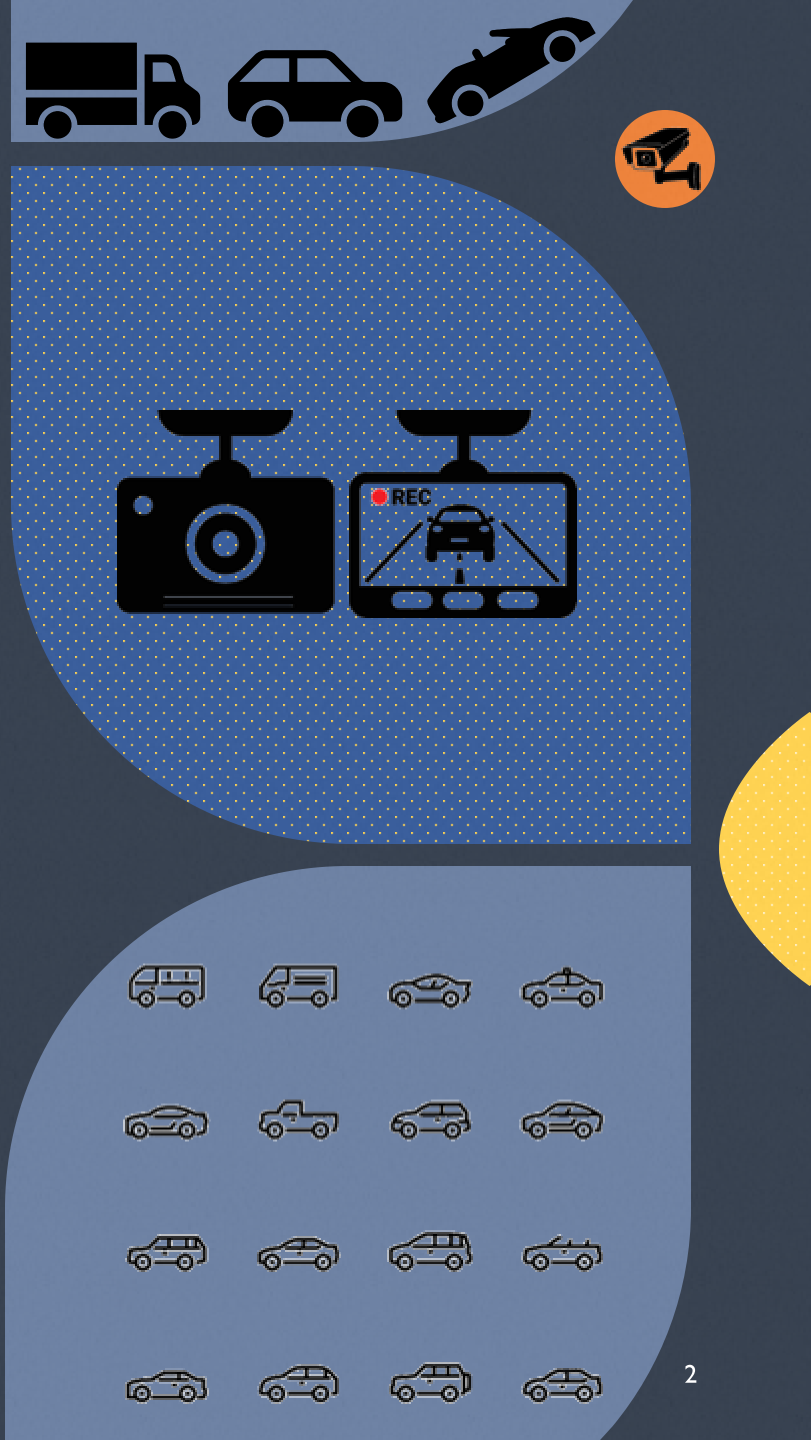
MotorHead Car Vision

Apoorv, Austen, Dylan, Michael, Sivaram

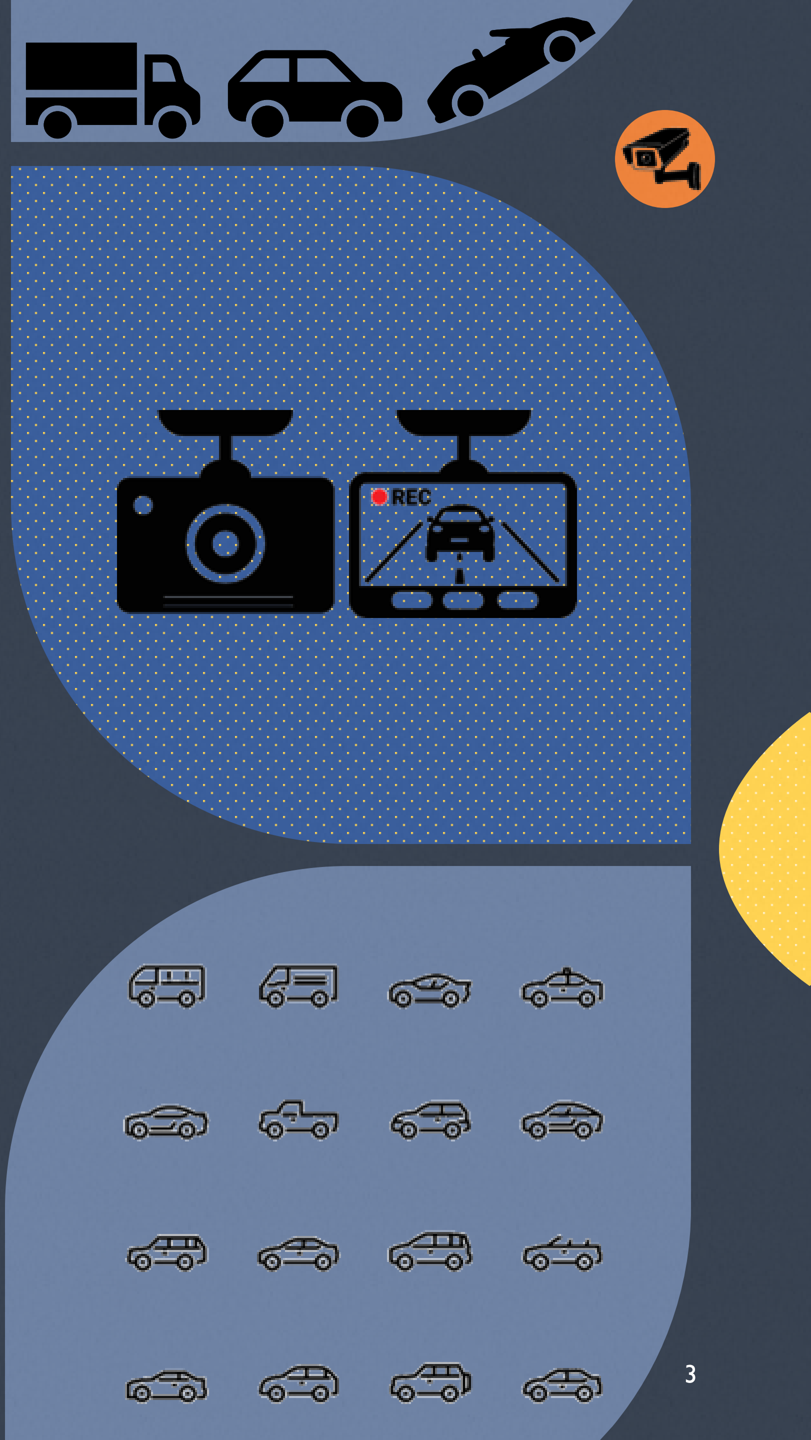
Agenda

- Business Problem
- Methodology
- Dataset Overview
- Exploratory Analysis
- Object Detection Approaches
- Classification Approaches
- Ensemble Approaches
- Findings & Conclusions
- Limitations
- Proposed Implementation
- Model Operations



Business Problem

- In the age of digital marketing, vehicle manufacturers could greatly benefit from more localized market information.
- By training a model to (a) identify vehicles from cameras traffic cameras or dashcams and (b) classify them by body type, we can glean valuable statistics about the vehicle “installed base” in a particular market.
- This approach could be easily adapted to a wide array of alternative classification tasks, such as:
  - Vehicle model identification (vehicle OEMs)
  - Vehicle age classification (auto insurers)
  - Accident trends (regulatory bodies)
  - Route optimization (commercial trucking)



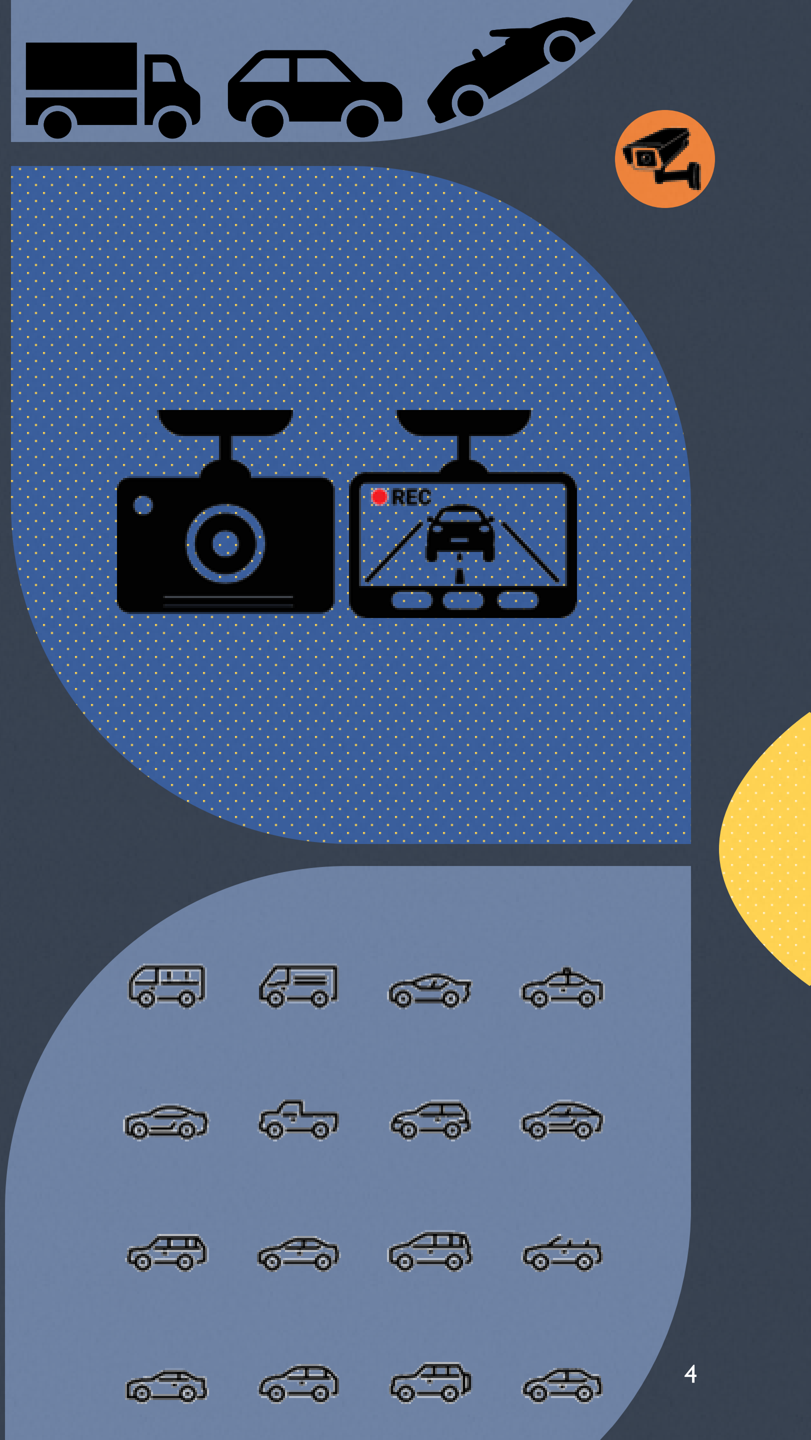
Methodology: *Analyze Performance of Two Competing Approaches*

Transfer Learning Approach

- Use car image dataset to fine tune pre-trained CNN to identify body type.
- Utilize region-based CNN (Faster R-CNN) to detect vehicles in traffic camera / dashcam footage.
- Apply transfer learning to classify car images extracted from Faster R-CNN model.

YOLO Approach

- Use YOLO to detect vehicles in traffic camera / dashcam footage.
- Manually labeled vehicles using Roboflow.
- Train YOLO to classify body type from manually labeled images.



We use 2 Kaggle Datasets to train and test our models

CUHK Comp Cars for Classification

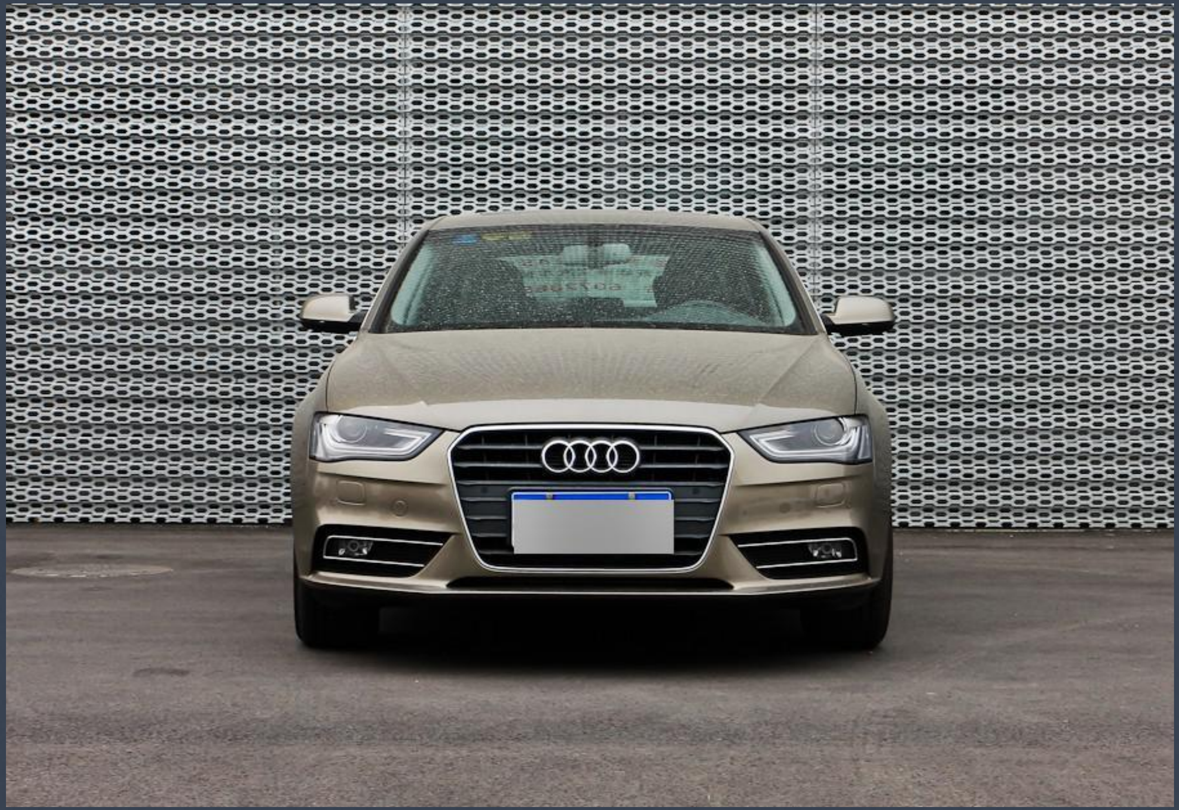
- Featuring images based on brand, type, model, and year
- Images are front, left, right, and rear of cars including top angles
- 17 GB of 130K images randomly **sampled down to 9.6K images balanced by class**

Lyft 3D Objection Detection for Autonomous Vehicles

- Featuring dashcam images taken from various angles of cars
- 100+ GB of 640K randomly **sampled down to 700 images**
- Other LIDAR data was available but not used for this problem

Further Overview of Images

Car Classification



Came with labels

Lyft Dataset

Unlabeled

Exploratory Analysis – Cars

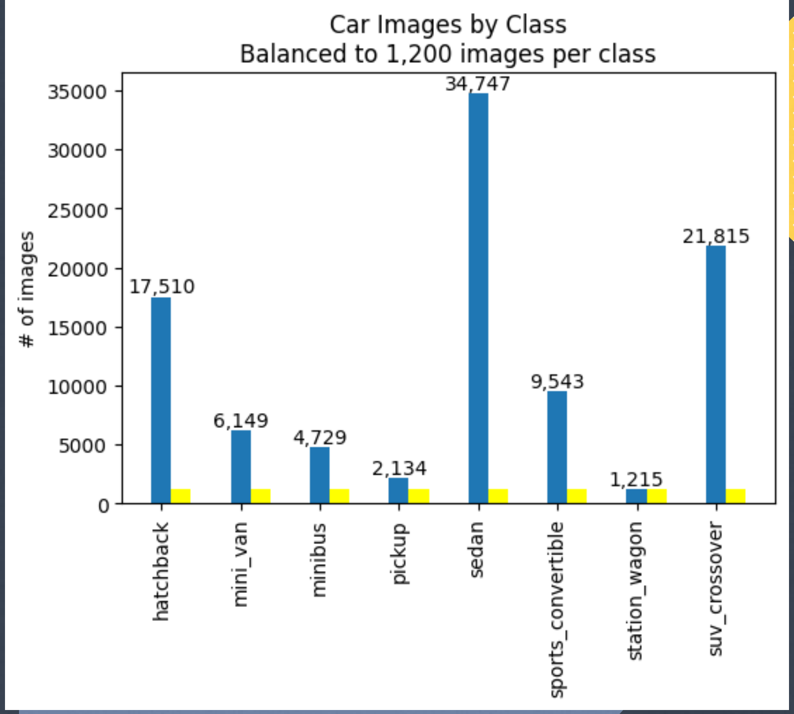
- 1,716 distinct models with 13 assigned car type classes
- "Unknown" type assigned to 748 models that would be predicted as other classes were removed, leaving over **97K** images remaining
- Consolidated 12 remaining classes into eight classes due to some classes containing similar or overlapping vehicles
- Ended with a balanced sample of 9,600 images, 1,200 per class (about 10% of data)

Images from original classes now within "sports_convertible"

Sports Car

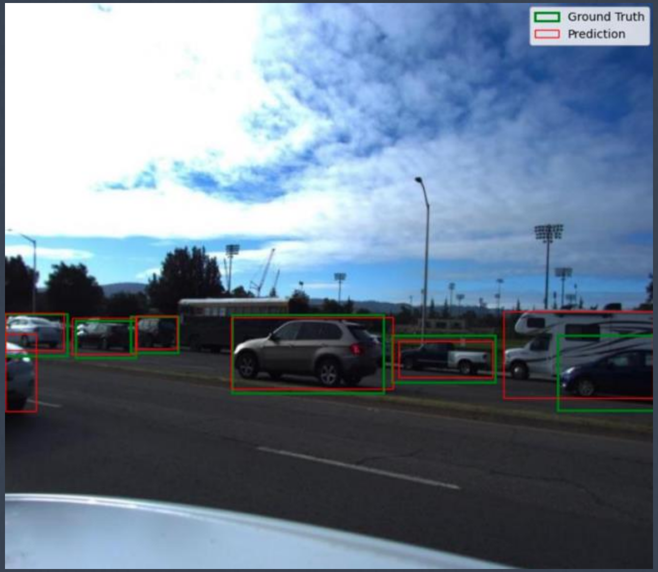
Hardtop Convertible

Convertible



Exploratory Analysis – Lyft

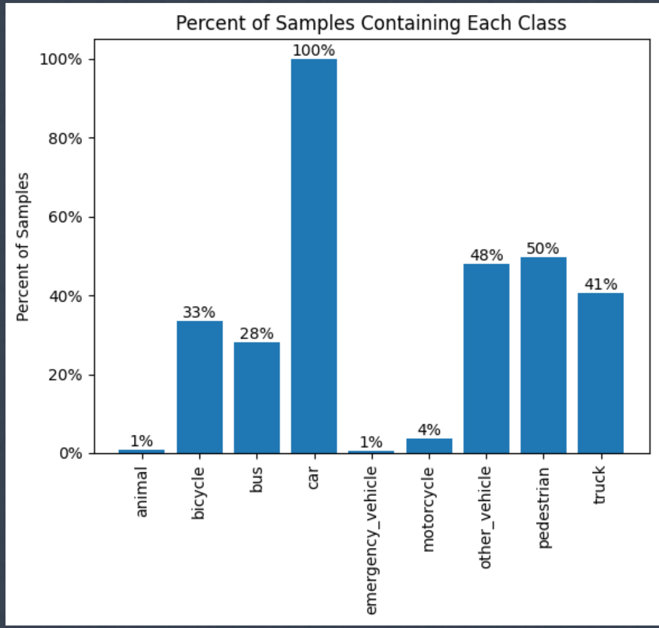
- Images provided in "samples" of 7 cameras on top of a vehicle
    - 6 images around the car and 1 image from higher above
- Photos are mostly from the US, whereas cars are more global

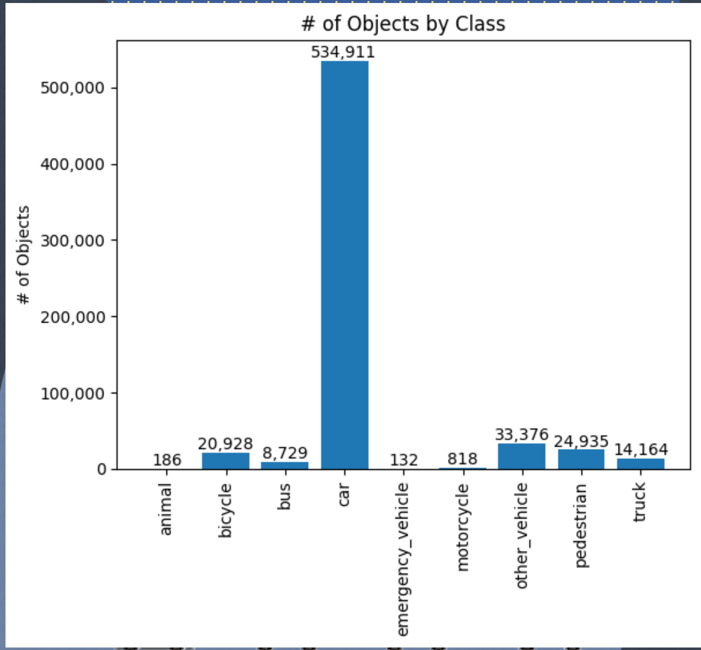


Left: labeled example
Above: elevated image
Right: 6 images 360 degrees around

Exploratory Analysis – Lyft

- 89% of labeled objects were "car" (84%) or "other vehicle" (5%)
- On average, **4** objects per image and **28** per sample
- "car" objects present in every class
- Took 100 random samples (700 images) for labeling and evaluation
  - Labels provided on the "sample" and not image-level



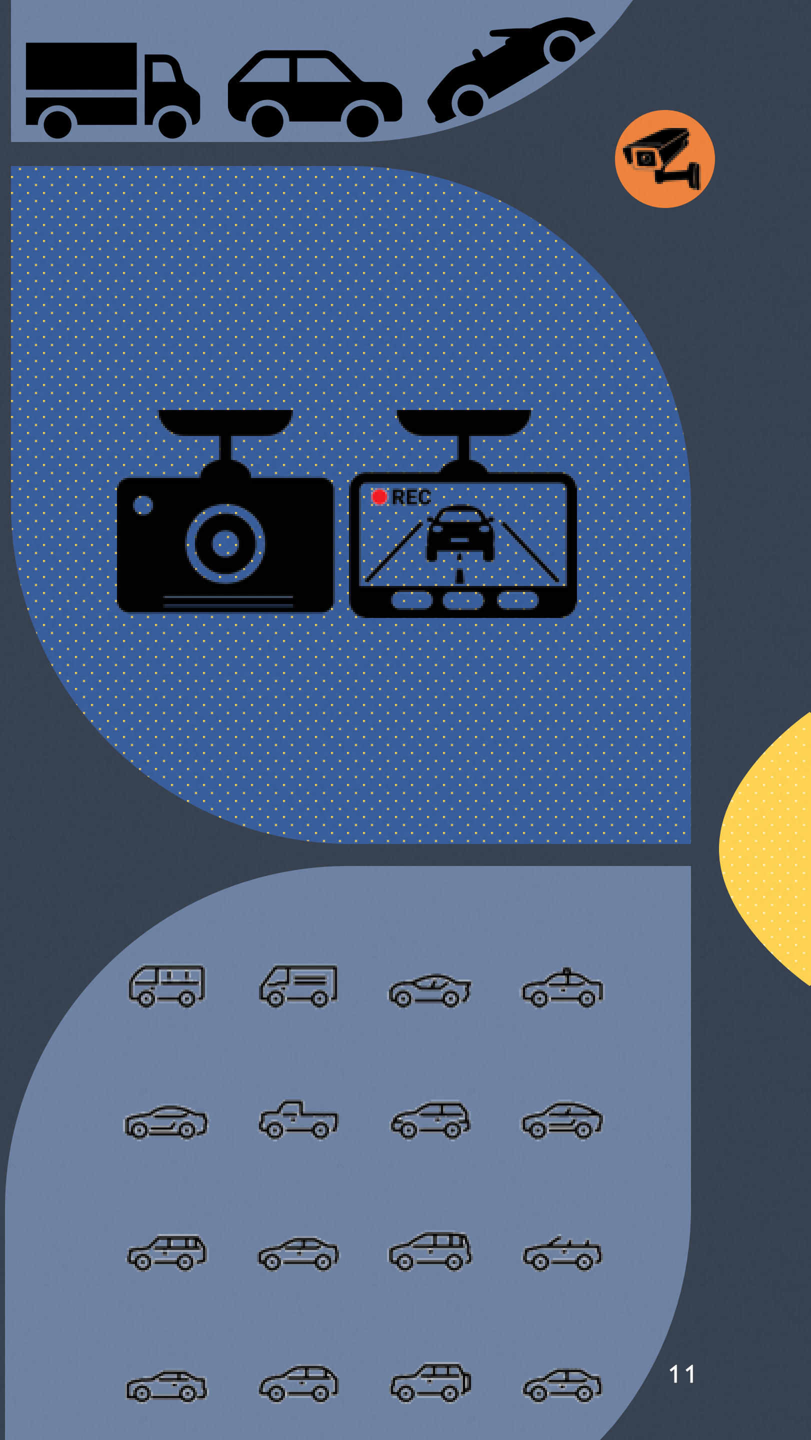


To evaluate the Lyft data, we used Roboflow to add bounding boxes and car classifications

Added bounding box labels + car class labels

- Used Roboflow to label our Lyft dataset so that we had context to test our model's predictions
- ~700 images labeled using the new AutoFlow feature in Roboflow – automatically detected car bounding boxes in our Lyft images to test our object detection models
- Hand labeled ~70 images with bounding boxes and class to test our classification models

Object Detection Approaches



Faster R-CNN

Overview

- Generates region proposals, extracts features with a CNN, and classifies each region with SVMs.
- High accuracy for detecting objects in complex scenes.
- Pros: Accurate and handles overlapping objects well.
- Cons: Slow and computationally expensive.

Model Evaluation

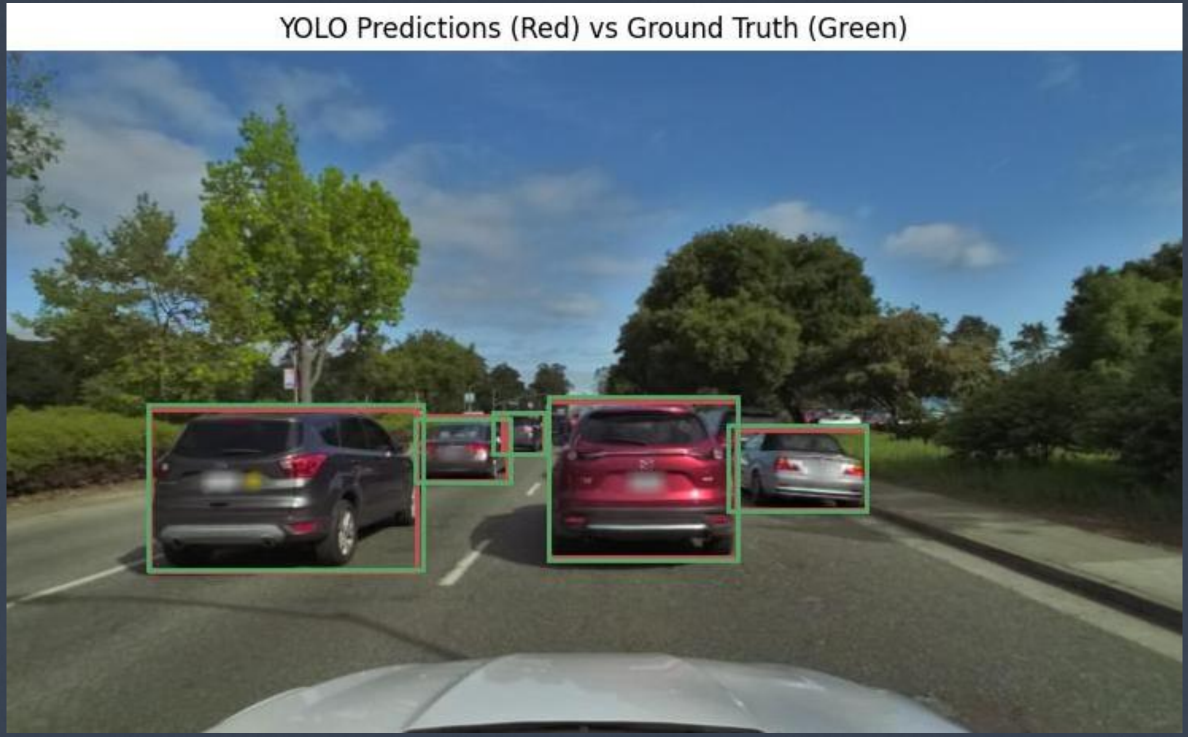
- Adept at generalizing to unseen data.
- Often detected larger vehicles (RVs, buses, etc.) despite not being labeled in training data; otherwise, <u>surprisingly few false positives</u>.

YOLO Object Detection

Overview

- Trained YOLO on ~700 RoboFlow labeled Lyft images to detect cars on the road
- Pros: Fast, real-time performance and easier to train.
- Cons: Lower accuracy for small or overlapping objects vs. region-based CNNs.

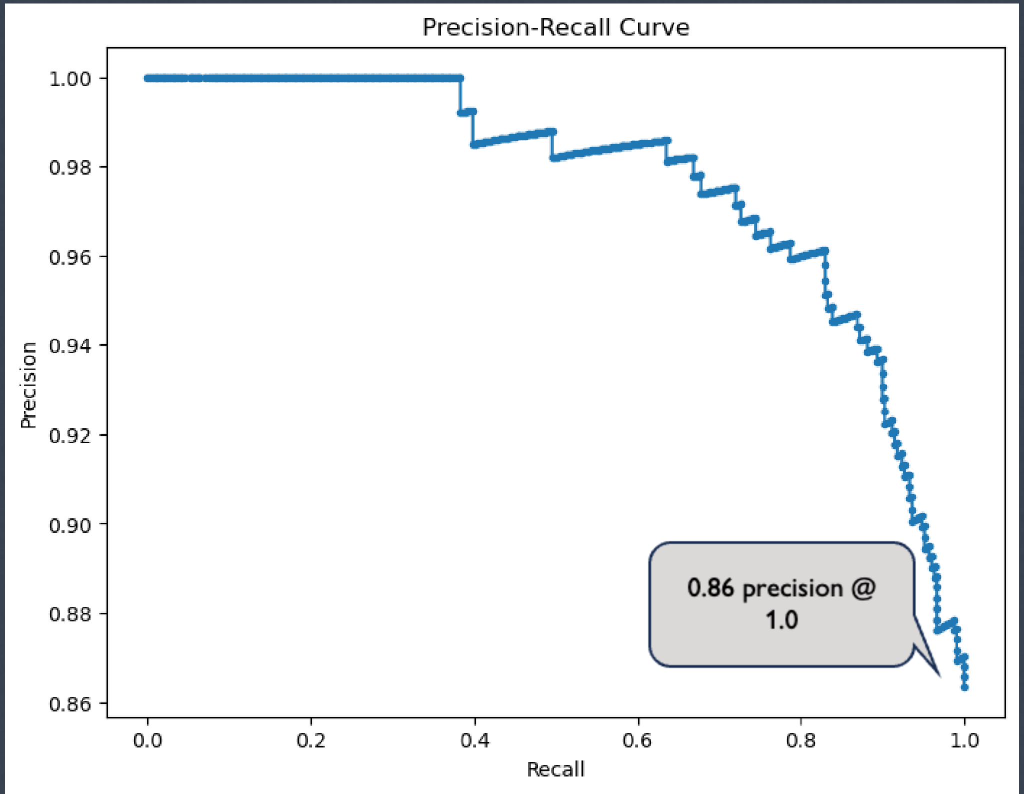
Model Evaluation



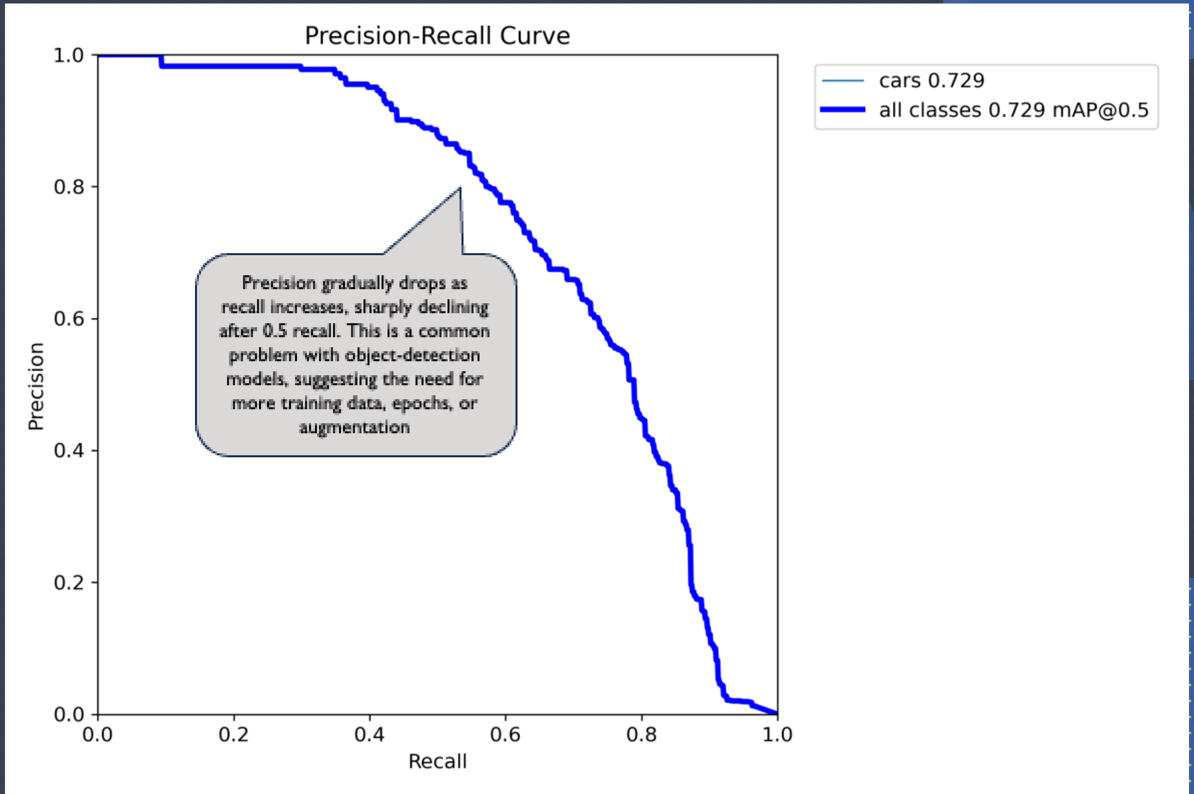
- Extremely quick and efficient predictions due to only looking once
- Missed many cars in the grainier images – solved with better imaging and more training data/time

Model Comparison

Faster R-CNN



YOLO Object Detection



- Faster R-CNN object detection appears to outperform YOLO, particularly due to its impressive avoidance of false positives.

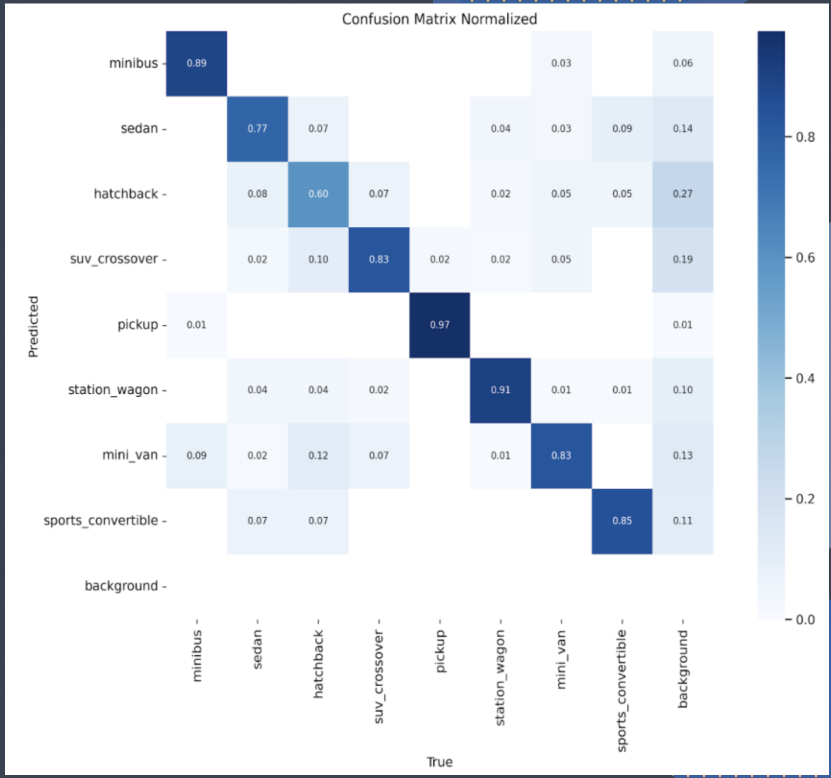
Classification Approaches

YOLO Car Classification

Overview

- Trained YOLOv8n on Cars dataset that came with class labels
- Transfer learned classes to YOLO's pre-trained class detection capabilities
- Train-test split the Cars classification dataset – tested model's ability to classify single car images in dataset

Model Evaluation



- YOLO was very successful at classifying cars in the single-image Cars Dataset – not too much in the photo other than the cars
- Particularly good at classifying more "identifiable" cars like minibuses and pickups, but still performed well with more "common" cars like SUV/Crossovers (**83%**) and sedans (**77%**)

EfficientNet (Pre-Trained CNN)

Overview

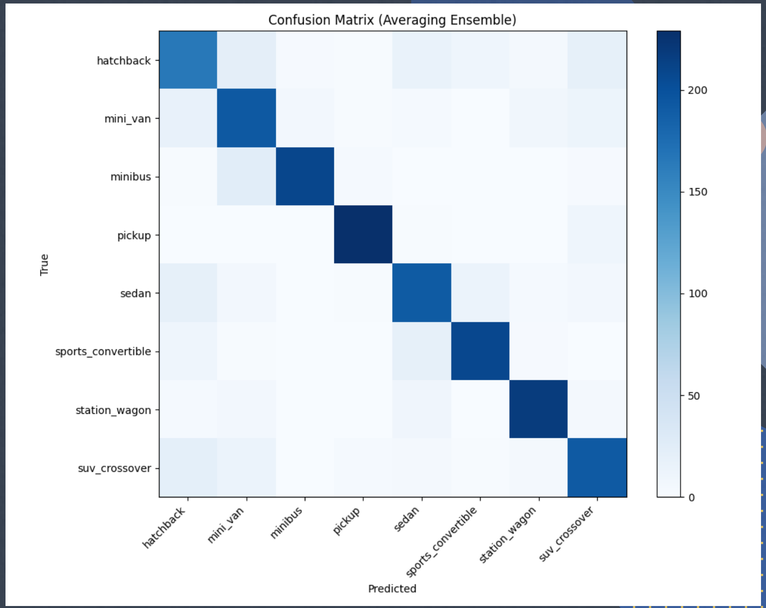
- Our model builds on EfficientNetV2L, adding custom layers and using fine-tuning for car classification
- Employ 5-fold cross-validation for robustness and create an ensemble of models to maximize accuracy.
- Pros: High classification accuracy and more generalizable.
- Cons: Longer training time and more complex implementation.

Model Evaluation

- F1 Score: 0.84
- Accuracy: 0.92
- Recall: 0.83

Classification Report (Averaging Ensemble):

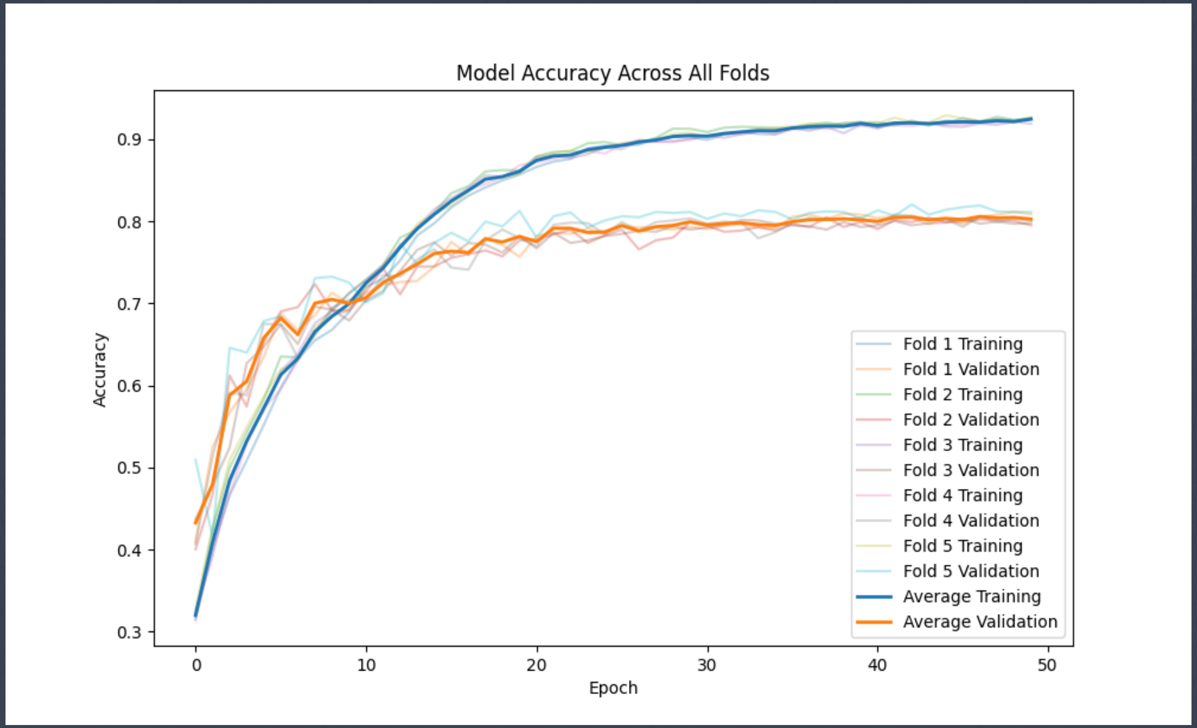
	precision	recall	f1-score	support
hatchback	0.70	0.69	0.69	240
mini_van	0.72	0.80	0.76	240
minibus	0.96	0.87	0.91	240
pickup	0.96	0.95	0.96	240
sedan	0.79	0.80	0.79	240
sports_convertible	0.89	0.87	0.88	240
station_wagon	0.90	0.91	0.91	240
suv_crossover	0.78	0.80	0.79	240



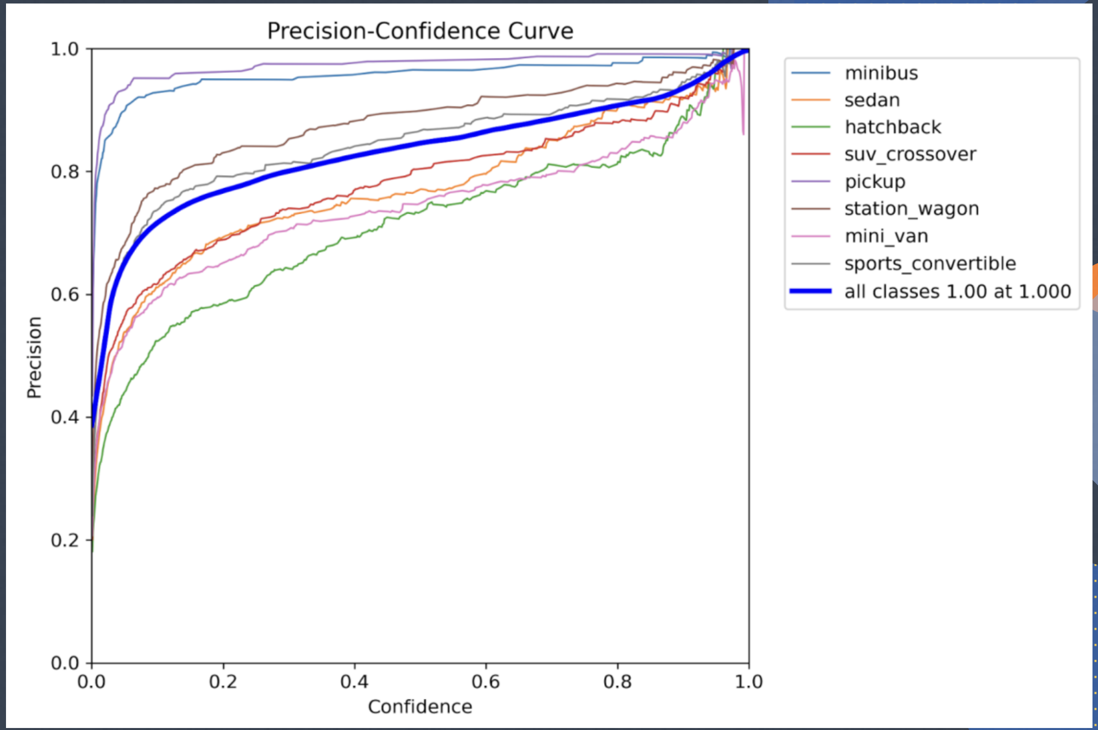
- Performed well on classes where body type is unique such as pickup vans
- Was not as accurate when it came to classes that looked similar, such as SUV and hatchback

Model Comparison

EfficientNet Car Classification

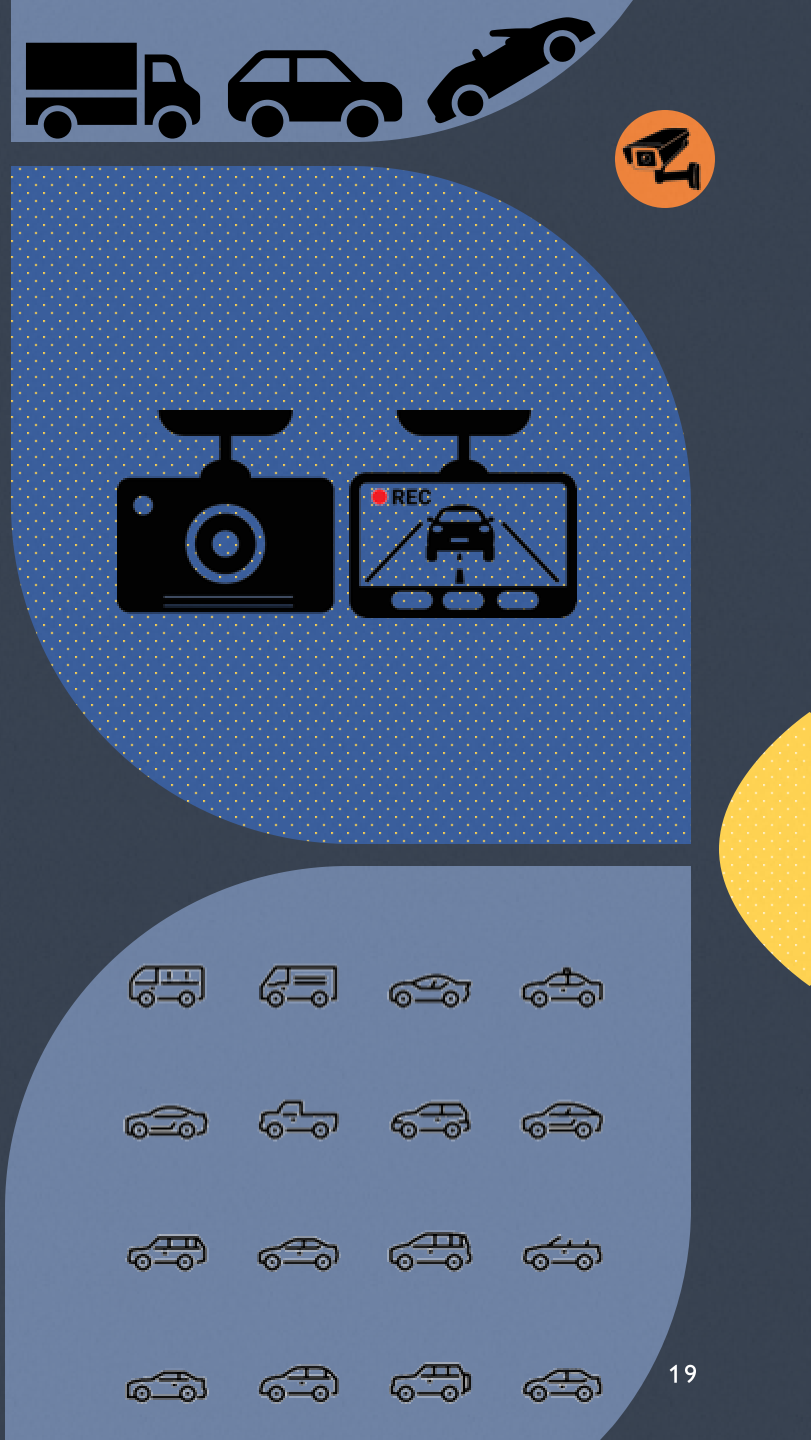


YOLO Car Classification



- The EfficientNet's accuracy in classifying models reached an excellent accuracy when ensembled
- YOLO performed extremely well across classes in terms of precision vs confidence, showing a robust ability to classify cars when just the car image was trained and tested

Ensemble Approaches

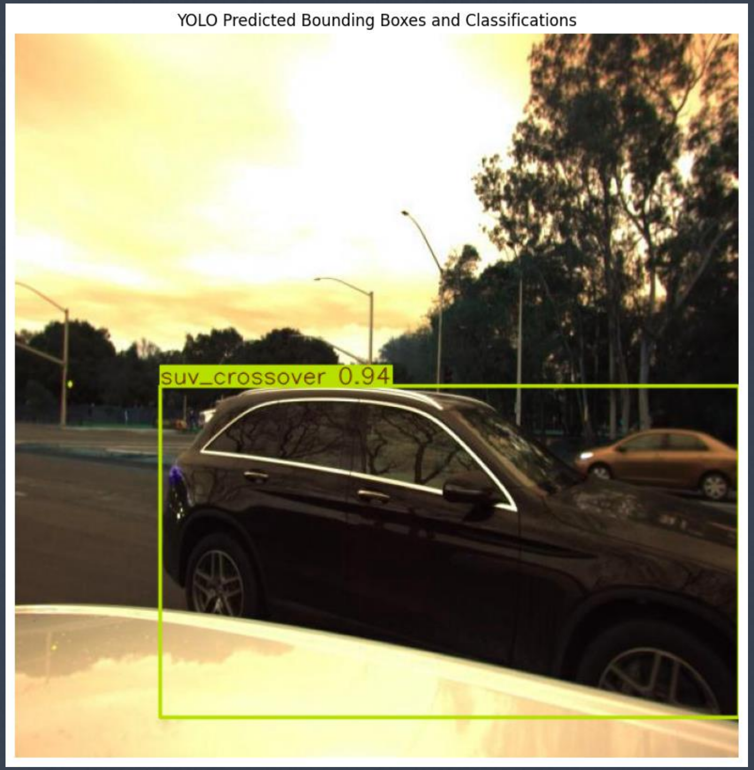


YOLO Classification + Object Detection

Overview

- Simplifies workflow by combing detection and classification into one model.
- Trained on Cars dataset to detect custom classes and objects
- Used to predict classes (car make) and object detect Lyft dashcam data
- Used RoboFlow to label classes and objects as a validation sets

Model Evaluation



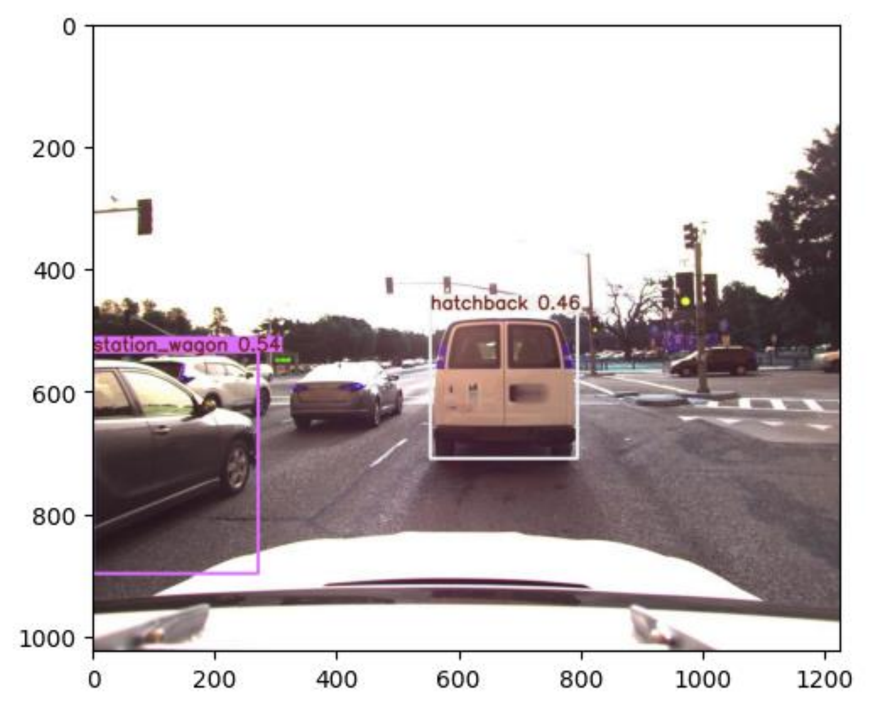
- The combined YOLO model was efficient and quick at detecting all of the cars in a frame
- This model worked well on some classes (such as SUVs and station wagons) while struggling with rarer car types (minibus, sports convertible, etc)
- Car classifications could be improved with better labeled images via Roboflow

Faster R-CNN + EfficientNet

Overview

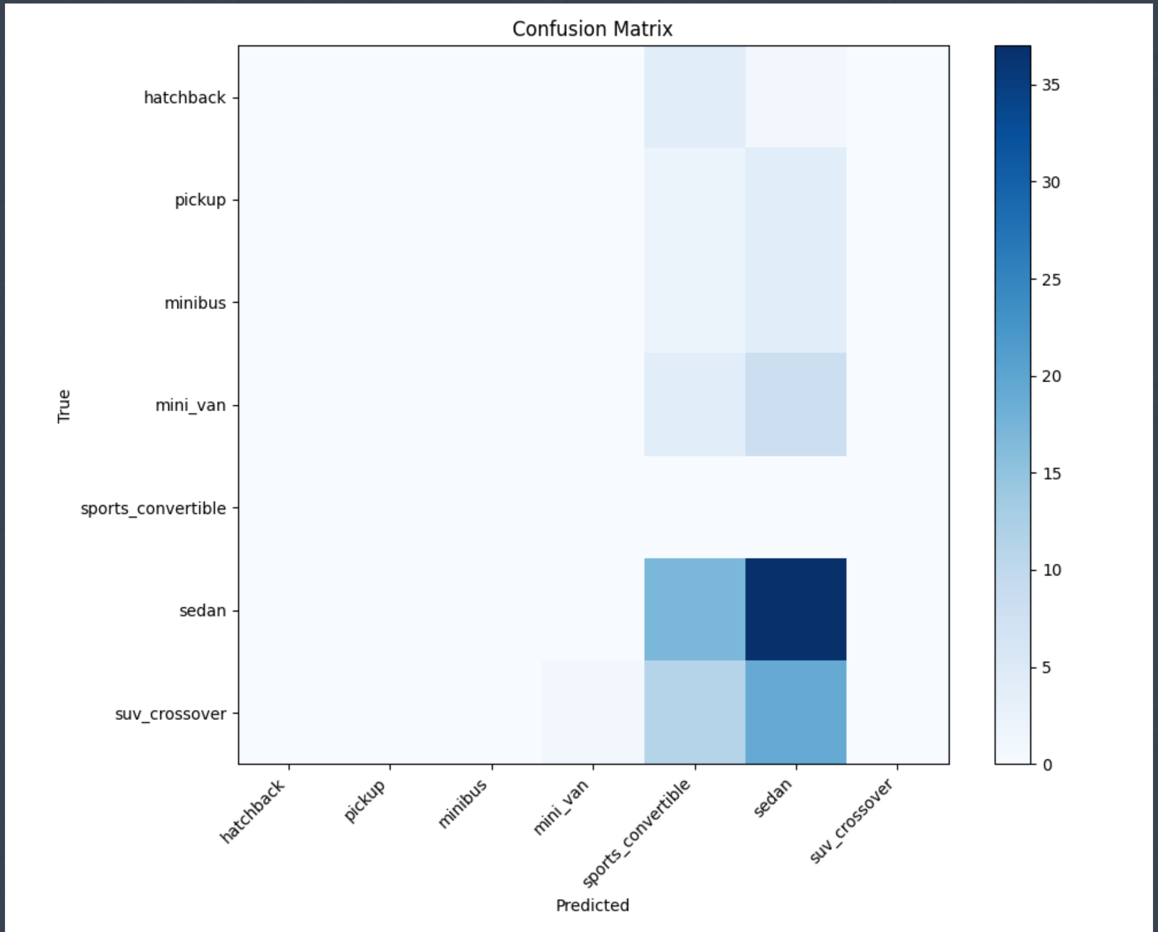
- EfficientNet provides optimized feature extraction, Faster R-CNN delivers accurate object detection and localization
- Reduced computational cost compared to heavier models
- Flexible architecture allows fine-tuning for specific applications

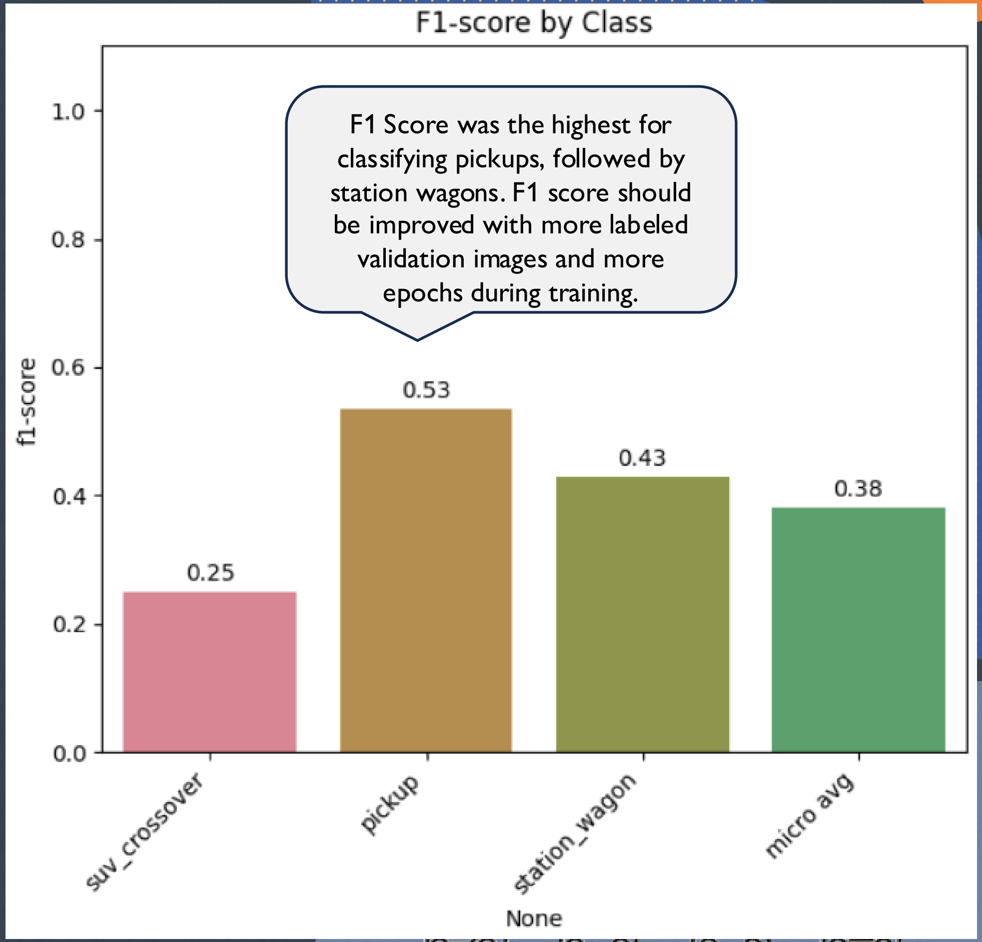
Model Evaluation



- This model had a tough time with car classes because of training set vs. Prediction set
- Car Classification data is all front-facing, clear car only images whereas Lyft data can be grainy images with many different angles of cars

Model Comparison





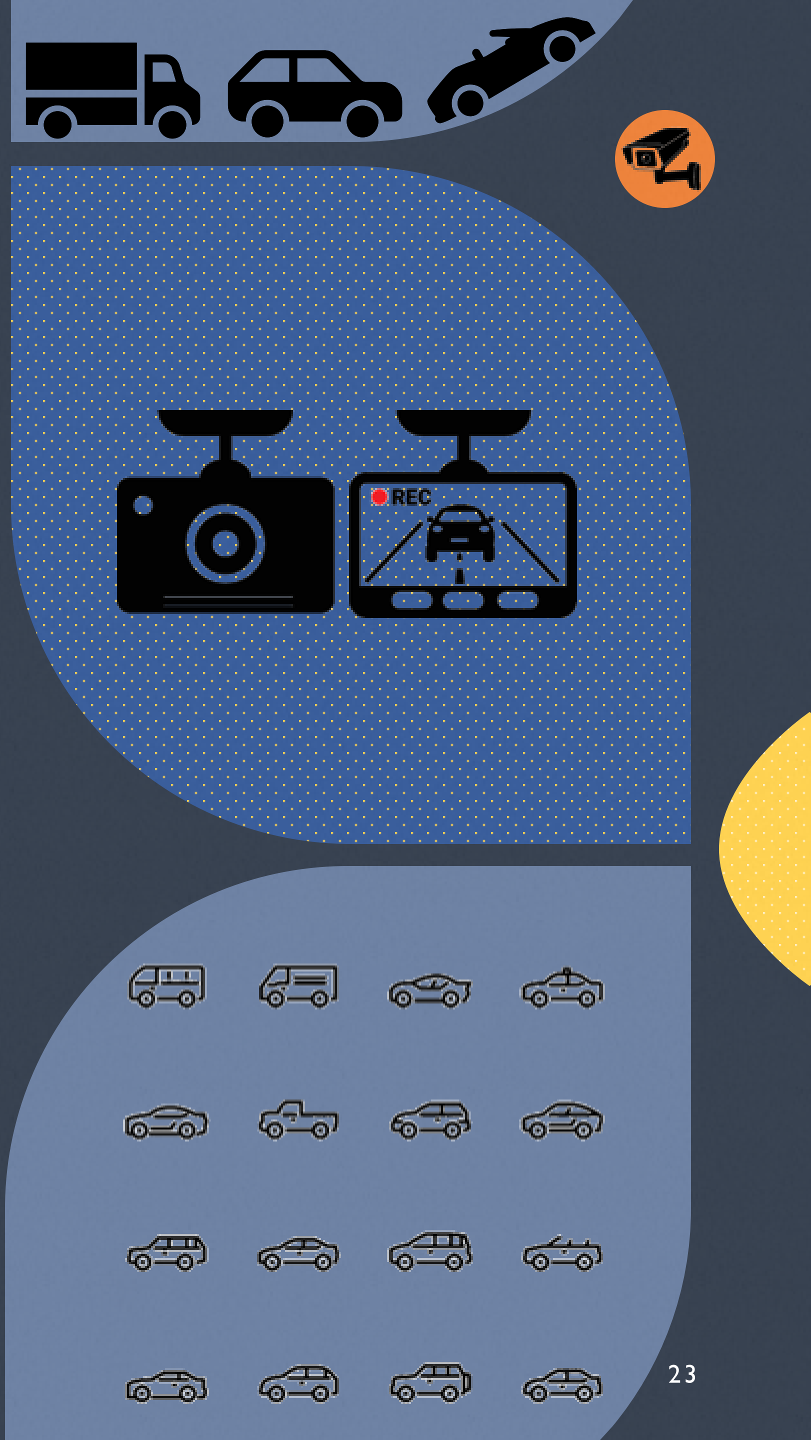
- Faster R-CNN + EfficientNet was not able to predict all classes, it was biased towards sedan and sports_convertible
- The transfer-learned YOLO outperformed our custom combination method in classifying objects and car classes in the Lyft dataset.
- Our dataset is biased towards American cars (mostly pickups, station wagons, etc.) A more balanced dataset may reduce bias

Findings & Conclusions

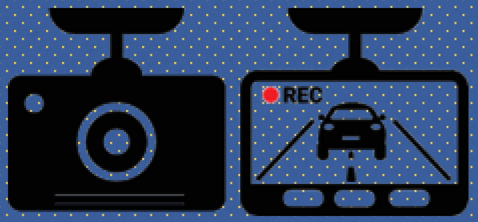
- Both object detection approaches (YOLO + region-based CNNs) are highly effective, even with moderately-sized training datasets; however, YOLO is less prone to false positives.

- Transfer learning can be very difficult in the context of image classification as subtle differences in image composition can meaningfully alter results (e.g., blurred-out license plates):
  - Model did not generalize well to different camera angles.
  - Model struggled with partial photos (e.g., front half of a car).

- YOLO is effective out of the box, but can still be challenging to successfully fine-tune in for highly-discerning classification tasks.



Limitations of Transfer Learning Approach



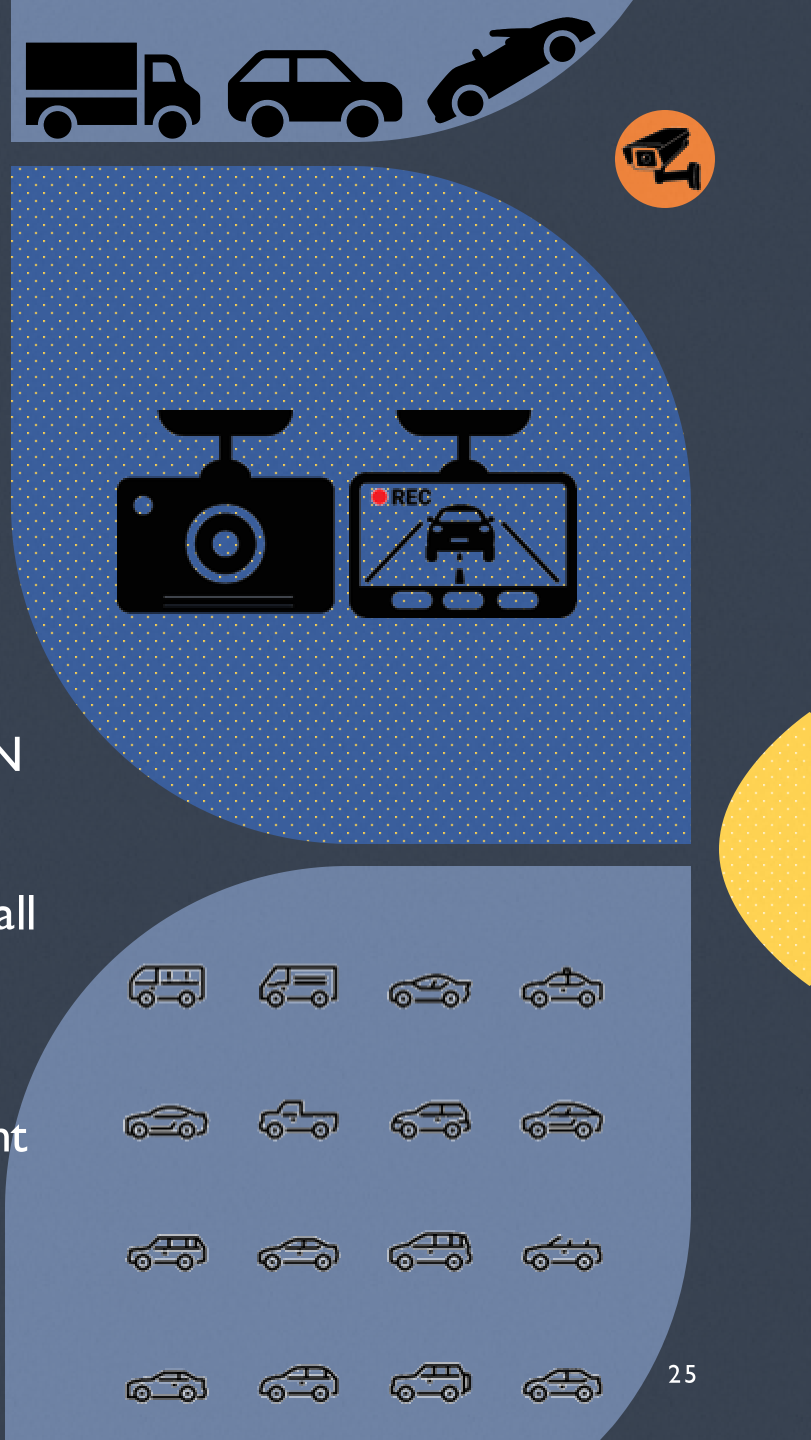
- The two datasets have some key differences:
  - Car images often in profile but dash cam footage shows cars from front and back with fewer profile shots.
  - EfficientNet was trained on a balanced sample of eight classes, but that may not reflect crossover-heavy US dash cam footage.
  - Training data included many Chinese and non-US cars that may never appear in footage.

- Classification: Tested a variety of pre-trained CNNs before landing on an EfficientNet ensemble. Much slower "trial and error" approach vs. simply relying on YOLO built-in detection capabilities.

Proposed Implementation

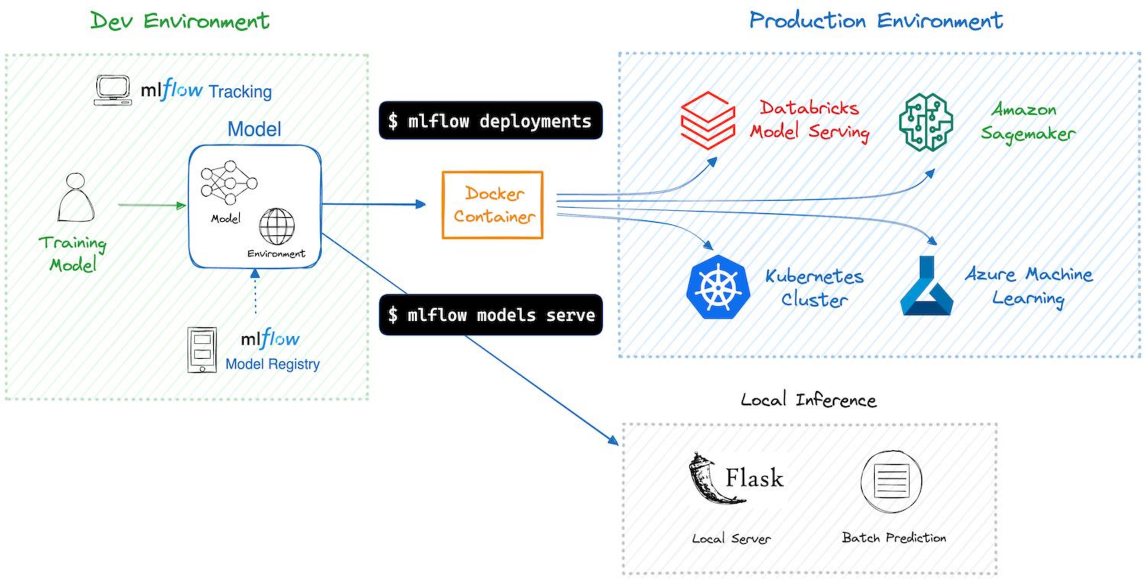
- To best identify vehicle types from Lyft dash cams, we propose leveraging a **<u>YOLO Classification + Object Detection</u>** model to collect this market-level data going forward because:

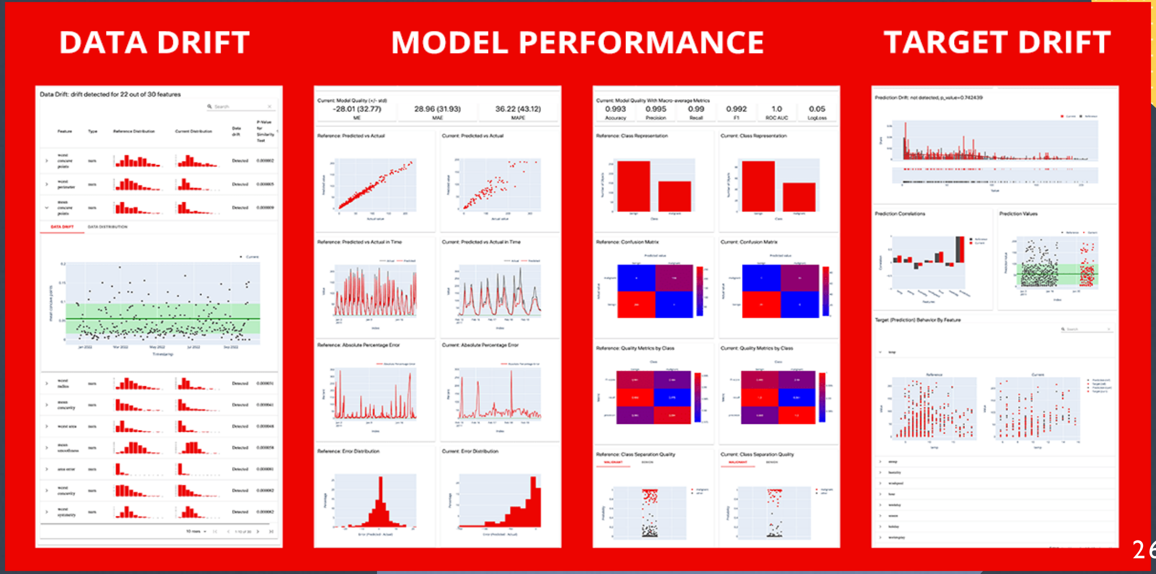
    - <u>Difficult Object Detection</u>: Cars along the road (parked or in driveways) may be close together and require more precise detection Faster R-CNN + EfficientNet struggled with
    - <u>Identifying Multiple Objects:</u> YOLO Performed better in tagging all the vehicles in the Lyft Images, Faster R-CNN + EfficientNet did not detect all
    - <u>Customizable</u>: The car data could be leveraged to classify makes and models, which could provide additional value to users
    - <u>Future Work:</u> We used the YOLOv8n due to compute and time restrictions, if we use a more advanced model such as YOLOv8l, we might be able to improve this further



Tools like MLFlow can help us deploy the model; EvidentlyAI would provide robust model monitoring

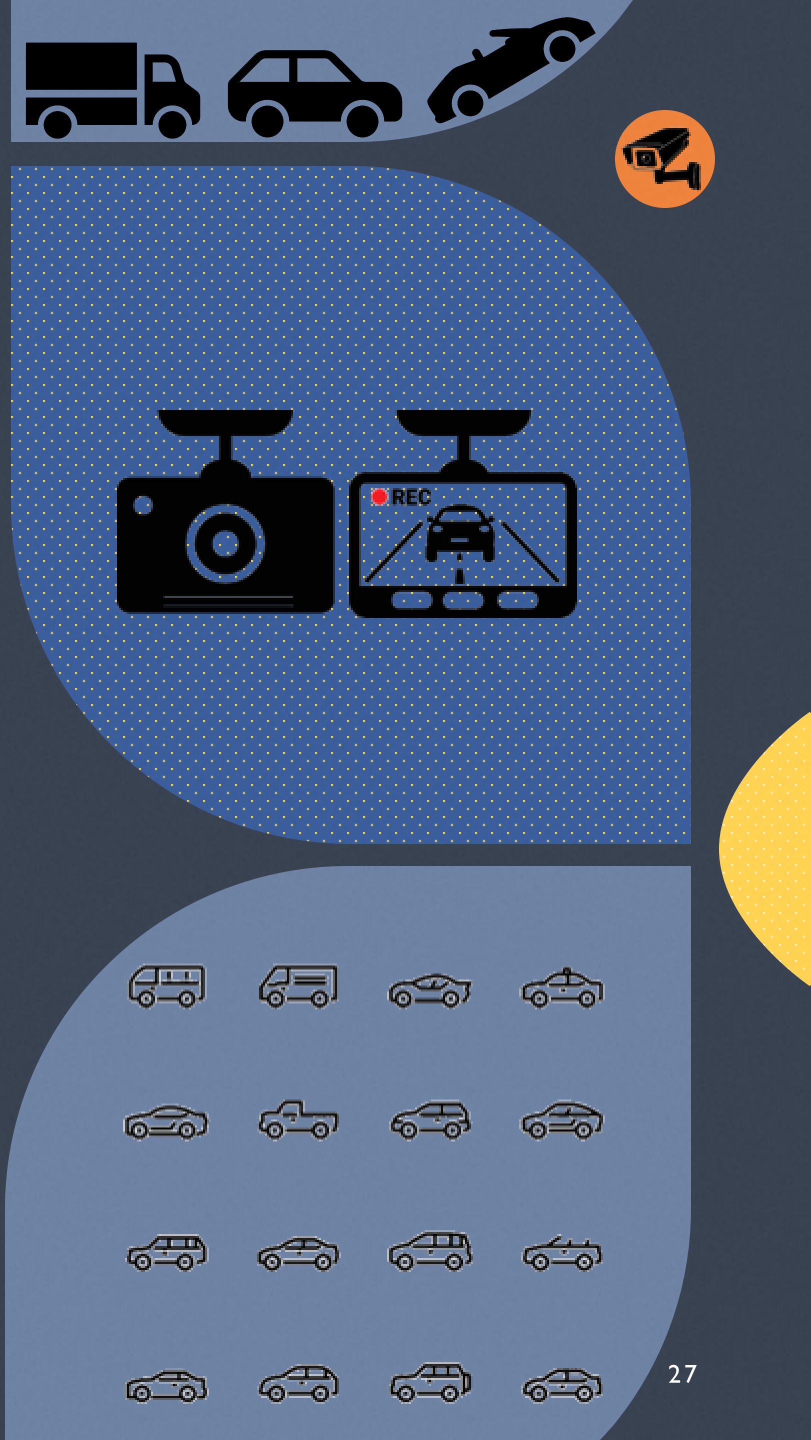
- Our pipeline uses AWS S3 for efficient storage and management of Lyft dashcam data, enabling seamless access for model training and evaluation.
- MLFlow allows us to log parameters, track model metrics, and artifact model versions while automating the training process
  - Continuously track and manage model versions iterate through our modeling pipeline to make better object detection and car class predictions
  - Deploy models as REST API's or with Docker containers to integrate cloud services
- Evidently AI allows us to track model performance and schedule retrains to ensure better performance as we get more Lyft dashcam data
  - Track any distribution shifts that may occur in the test data that could change model performance





References

- Yolo: https://arxiv.org/pdf/1506.02640
- Ultralytics: https://docs.ultralytics.com/usage/python/
- EfficientNet: https://arxiv.org/pdf/2104.00298
- Faster RCNN: https://arxiv.org/pdf/1506.01497
- EfficientNet Fine Tuning: https://keras.io/examples/vision/image_classification_efficientnet_fine_tuning/
- RoboFlow: https://docs.roboflow.com/



Questions?

For any follow-ups, please email:

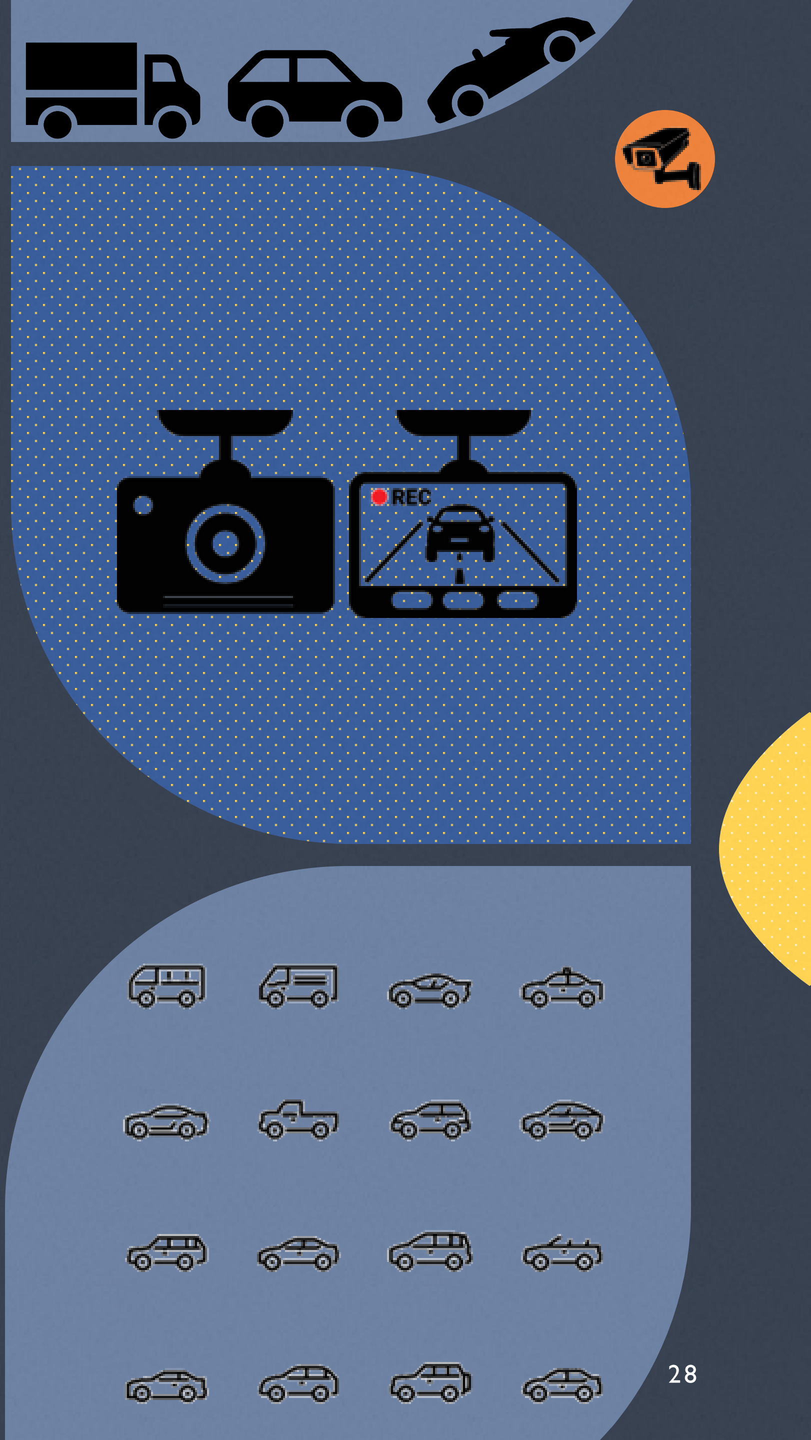
Apoorv Anand – apanand@uchicago.edu

Sivaram Mandava - sivaramm@uchicago.edu

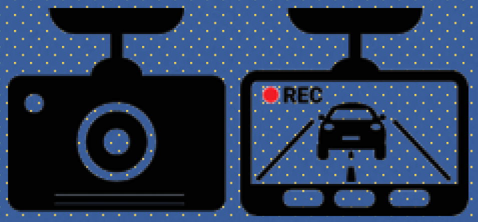
Dylan Versprille - dversprille@uchicago.edu

Austen Steinberg - austens@uchicago.edu

Michael Becker - mbbecker33@uchicago.edu



Appendix



Abstract

- Problem:
  - Vehicle manufacturers can gain localized market insights through AI driven vehicle identification and classification.
- Approach:
  - Fine-tune CNN models for vehicle body type identification.
  - Use Faster R-CNN and YOLO for vehicle detection and classification.
  - Train YOLO with manually labeled images for improved accuracy.
- Findings:
  - Both YOLO and region-based CNNs are effective for vehicle detection.
  - YOLO offers fewer false positives and performs well even with smaller datasets.
  - Transfer learning struggles with varying camera angles and partial images.
  - Overall YOLO performs better for the final Business Problem