# OccuPi: Dynamic Occupancy Tracking System

Timothy Leveille, Cody Cheng, Oliver Pender, Jae Butler

*Innovation Scholars 2023 - 24, University of New Hampshire, Durham, NH 03824*

## Introduction

**Goal:** Develop a cost-effective camera system to monitor occupancy and movement in the UNH CEPS Makerspace at Kingsbury Hall.

**Motivations:** High costs of existing foot-traffic monitoring technologies.

**Objectives:**

1. Accurately detect people in the space.
2. Stream real-time data to a website.
3. Generate and display a heatmap on the website.

## Challenges

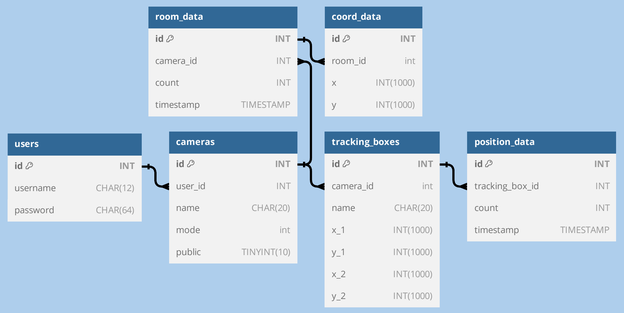
**Tracking method:** We initially used blob tracking for people detection but found it inaccurate and inefficient due to contrast issues. We switched to Ultralytic's pre-trained AI, YOLOv8, which proved far more effective.

**Data Management:** An erroneous command led to the loss of a day's data collection. This underscored the importance of meticulously managing server permissions.

## Database Structure

Database schema overview:

- **'users'**: Stores user data. Each user can own multiple cameras.
- **'cameras'**: Stores camera data. Each camera can have multiple tracking boxes
- **'tracking_boxes'**: Defines areas of interest in the camera's view.
- **'position_data'**: Stores person count within each tracking box over time.
- **'room_data'**: Stores overall person count for each camera over time.
- **'coord_data'**: Stores specific coordinates of detected people.




## Implementation

**Hardware:**

We use a Raspberry Pi Zero 2W and a 160° field of vision camera module to gather room occupancy data. The camera module is housed in a custom-designed 3D-printed case as shown to the right.

**Software:**

Using a Python script running on the Raspberry Pi, still images are sent to our Flask-SocketIO server every ten seconds. There, the OpenCV library processes them and the YOLOv8 AI model detects people and their positions. The detected data is then stored in an SQLite database. We use this data to generate weekly activity reports and heatmaps, which provide user insights. This information is displayed on our webpage for user access.

**Deployment:**

We deployed our Flask-SocketIO server on Amazon's EC2 service, used Nginx as a reverse proxy, and Gunicorn to serve the web pages efficiently. We used these because they are widely used and considered very effective. This allows our product to be scaled easily.

## Process Diagram


## Heatmap Analysis

A clear visualization was obtained from the positional data captured by the camera modules. Utilizing the YOLOv8 AI model, human presence in the frames was detected, and their (x, y) positional coordinates were recorded. Subsequently, these coordinates were processed using Python and Matplotlib, generating the figures presented below. Figure 1 illustrates the two primary workbenches located within the Makerspace, surrounded by significant patterns of activity. Figure 2 shows a lower density of activity. Nevertheless, the predominant traffic in Figure 2 is observed near the computer terminal designated for 3D model slicing in preparation for printing.

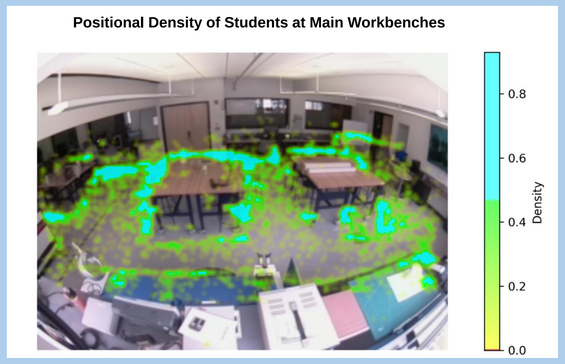



Figure 1.

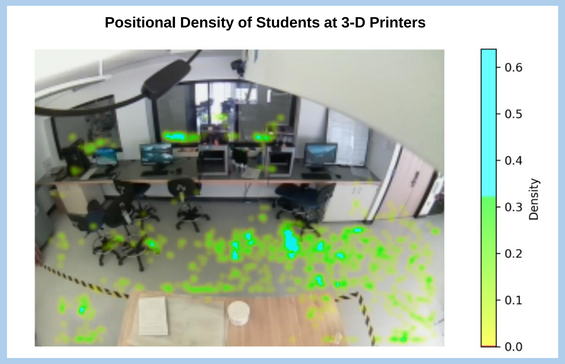
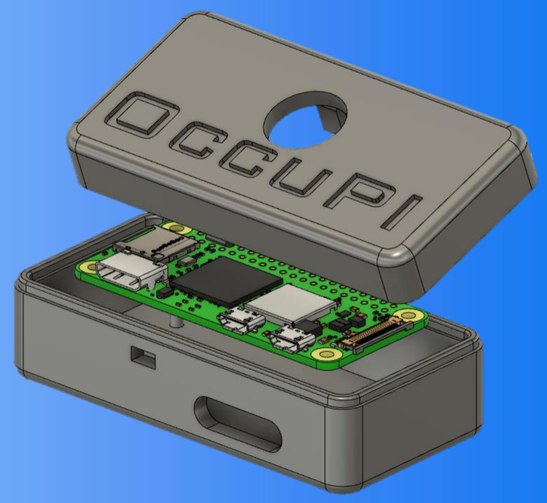



Figure 2.

## OccuPi Camera Module




## Future Improvements

**Future Improvements:**

- Moving the processing over to the Raspberry Pi with a Coral Edge TPU USB Accelerator would increase security by keeping the image frames on the Raspberry Pi and dissolve the need to increase server capabilities with more OccuPi modules.
- Design our own custom PCB to replace the Raspberry Pi, giving us greater control in component selection to reduce cost and increase functionality.
- Migrate from SQLite to PostgreSQL which is a more scalable database and would leverage its ability to handle higher levels of traffic.
- Create an even more modular design with individual batteries in each module allowing for simpler deployment and more flexibility for camera placement.

## Acknowledgements

Thank you to the UNH CEPS Makerspace staff and TSC Director Kevan Carpenter for providing us with access to their state-of-the-art facility, allowing us to employ this project in a real-world environment. Thank you to cohort leaders Kyle Ouellette and Dr. Dean Sullivan for their expertise and guidance throughout this project.

## References

- https://docs.ultralytics.com/modes/predict/
- https://flask-socketio.readthedocs.io/en/latest
- https://python-socketio.readthedocs.io/en/stable
- https://picamera.readthedocs.io/en/release-1.13
- https://docs.python.org/3/library/sqlite3.html
- https://docs.opencv.org/4.x/