PEOPLE COUNTER

Team CO: David Feng, Brian Li, Gary Qin

COUNT, ESTIMATE, and PREDICT the number of people in an enclosed space and provide REAL-TIME feedback to users

Use Cases

- Lecture Halls & Conference Rooms

- Public Transportation

Use Case Requirements

- Up to **90%** estimation accuracy
- Info updated to user every **1 minute**
- Ability to predict with **categorization**: “almost empty”, “not busy”, “busy”, “almost full”
- If deployed to cloud, OpenCV’s FPS >= **20fps**
- Ability to connect two or more cameras in case a space has multiple entrances/exits

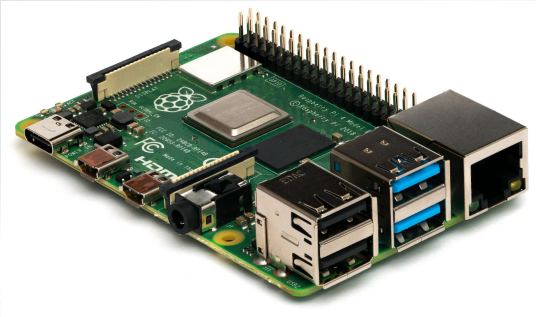
Technical Challenges

- Methods to reduce latency so that estimations are updated as often as possible
- High-accuracy computer vision and computation
- Distinguishing between humans and other moving objects

- Robust prediction algorithm that takes into account unpredictable movements
- Gathering enough data to train model for prediction
- Integrating data of multiple camera/sensor feeds and compute them as one

Solution Approach

- **Hardware:**
- Camera connected to a Raspberry Pi:
  - Send video to backend

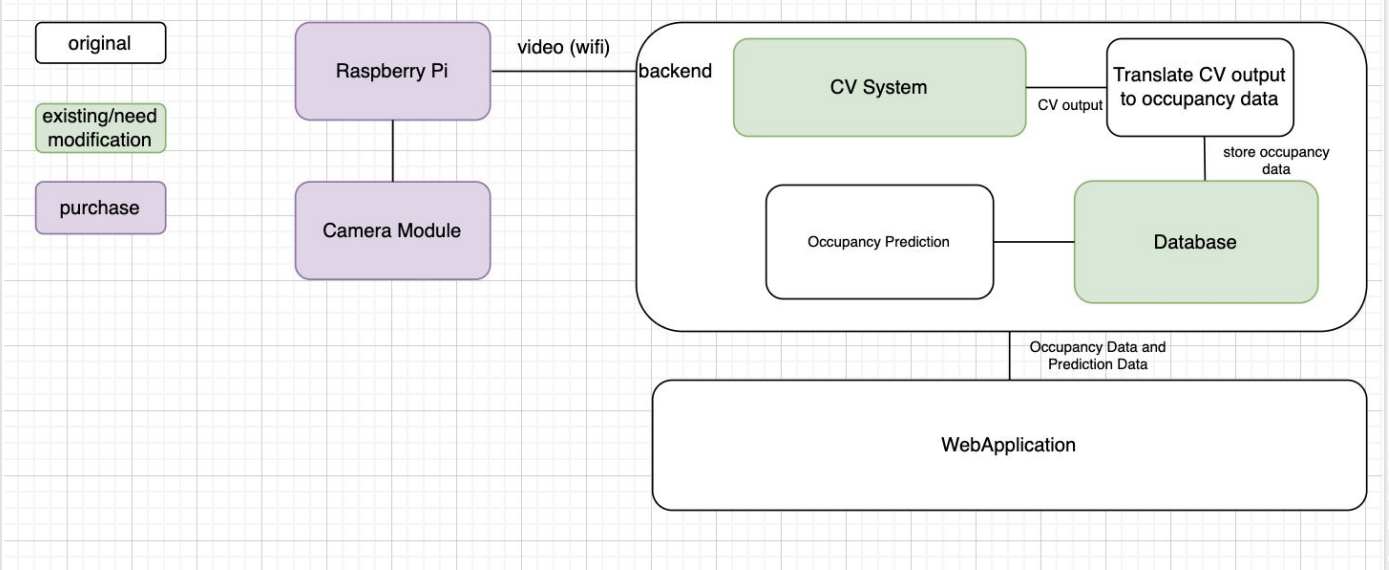


Solution Approach

- **Backend:**
- Computer Vision:
  - Object Detection and Tracking
  - Low Processing Time
- Process CV output
- Store occupancy data in Database
- Occupancy Prediction
- Send Data to Web App

Solution Approach

- **Web App**
  - Display occupancy data
  - Prediction feature



Testing & Verification

- **Stage 1** - Testing in a room with only one clear doorway
- **Stage 2** - Testing in a room with multiple doorways
- **Stage 3** - Testing in an open environment with broad entrances and exits

Metrics & MVP

Quantitative metrics to measure

- Accuracy
  - 90% estimation baseline
  - Over at least 10 hours
- Latency
  - 60s glass-to-glass
    - 20s camera module -> backend
      - Video @ 640x480, 20 fps
    - 40s backend -> user interface
      - Includes CV algorithm run time, updating information in database, change estimation and prediction on UI

Division of Work

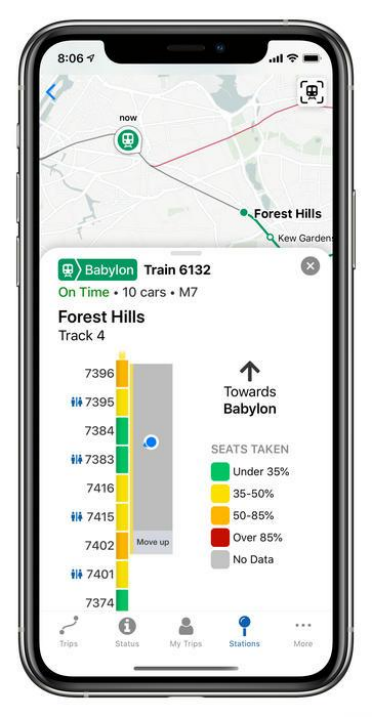
David	Gary	Brian
- Configuring camera module with Raspberry Pi - Calibrating camera configuration to testing environment - Connecting video feed to CV backend over Wifi	- Web App: connecting backend data with server side with Django, HTML, Javascript, and CSS - Sanitize data to prevent web-based security attacks - CV testing	- Computer Vision system to detect and track people - Translate CV output to occupancy data - Prediction based on historical data

Schedule

Category	Task	Week 4 2/6-2/12	Week 5 2/13-2/19	Week 6 2/20-2/26	Week 7 2/27-3/5	Week 8 Spring Break	Week 9 3/13-3/19	Week 10 3/20-3/26	Week 11 3/27-4/2	Week 12 4/3-4/9 Interim Demo	Week 13 4/10-4/16	Week 14 4/17-4/23	Week 15 4/24-4/30
Design	*Aquire components*												
	Design Review												
	Research camera libraries												
	Research CV Libraries /Deployment methods												
Hardware	*Access camera from Pi*												
	Calibrate camera to test environment												
	Connect video feed from camera to Pi												
	Connect sample video from Pi to backend												
	Assemble full video pipeline												
Backend	*Implement People Detection & Tracking*												
	Processing Time Benchmark												
	Convert CV Output to Occupancy Data												
	Occupancy Predicion Algorithm												
	Error Check / Refine												
Web	*Create local app with HTML and CSS placeholders*												
	Integrate backend data to web and test locally												
	Create Django application that runs on server-side												
	Data accuracy, latency, and web security testing												
Integration	*Video Feed & Backend*												
	Backend & WebApp												
	Full integration												
Deliverables	*Proposal Presentation*												
	Design Review												
	Design Review Presentation												
	Design Review Report												
	Ethics Assignment												
	Interim Demo												
	Final Presentation Slides												
	Final Presentation												
	Final Report												
Misc	*Slack*												

David
Gary
Brian
All

Ethical Challenges



- **Privacy** issues
- **Legal** challenges for testing and deployment in public areas (e.g. buses and trains)
- **Algorithmic** bias (e.g. racial bias, gender bias, etc.)