VehID – Vehicle Recognition Software

Members: Remington Greko, Spencer Hirsch, Thomas Johnson, and Alexis Nagle

Faculty Advisor: Dr. Silaghi

Client: Clayton Levins

Goal and Motivation

- Machine learning to recognize vehicles based upon characteristics
    - Color, body type, make, and/or license plate.
- Will be used to aid in public safety in a variety of situations
    - AMBER alerts, stolen vehicles, and criminal offenses.
- Improvement upon the current systems used in these situations
    - Typically rely on pure human interaction

Approach

Identify vehicles based upon a given criteria

Identify numerous vehicles in real-time

Report Vehicles when full or partial matches are found

Algorithms and Tools

- Python
- Tensorflow & Keras
- Open CV
- Convolutional Neural Networks
- MERN Stack
    - MongoDB, ExpressJS, ReactJS, & NodeJS

Novel Features and Functionalities

Convolutional Neural Network

Identify vehicles based on characteristics, offering a constant patrol for suspected vehicles used in crimes.

Existing Network of Cameras

Integrate software with existing network of cameras.

Vehicles as Identifiers

Rather than using license plate recognition, use vehicles as primary identifier and verify using license plate recognition.

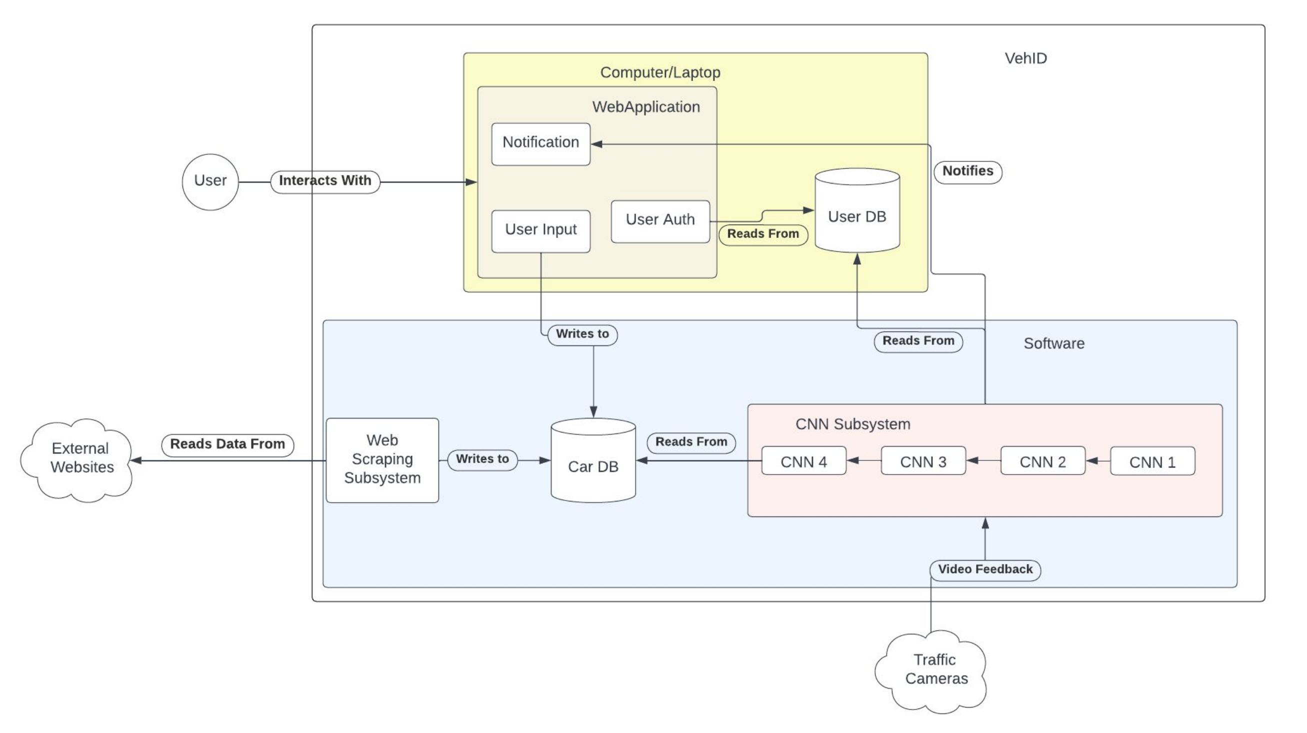
Technical Challenges

1. Training CNN to meet project requirements, limited knowledge of neural networks will at additional difficulty to this task.

2. Computer vision and recognizing numerous vehicles in a single frame. Model needs to accurately identify every vehicle contained within a given frame.

3. Designing a web application using unfamiliar technologies such as database hosting and full-stack web development.

Design


VehID
Computer/Laptop
WebApplication
Notification
User Input
User Auth
Reads From
User DB
Notifies
User
Interacts With
Writes to
Reads From
Software
CNN Subsystem
CNN 4
CNN 3
CNN 2
CNN 1
External Websites
Reads Data From
Web Scraping Subsystem
Writes to
Car DB
Reads From
Video Feedback
Traffic Cameras

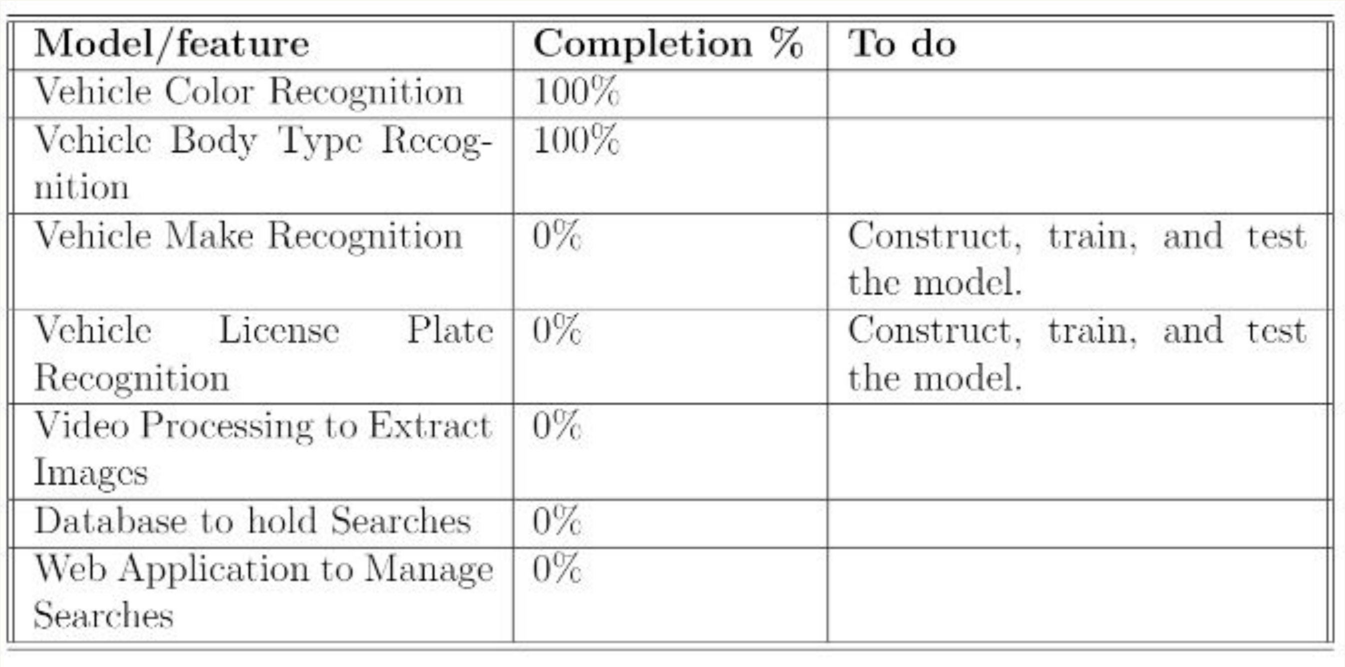
Evaluation

- Accuracy
    - Ensure accuracy of model predictions
    - Meet correctness requirements for real-world applications
        - Greater than 90%
- Reliability
    - Ensure correctness of predictions in video feed
    - Ensure correctness of reported data
- Intuitive UI Design
    - Ensure easy-to-follow UI
    - Create a comfortable and natural feel for users
        - Can at times be difficult for technical applications

Evaluation Cont.

- Speed
  - Swift data transfer between neural networks, database, and web application
  - Clean and efficient transactions between all model components
  - Quick decision making in all neural networks
    - 4 different models to process vehicles
- User Survey
  - Issue user survey to first time users of web application
  - Collect feedback and make necessary changes

Progress Summary



Model/feature	Completion %	To do
Vehicle Color Recognition	100%	
Vehicle Body Type Recognition	100%	
Vehicle Make Recognition	0%	Construct, train, and test the model.
Vehicle License Plate Recognition	0%	Construct, train, and test the model.
Video Processing to Extract Images	0%	
Database to hold Searches	0%	
Web Application to Manage Searches	0%	

Milestone 4

- Begin implementation of backend database for queries
- Begin implementation of web application for database entries
- Implement vehicle make recognition model

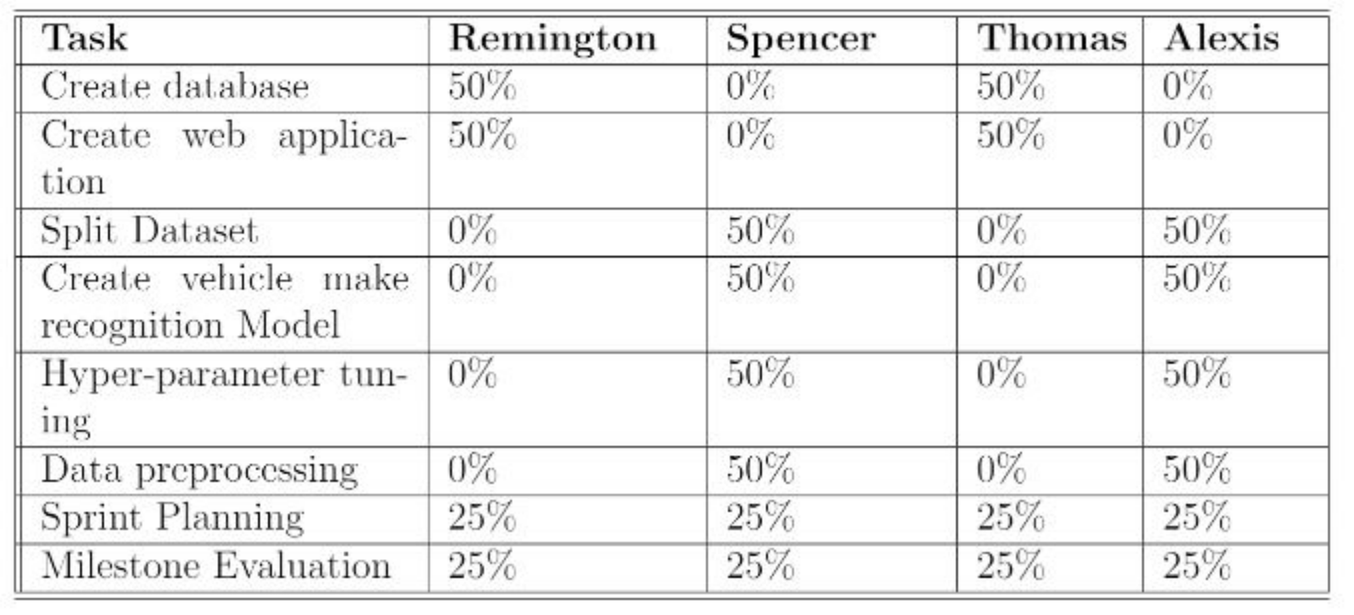
Milestone 5

- Implement license plate recognition model
- Implement video processing to extract images to categorize
- Create poster and ebook page for Senior Design Showcase

Milestone 6

- Test/demo of the entire system
- Conduct evaluation and analyze results
- Create user/developer manual
- Create demo video

Task Matrix - Milestone 4



Task	Remington	Spencer	Thomas	Alexis
Create database	50%	0%	50%	0%
Create web application	50%	0%	50%	0%
Split Dataset	0%	50%	0%	50%
Create vehicle make recognition Model	0%	50%	0%	50%
Hyper-parameter tuning	0%	50%	0%	50%
Data preprocessing	0%	50%	0%	50%
Sprint Planning	25%	25%	25%	25%
Milestone Evaluation	25%	25%	25%	25%