PetSTAR

B7: Rebecca Manley, Brandon Wei
18-500 Capstone Design, Spring 2023
Electrical and Computer Engineering Department
Carnegie Mellon University

Product Pitch

We set out to create a pet monitoring system with features beyond currently available camera systems: keeping track of pet activity and notifying the user if a pet tries to enter somewhere it shouldn't. To this end, we needed our system to track pet movements accurately - estimate each animal's position within 1 foot - and respond to any events quickly - forbidden location detection within 1 second, and a notification delivered in less than 10 seconds.

Our final product utilizes a Raspberry Pi and web application to meet these requirements. We have achieved an average accuracy within **3-6 inches**, and an average speed of **0.6 seconds** to detect and **1.1 seconds** to report.

System Architecture

Two major subsystems: the web app and Raspberry Pi. These systems communicate relevant pet data back and forth via requests. The web application consists of a frontend (user interaction) and backend (data storage). The Raspberry Pi hosts the computer vision tasks to process video feed and report to the web app.

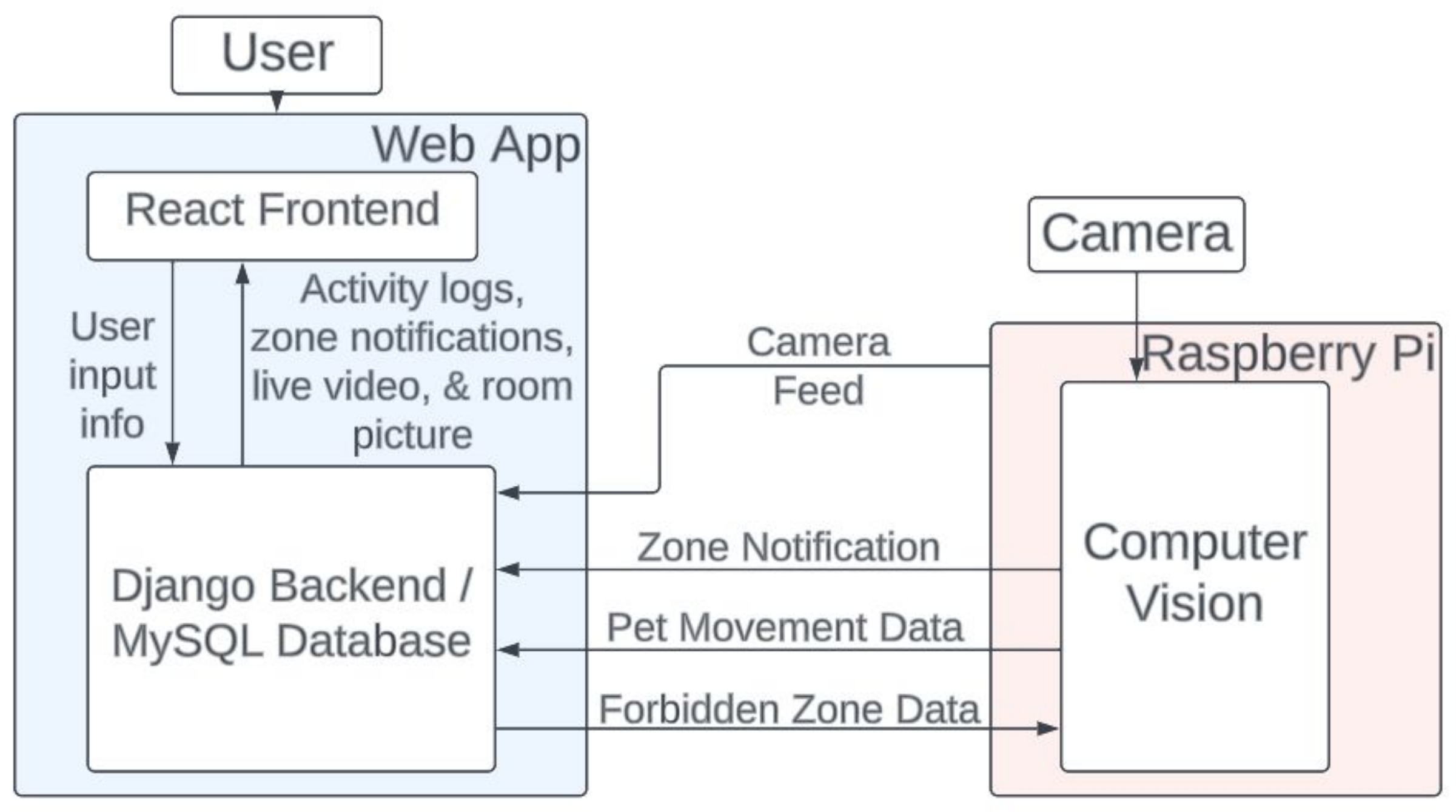

Figure 1. Overall Block Diagram

User experience of our web app is expressed with a flow diagram. First time users must add a room picture, which is used as a reference for the other tasks. Then, the user can engage in four major tasks that help them to monitor their pet(s): specifying forbidden zones, viewing pet activity logs, watching live video feed, and inputting new pet information.

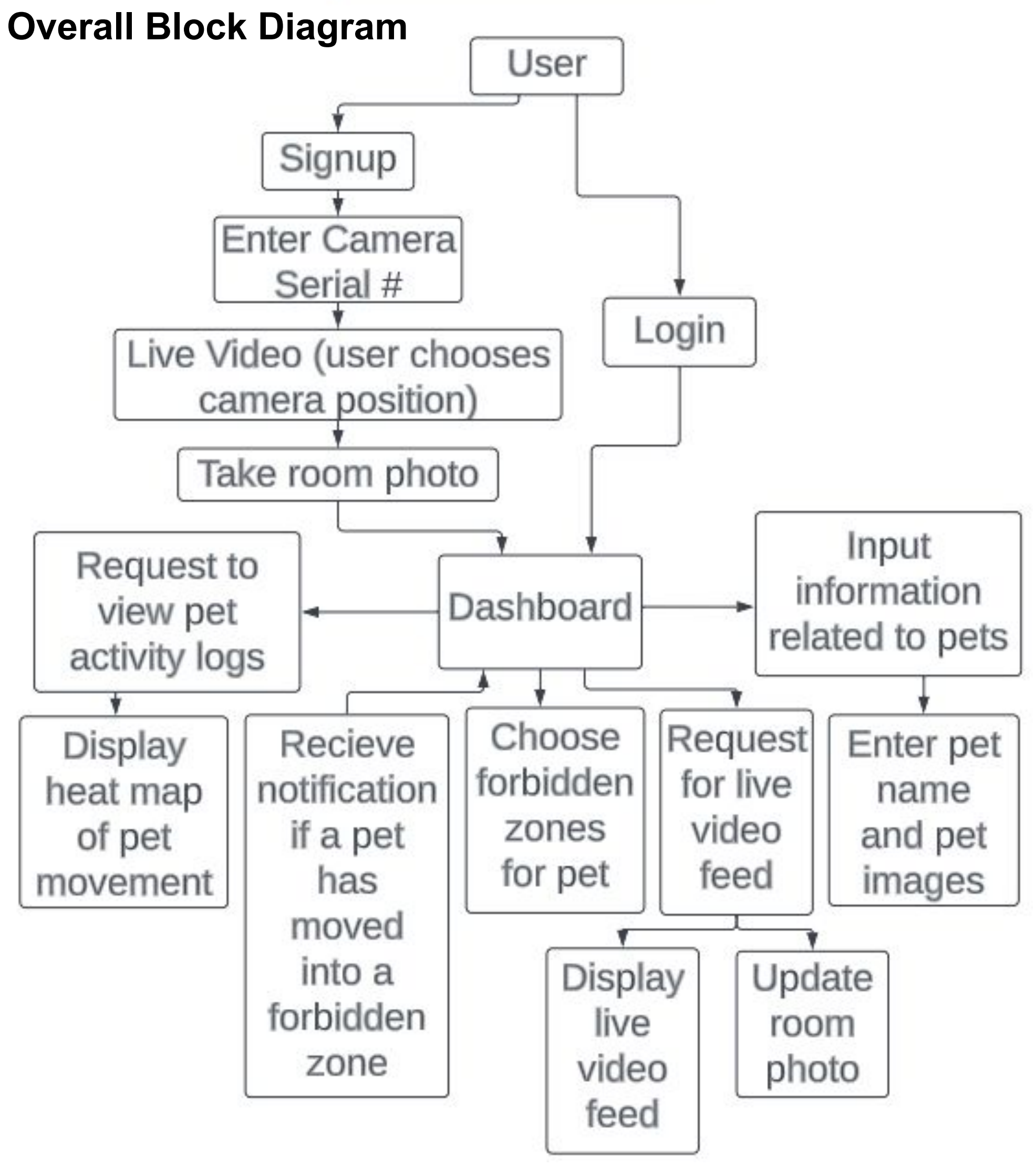
User
Signup
Enter Camera Serial #
Live Video (user chooses camera position)
Take room photo
Login
Dashboard
Request to view pet activity logs
Input information related to pets
Display heat map of pet movement
Recieve notification if a pet has moved into a forbidden zone
Choose forbidden zones for pet
Request for live video feed
Enter pet name and pet images
Display live video feed
Update room photo

Figure 2. Software User Flow

Conclusions & Additional Information

Our project initially aimed to incorporate an aspect of machine learning - to differentiate between multiple animals in the home - but needed to be rescoped in the middle due to unavoidable personal setbacks. Relative to the new definition, we feel that we accomplished what we set out to do. Given more time though, we would have liked to execute this additional aspect of ML and refine the major systems in our project.

Through this project, we learned how to make the best of a difficult situation when challenges arise. We also learned the importance of the design process and team communication in a successful project. Future possibilities for this project might be to include the ML or to incorporate more sophisticated vision algorithms to estimate position in 3D for greater accuracy.

http://course.ece.cmu.edu/~ece500/projects/s23-teamb7/

System Description

The web application uses React for the frontend and Django for the backend. Communication between these parts is done using Axios to handle requests and Django REST Framework to format data for the backend. Two unique features of the web application are specifying forbidden zones where pets should not go and viewing pet activity via a heat map.

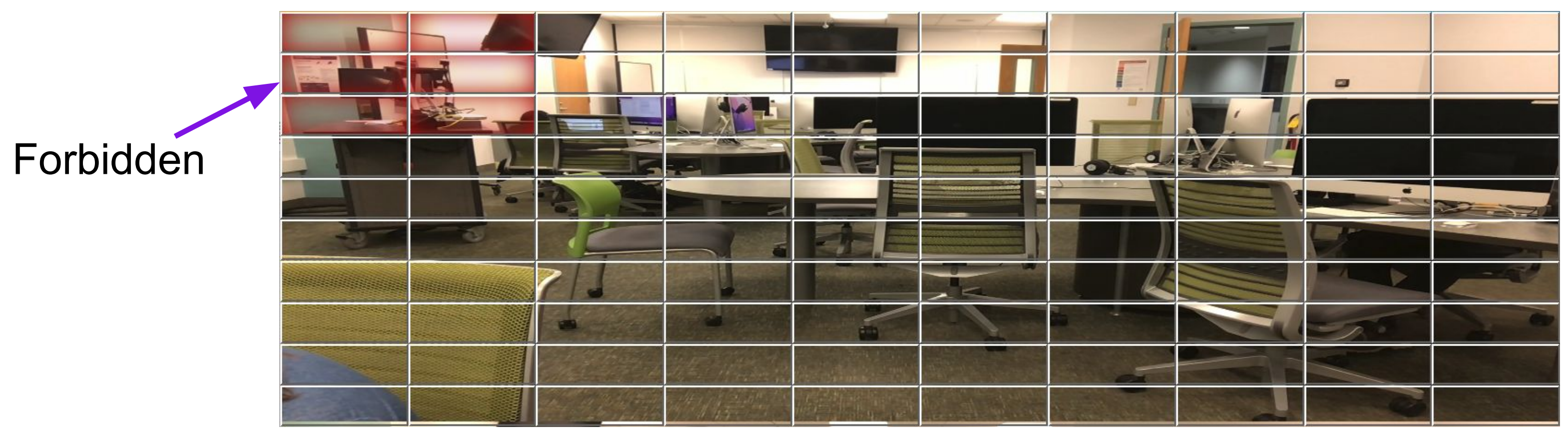

Figure 3. Forbidden Zone

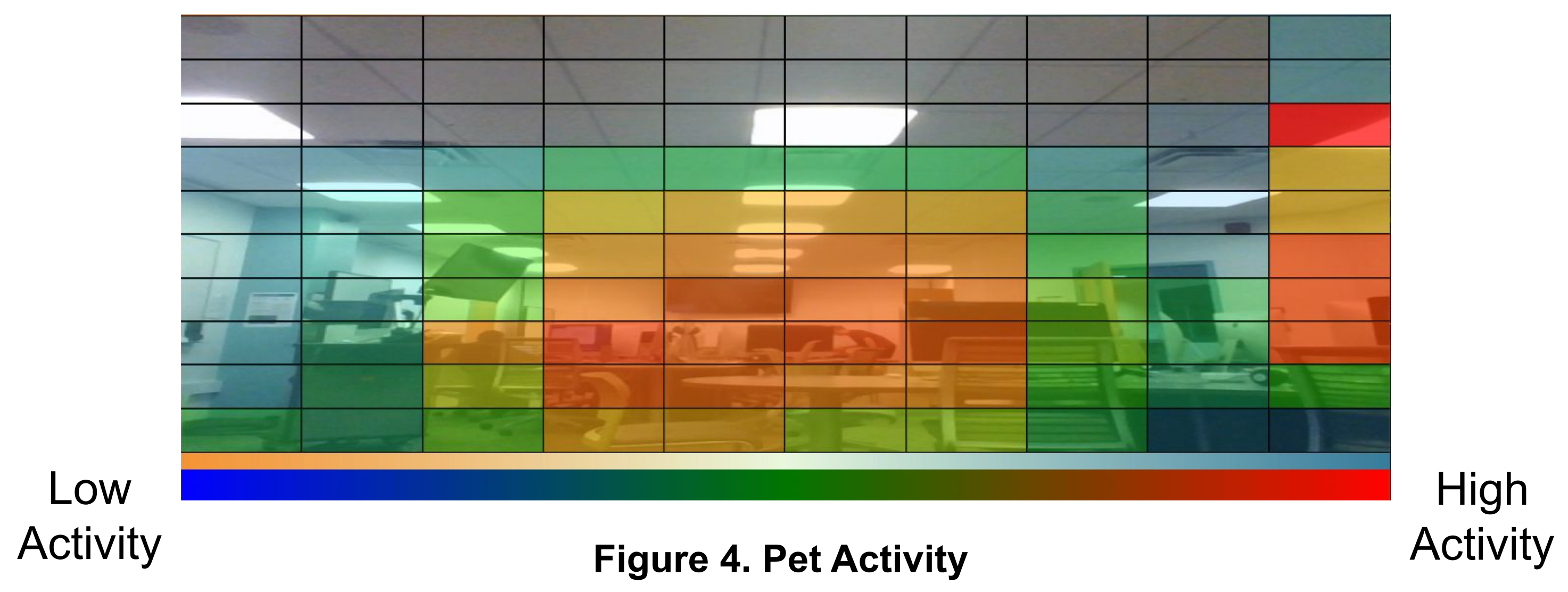

Figure 4. Pet Activity

The hardware for our project consists of a Raspberry Pi Model 4B connected to an RPi camera module 3. The RPi runs computer vision algorithms implemented with OpenCV. It detects new movement via pixel differences and tracks identified movements using OpenCV built-in tracking algorithms.

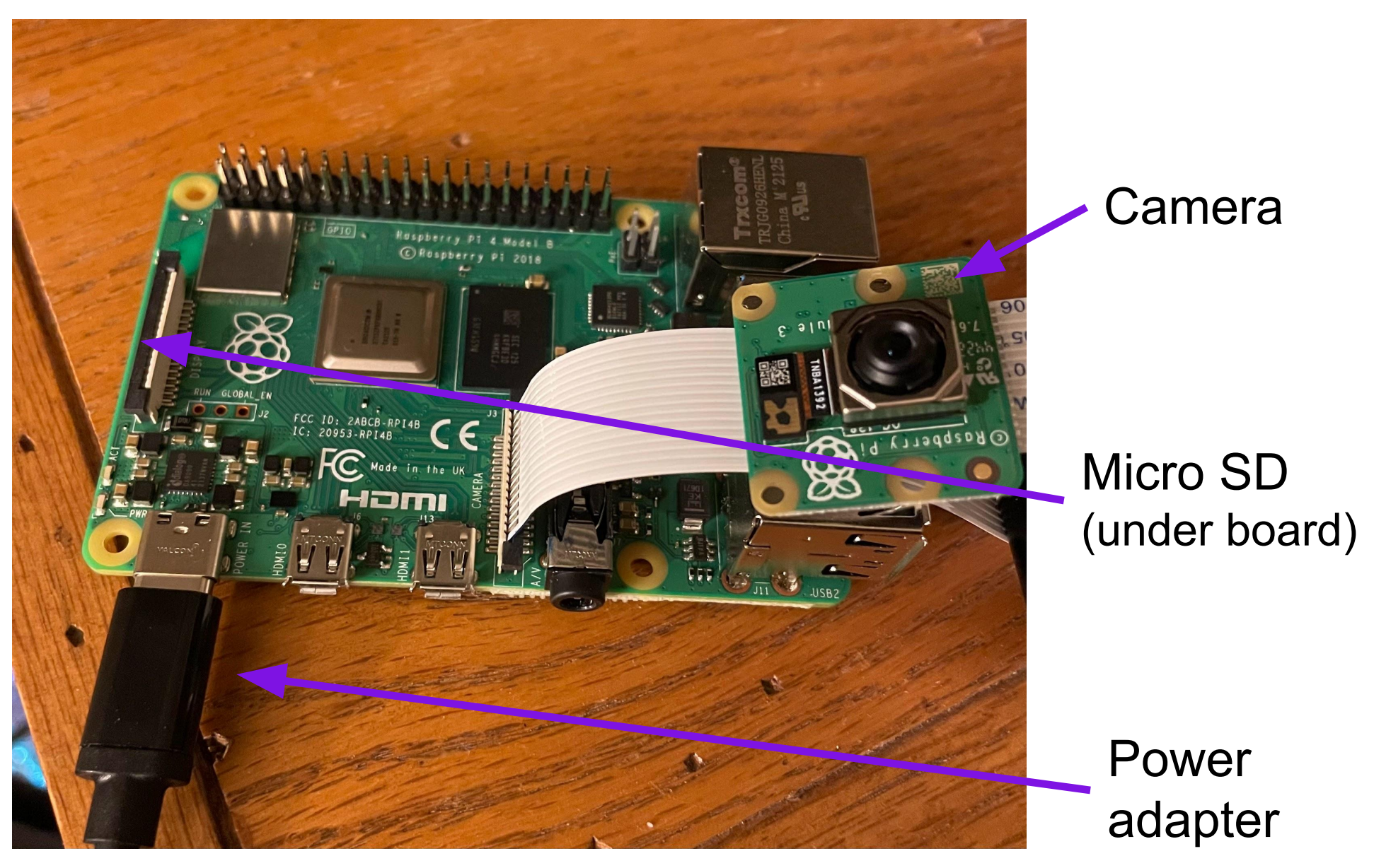

Figure 5. Raspberry Pi Configuration

System Evaluation

The approach to any speed-related tests was to use a slow motion camera to more precisely calculate the time between two visual events. Accuracy was measured by comparing human chosen position to computer chosen position.

Test	Method	Metric	Goal	Result*
Zone detection speed	Slow-Mo Video	Pet enters zone in real life -> CV detection	< 1 second	~0.625 seconds
Zone notification speed	Slow-Mo Video	Pet enters zone in real life -> Web app notification	< 10 seconds	~1.125 seconds
Accuracy of tracking	Human vs. Computer	Distance between centers of human-chosen vs. computer-chosen bounding boxes	Within 1 foot	Generally 3-6 inches
Detection speed of new animals	Slow-Mo Video	Animal enters frame in real life -> bounding box appears	< 5 seconds	~0.75 seconds

The biggest tradeoff we faced for speed was the frequency of requests between the CV, backend, and frontend. Faster request rates means info on the web app is more up to date (including notifications), but the extra computational power/time uses more web server resources and lowers our frame rate.

*Numbers listed above were computed running the CV code on a laptop.

Electrical & Computer ENGINEERING
Carnegie Mellon