Grocery Store Checkout System

C3: Brian Chhour, Shubhi Jain, Simon Xu
18-500 Capstone Design, Spring 2024
Electrical and Computer Engineering Department
Carnegie Mellon University

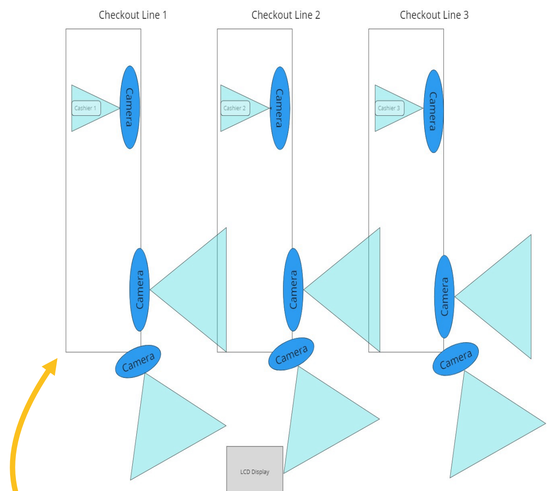
Product Pitch

Ever get annoyed in a grocery store when someone at another checkout lane checks out faster than you? Our system speeds up the process to get checked out by telling users which lane to enter based on some key data. Currently a lot of users either spend a lot of time observing lines and wondering if they entered the fastest line, or they spend a lot of time in the line itself. This system uses several WiFi connected cameras to observe the lines. The use case requirements of our system are to predict the fastest checkout counter and communicate it to the grocery store customers in real time.

It tracks and stores the **cashier throughput, the number of people in line, and the fullness of each cart for all counters in real time**. Based on how fast the cashier is operating at, as well as the data on the line and carts, the system will compare each line to determine the fastest line. The system will then send the result to a raspberry pi to display on a monitor screen, which customers, or our "users" can use to guide themselves to the best line.

System Description

Our system is divided into several parts. Our system input is the camera data from the three cameras on each counter. There are cameras set up around the checkout counter to be able to capture video footage, which will then be uploaded over WiFi using a Raspberry Pi to an S3 bucket. On the computer that runs the system, the camera footage will be downloaded in real time and then processed using 3 separate YOLOv8 models (for carts, people, and hands) in parallel to gather the throughput, number of people in the line, and how full each cart is. This data will be uploaded to a database continuously and retrieved by the system to calculate and rank the counters based on how much faster each counter is from each other. The counter taking the least amount of time will be sent to another Raspberry Pi and displayed on a monitor for customers to see and follow to the fastest counter.

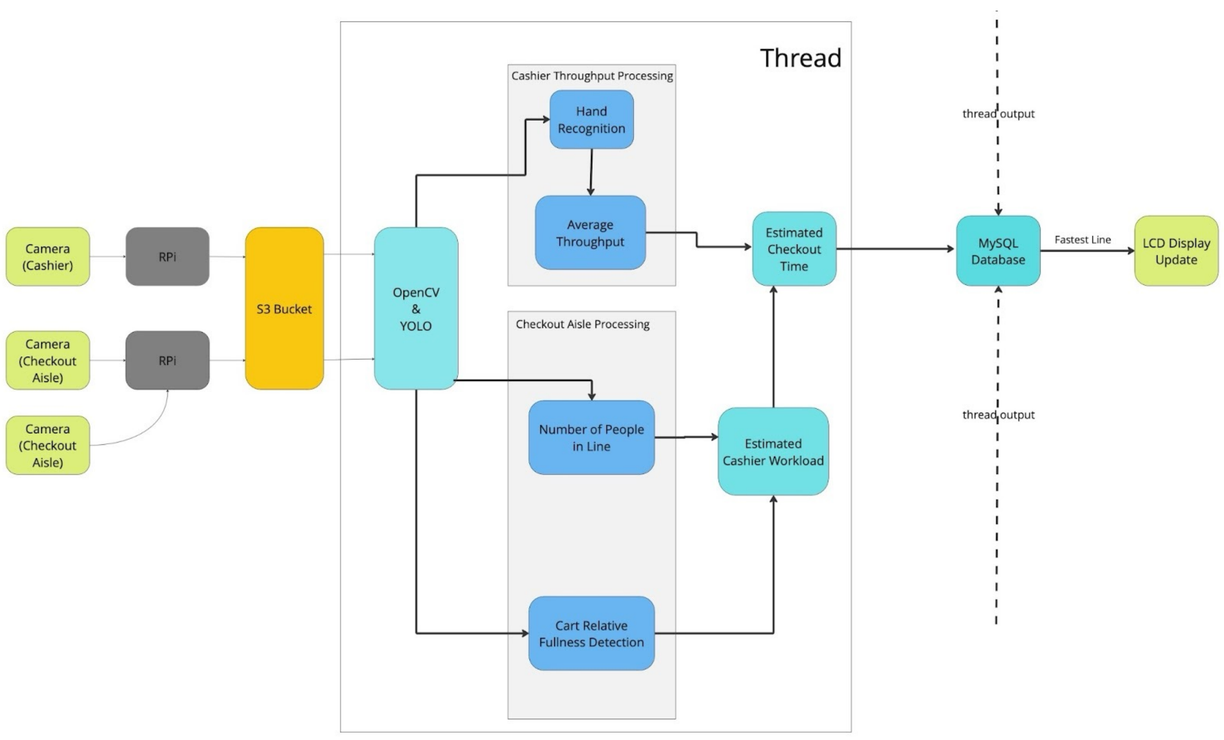


Camera layout at grocery store

Relative Fullness and Line Detection Cameras

Cart detection in relative fullness module

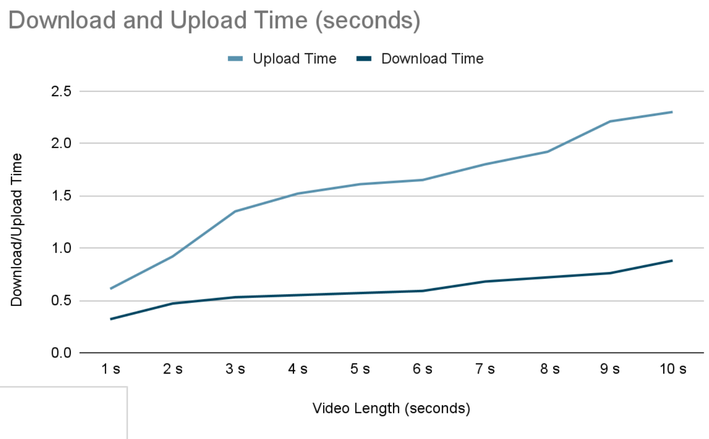
System Architecture

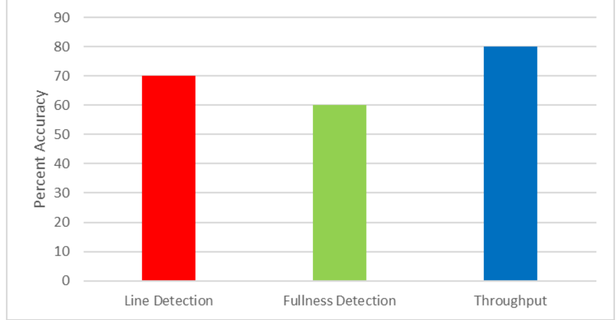


We have one thread running for each counter, and as the database is updated in each thread, the system will also calculate and display the results on a monitor.

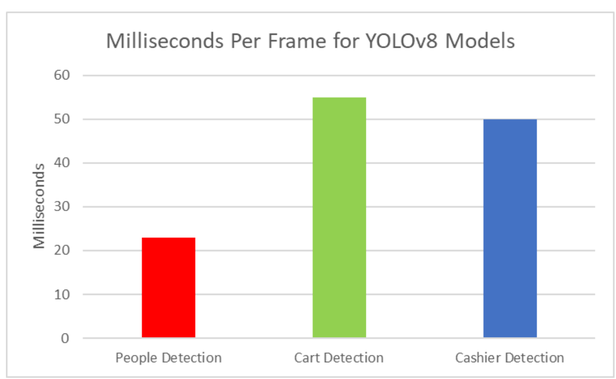
System Evaluation

This graph shows the upload and download time from the S3 bucket with respect to videos of different length. We used this data to determine what length videos would be best for meeting latency requirements, which ended up being 2 second videos.





This bar chart shows the accuracy of our software components as of right now. Line detection is correctly tracking the number of people in line, fullness is how close our estimate is to actual cart fullness, and throughput is how many items we think have been scanned vs. actual.



This bar chart shows how fast our system is able to process video footage frame by frame.

Conclusions & Additional Information

https://course.ece.cmu.edu/~ece500/projects/s24-teamc3/

Our system is an effort to simplify the checkout process at grocery stores for customers in a rush. While we have had to pivot at multiple points to accommodate various technical, physical, and ethical constraints, we have learned a lot from this experience. Our camera layout changing due to people's ethical concerns as well as physical restraints around the counters has been challenging to cater our models and algorithms to. Furthermore, we initially wanted to count the number of items from footage of people's carts, which ended up being nearly impossible due to limitations with object detection and our camera angles. This led us to instead estimate the relative fullness of a cart rather than provide a direct count of the number of items in a cart, and use that for our calculations instead. At the same time, there is room to make the algorithms and logic more robust to account for the large variety of different items in a grocery store: in order to calculate throughput, we tracked the hands of the cashiers, however, counting items on the register would be feasible with the correct camera angle and a better object detection model.

Electrical & Computer ENGINEERING
Carnegie Mellon