam 3W Camera, and a microphone for voice queries.

Software: Our main webserver runs using Flask and Gunicorn, and exists in a Docker container for portability. The image processing pipeline uses YOLOv11 or Grounding DINO for object detection. Users have the option to run YOLOv11 locally, or GPU-accelerated GroundingDino on a secondary webserver, built using the same tech stack as our first webserver, but running on a more powerful machine. We use numpy to accelerate MSE-based image filtering and heuristic calculation for our Smart Storage heuristic. We use SQLite as our database. The web interface is built using HTML5, CSS3, and JavaScript.



Web Interface Home Page

System Evaluation

Use-Case Requirements:

Workflow Elements	Metrics	Meets Target?
Camera Preprocessing Latency	0.0253s	Yes
Speech-To-Text Latency	0.193s	Yes
Web Server “Ping” Latency	8.00*10^-7 s	Yes
Object Detection (ML) Latency	3.35s	Yes
Database Write Latency	0.12s	Yes
Database Read Latency:	0.00254s	Yes
Other ML (Langchain) Latency:	5.1945s	Yes
Total Cross-Component Latency (Q)	5.34s	Yes
Total Cross-Component Latency (M)	3.54s	Yes

Query and Monitor Workflow Metrics

	MAP-50	MAP-50-95	Beats Target
YOLOModelCfg1	0.6682695/1	0.5301794	No
YOLOModelCfg2	0.7478448/1	0.5984732	No
Grounding DINO	0.8330/1	0.6422313	Yes

Model Accuracy

Approach to Testing:
Unit testing: Image Processing pipeline, storage logic, database reads/ writes etc.
Integration Testing: Frontend, backend and database data flow and query handling
Performance Evaluation: Metrics like inference time, query response time, model accuracy
Validation against set targets: Cross checked against design goals (seen in the table)

Conclusions & Additional Information

Please visit our blog site for further information

Our system successfully implements core functionalities like real-time image processing, voice-based querying, and efficient storage, which aligns closely closely with our aspirations.
The project has potential for broader applications in security, elder care, and home automation. Future development could focus on enhancing model training with bigger and diverse datasets, scaling the system using cloud infrastructure, and refining user interfaces. As a team, we learned the importance of clear task delegation, iterative testing, playing to individual strengths and adapting to technical challenges along the way.

Electrical & Computer ENGINEERING
Carnegie Mellon