No Time To Dine

Team B0: Chi Nguyen, Nii-Okaitey Tetteh, Likhitha Chintareddy
18-500 Capstone Design, Spring 2023
Electrical and Computer Engineering Department
Carnegie Mellon University

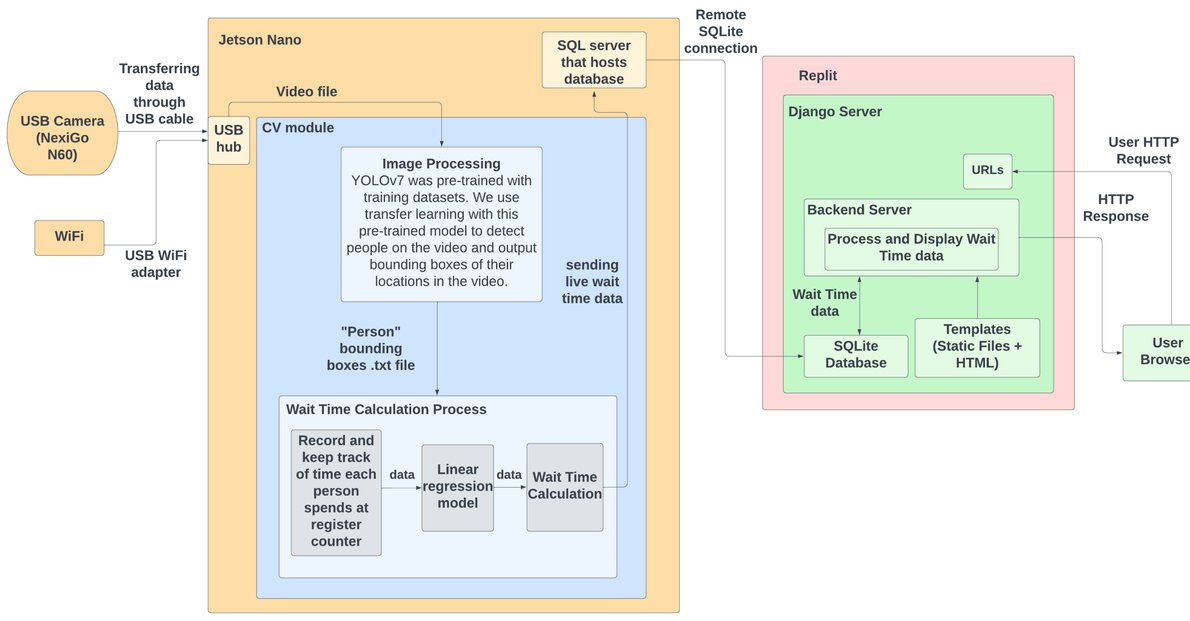
Product Pitch

Due to busy schedules, students may be in a rush to grab food on campus. Everyone tries to leverage wait time into the small eating periods they have and a busy dining hall with long lines may mean being late to class. Our aim is to help students manage their time better by providing an application which provides wait time information to students.

For this system to be useful we needed to engineer a **fast** and **accurate** system, which dynamically updates its information in real time. We were able to create a solution which provides a wait time approximation within a margin of error of at most **1 to 2 minutes**.

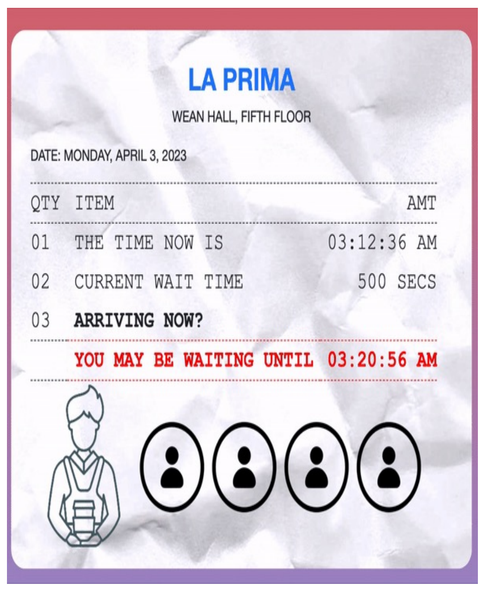
System Architecture

Our system uses a USB camera to capture the live video footage of the line at the dining location of interest. This USB camera is connected to the Jetson Nano through a cable and its USB hub. The Jetson Nano houses CV module where the image processing, which is the YOLOv7 algorithm, and wait time calculation take place. The YOLOv7 algorithm takes in the video footage captured by the camera, detects people in the video and outputs a text file that has the information regarding the bounding boxes of people in each frame. The wait time algorithm then takes in this text file, analyzes the data given in this file and estimates the best current wait time at the dining location. This information is transferred to the SQL server that hosts the database and communicates with the SQLite database on the frontend side through remote SQLite connection. The web application displays data for the users by fetching results stored in the SQLite database.

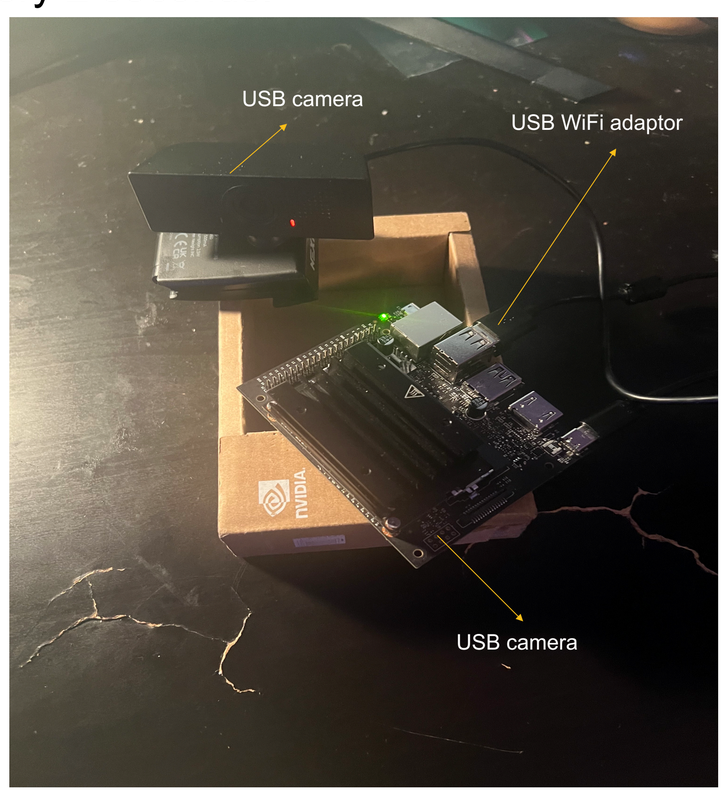


System Description

Half of our system is built on a 2GB Jetson Nano. We connect a USB Webcam to the Jetson and use its live feed to create our wait time prediction. Our data is then sent to a database hosted on the Jetson, and sent over wifi to our front end. The wait time information and other data calculated in the backend is displayed on the web app that is built on the Django web framework. The UI components are styled using Bootstrap CSS, React JS and D3 JS. AJAX implementation retrieves information from server every 2 seconds.



Screenshot of section of webapp displaying wait time and number of people in line



System Evaluation

Testing metrics	Use Case Requirement	Test result
Web application response time	Less than 2 seconds	~ 2 seconds
Margin of Error for Wait Time	1-2 minutes	<1 minute

Miscellaneous testing metrics	Test Results
Speed when running the CV code on Jetson Nano	~11 s/frame
User Survey – 6 participants	6/6 reported received the information they need to make a choice about the dining hall 5/6 said information was easy to read

Design Tradeoffs	Reasoning
Changed database from MongoDB to SQLite	SQLite is more suited for our use case, simple and inexpensive vs MongoDB is too complex for use case and expensive
Switched to USB camera instead of SainSmart camera	USB camera is easier to work with the Jetson Nano

Conclusions & Additional Information

With this project we aspired to make a system which would make a tangible improvement to student life experience. To this end, we were hoping to have a system that were a little more accurate than existing ones. However, we were able to create a good foundation. Future work could easily be continued to better prediction accuracy, only furthering its usefulness.

http://course.ece.cmu.edu/~ece500/projects/s23-teamb0/

Electrical & Computer ENGINEERING
Carnegie Mellon