Ben Fernandez, Kyle Rood, Suraj Shah

1/26/2020

30% 40% 70% 90% Demo Overview

Next week mentor meeting deliverables:

What's working so far

* Bootcamp
	+ Three independent parts
		- API/Data collection - gets data from TrafficLand API and store the image information in the database as well as create API to connect to the database
		- Machine Learning - be able to detect cars in a single image, sending detections to API, beginning color detection
		- Front end - have a tool that can display an image of a car with its labels, already stored somewhere, and get information manually about the car (without the API)
* What will be ready by 30%
	+ Integration
		- Frontend and Backend services integrated on local database system
		- Usage of image and camera logistics API to display the latest downloaded image from TrafficLand services
	+ Machine Learning
		- Able to detect cars
		- Able to download images from API
		- Run Car detection on images and output bounding boxes
		- Working on color and size detection
* What will be ready by 40%
	+ Machine Learning
		- Honing color/ size detector to give us a better distinction
		- There is a pipeline between the API and the machine learning
			* machine-learning can get images and cameras from the API, and can send bounding boxes/detections to database with POST request
	+ Data Collection:
		- Using a NoSQL database instead of SQL so that the JSON is retrieving document data instead of converting it
			* Will result in faster API collection
	+ Front-end:
		- All the camera metadata will be added to the frontend web app
		- This includes orientation, location, full name, last updated time, etc
* What will be ready for 70%
	+ Integration
		- Server logistics → will take time to set up live server
			* Want to instead have everything locally
		- Will have fully integrated local version
			* Ready to deploy once we get server set up
	+ Machine Learning
		- Good color detection results
		- Training model to improve YOLOs initial output
	+ Data Collection
		- Have image data collection optimized to be as fast as possible
		- Have all the endpoints for the API for machine learning to POST bounding boxes
	+ Front-End
		- Initial creation of query function for historical data to be displayed on frontend
* If the whole project was one zip file?
	+ You would see a few separate files and a front-end application hosted in one Single Page Application that uses parts of each
	+ Since each part is independent of each other, you would see a few files that are for the API and the data collection
	+ Machine learning files that put bounding boxes that gets images from an API call with our own self-built API
	+ SPA that calls our self-built API to get the bounding boxes labels and the image and displays the image with the bounding boxes and retrieves information about the specific image

**70%**

What were your goals and what were you able to achieve?

* Data-Collection/API
	+ The primary goal was an integration between the machine learning, front-end, and the database/API
	+ Having the server so that we can run this in the background and host it to the public
* Frontend
	+ Showing detected cars on the frontend
* Machine learning
	+ Didn’t get as positive results as we wanted (despite positive visualizations)
		- Continue to debug this
		- Will also pursue Deep Learning approach

**90%**

What are the goals and what will you have working?

* Machine Learning
	+ Solid Color/Size Detection
	+ Running quickly in the background to be as close to real-time as possible
* Data Collection/Analysis
	+ Heatmap of vehicle count so that it can show real-time congestion
	+ Tracking whether a car is in a grid of cameras and using human intuition to provide a range of time of when a car crossed the traffic camera
* AmberVision Logistics/Access
	+ Full application running on SEAS hosted webserver
* Front End
	+ Query function to search for images
	+ Display carousel of images depending on time of day or camera
	+ Tracking whether a car is in a grid of cameras and using human intuition to provide a range of time of when a car crossed the traffic camera
	+ Heatmap of vehicle count so that it can show real-time congestion