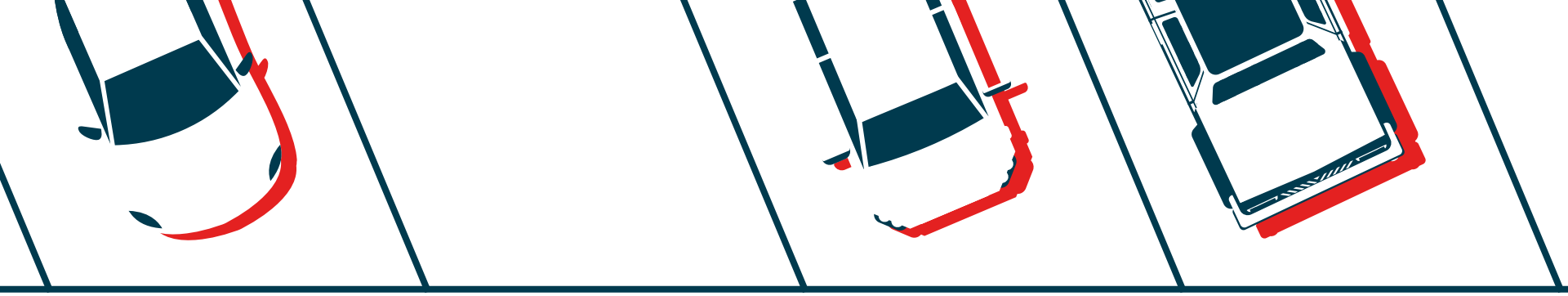
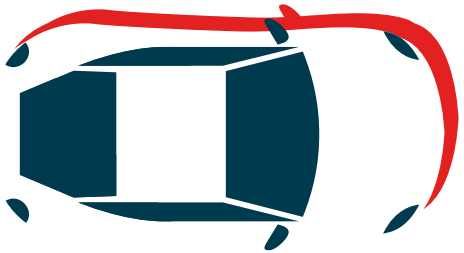


Critical Design Review

Parking Lot Vehicle Classification

Team Members: Aniruddha Srinivasan, Coleman Todd, Fabianna Barbarino, Jacqueline Mioduski, Spencer Cho, Tyler Roosth

Presentation Contents

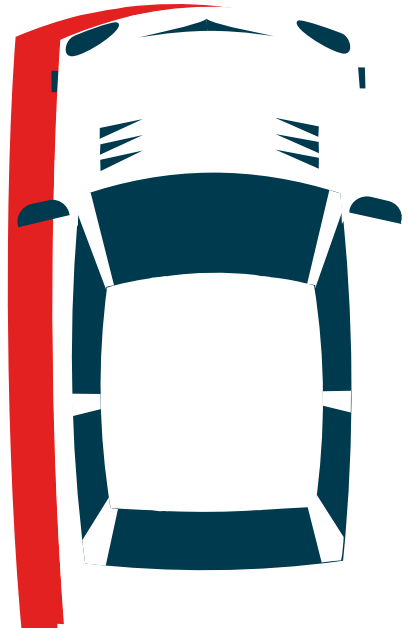


Problem Background

- Increased vehicle traffic as business growth occurs
- Security personnel need to identify a large number of cars simultaneously
- Analysts need to track heavier traffic flow

Requirements

- Take in videos
- Tracking at least 5 vehicles at 10+FPS
- Desktop App
- Run in Docker Container
- Vehicle States
  - Parked
  - Stopped
  - Moving



Goals & Objectives

Our main goal to build a containerized app that detects and tracks vehicles across a parking lot and continually logs the driving patterns.

- ➢ Our 3 states are: parked, moving, and stopped
- ➢ Monitored through the user interface by security personnel or data scientist.

Vehicle detector can track at least 5 cars

Cameras will be 12 feet off the ground looking out. This has been successfully completed!

Bounding Boxes around the cars

The coordinates will be sent to our database. This has been successfully completed!

Tie the back-end with the front-end successfully

GUI should show data associated with the video playing. This is in progress.

Implementation works as a desktop app on a GPU workstation/laptop

Evaluation of Alternative Solutions

Detection Algorithms

YOLOv8

Fastest

Most accurate for object detection

Mask R-CNN

Slower

Most accurate for object area

Tracking Algorithms

DeepSORT

An established method

Poor FPS

ByteTrack

Higher FPS than DeepSORT

Better supporting documentation /tutorials

SMILETrack

Cutting edge- Ranked the highest for HOTA, MOTA, & IDF1

Evaluation of Alternative Solutions Cont.

Databases

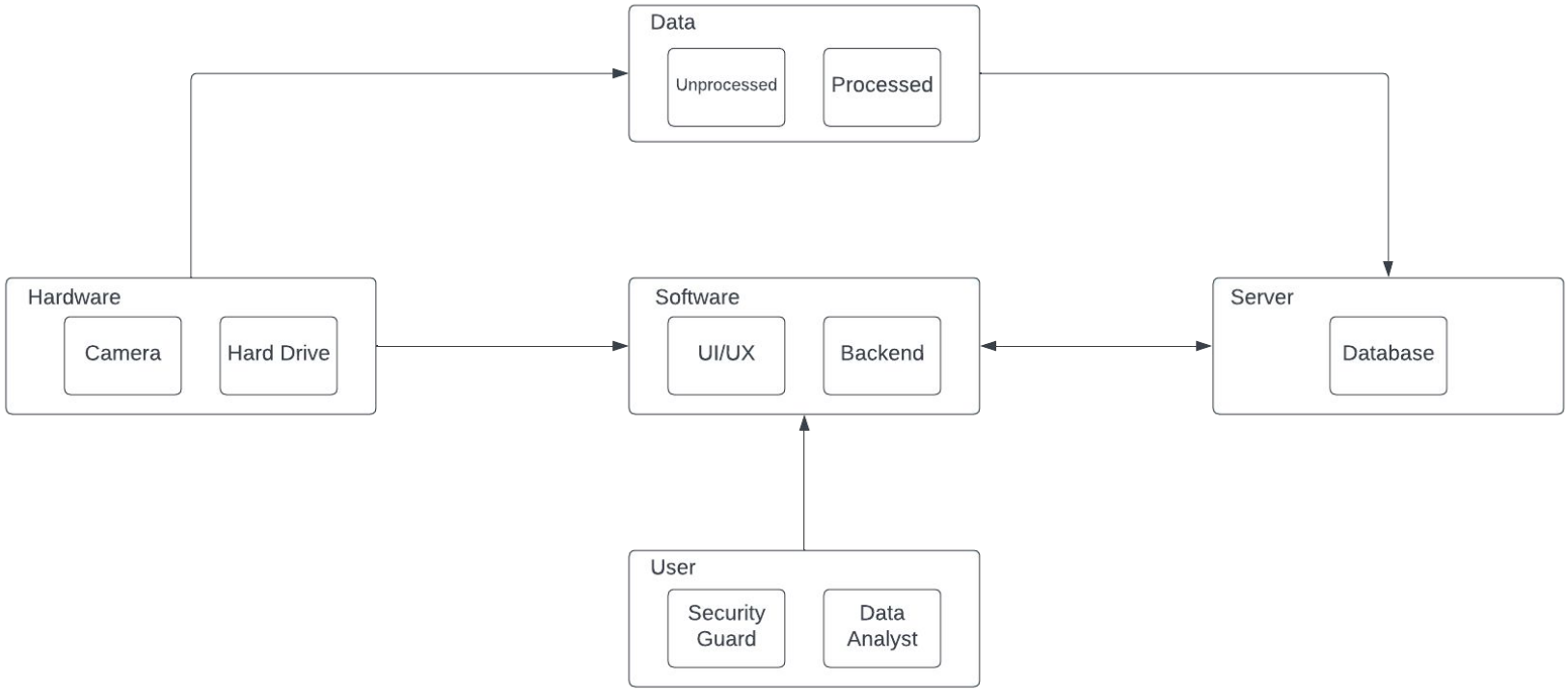
MongoDB

- Flexibility
- Ease of Use with Atlas
- Preservation of Data if changes need to be made

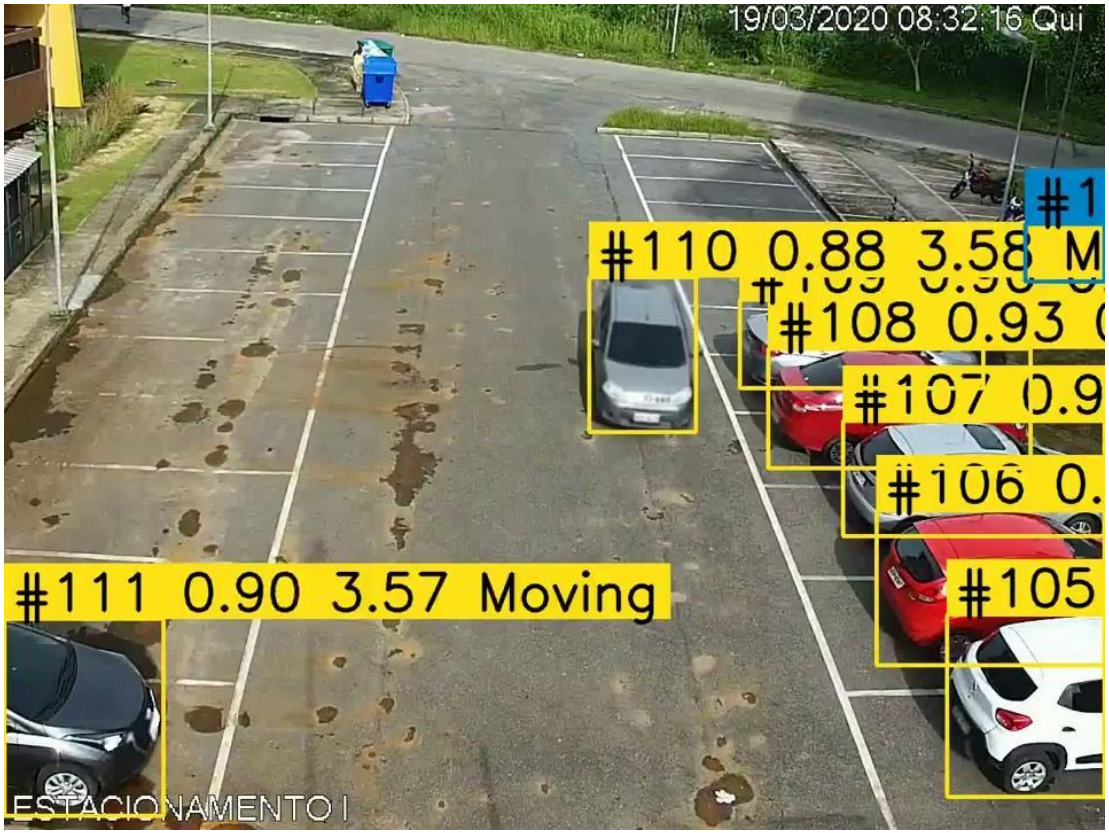
SQL

- Not as Flexible
- Changes result in obsolete Data

System Level Description

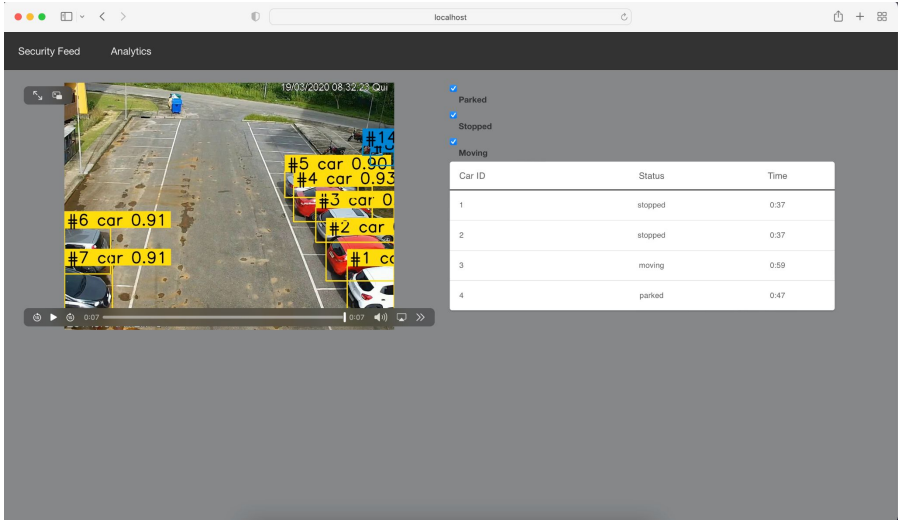


Machine Learning Video Results

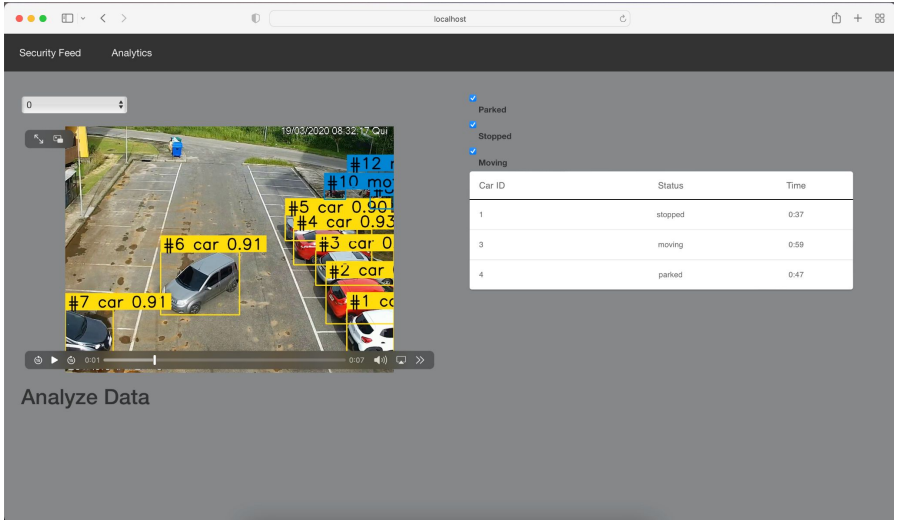


Our User Interface

Security Guard View



Analyst View



Database

- Currently using MongoDB for the Database.
- Easy to implement with the Frontend (nodejs / react)
- Information is stored into MongoDB Atlas
    - Cloud Platform
- Contains the following
    - X and Y
    - BBox Width and BBox Height
    - Timestamp
    - State
    - Unique ID

Approach for Design Validation

- MVP Requirements
- 1 USB Camera should be able to send a live feed to our application (up to 5 cars in frame)
- *Does not have to be live, annotated video can be premade
- Web Application should be able to view this feed, view details of parked,stopped, not stopped vehicles. Illegally parked, speeding are optional metrics
- Business analyst view should update with data from MongoDB database. Have historical data in table that can be filtered by state or time or camera
- Application should run in a Docker container smoothly with reasonable usage of resources

ML Detection

5 cars 10 FPS
>85% accuracy
Vehicle status

Backend

MongoDB Database
Record position,
time, status,id

GUI

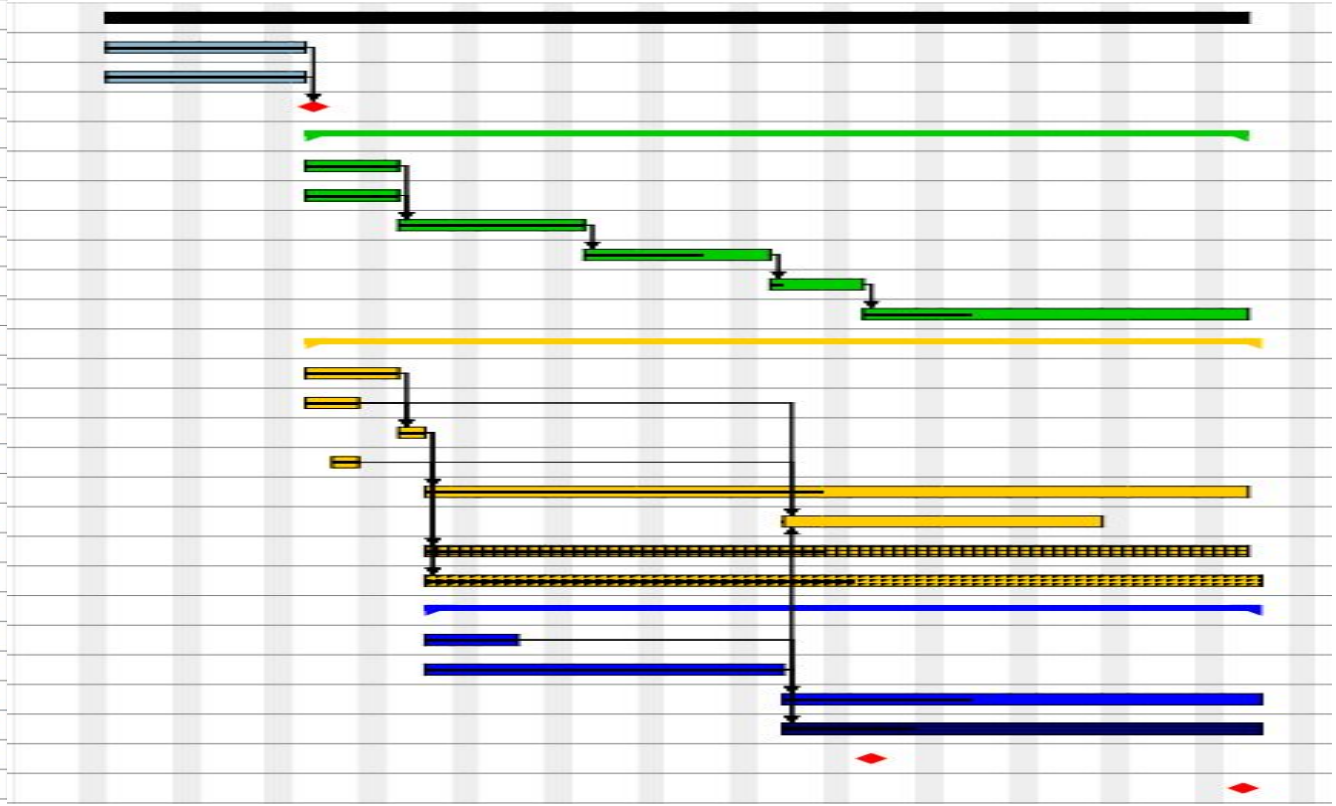
2 separate views
Filter by status

Infrastructure

Runs in Docker
container
1GB VRAM Linux

Timeline

Vehicle Classification Project	1/30/23	4/25/23
Project Proposal	1/30/23	2/13/23
Project Proposal Slides	1/30/23	2/13/23
Project Proposal Presentation	2/14/23	2/14/23
Vehicle Classification Machine Learning	2/14/23	4/25/23
Organize Machine Learning Data	2/14/23	2/20/23
ML Model Brainstorming	2/14/23	2/20/23
ML Model Implementation	2/21/23	3/6/23
ML Model Training	3/7/23	3/20/23
ML Evaluation	3/21/23	3/27/23
ML Tuning	3/28/23	4/25/23
Database	2/14/23	4/26/23
Database Schema Design	2/14/23	2/20/23
Database Server Setup	2/14/23	2/17/23
Database Table Creation	2/21/23	2/22/23
Database I/O Format	2/16/23	2/17/23
Database Management Software...	2/23/23	4/25/23
API Implementation	3/22/23	4/14/23
Security User Data Access Organization	2/23/23	4/25/23
Analyst User Data Access Organization	2/23/23	4/26/23
UI	2/23/23	4/26/23
User Interface Sketch	2/23/23	3/1/23
User Interface Prototype	2/23/23	3/21/23
User Interface Security	3/22/23	4/26/23
User Interface Analyst	3/22/23	4/26/23
CDR Presentation	3/28/23	3/28/23
Project Demo	4/25/23	4/25/23



Project Management and Teamwork

- Members divide into 3 pairs
- Tasked with module connections
- Assignments are flexible
- 2 in-person meetings weekly

ML Recognition

Aniruddha Srinivasan & Jacqueline Mioduski

Database Server Management

Tyler Roosth & Coleman Todd

User Interface

Fabianna Barbarino & Spencer Cho

Docker & Documentation

All Member & Rotating

Vision for Final Product

- Fully containerized CV pipeline
- Graphical user interface for security personnel
- Ability to identify 3 main vehicle states
  - Moving, parked, stopped
- Use of video files to be processed

Stretch Goals

- Additional vehicle states
  - Speeding, illegally parked
- Analyst GUI
- Live video feed to be processed

Societal, Safety, and Environmental Analysis

Beneficial Impact

- Increase in parking lot safety

Detrimental Impact

- Potential loss of privacy due to information being stored and cameras recording

Safety Precautions we must take

- Inclement weather may damage camera

Environmental Impact

- Carbon emissions from streaming video footage, but these cameras have to be on.

Manufacturability, Sustainability, and Economics

Economics

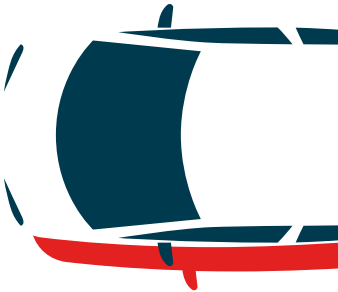
- ★ creates a flexible, repeatable, efficient, & cost-effective process to assist parking lot monitoring

Sustainability

- ★ analytics may be used to reduce parking lot traffic density = lower carbon footprint

Manufacturability

- ★ will use existing architecture to gather, process, & store data



THANK YOU!

Do you have any questions?