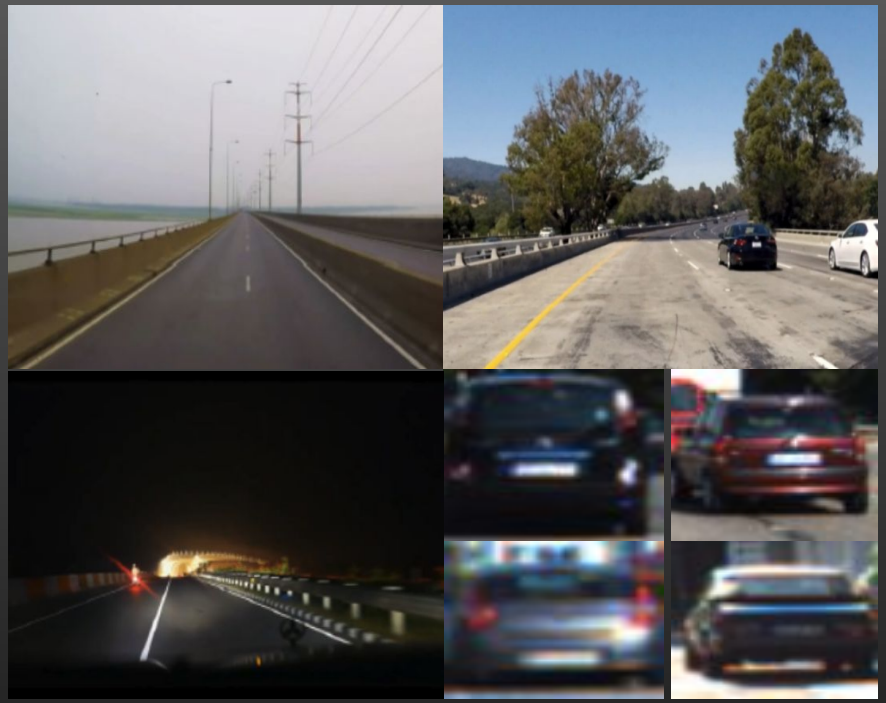
Lane and Vehicle Detection

CMPT 733 Project

–Jingwen Che, Haoran Ding, Chen Qiao, Dongxu Huang

Overview

- Design the deep learning neural network models for the Lane and Vehicle detection
- Train the models on the images of the real vehicles and road Dataset and then test it on the frames of the video taken on highway
- Compare the results with some other different models



Our output

The images that the Lane and vehicles are correctly annotated.

Models - Lane Detection

Model	Accuracy (%)	Dice Coefficient (%)	IoU (%)	Dice Loss(%)	Number of Parameters (Million)	File Size (Mb)
PSPNet	96.32	96.96	97.33	2.03	0.70	8.3
LLDNet	96.36	97.81	97.71	1.79	0.26	1.8
LLDNet with deeper ResBlock	96.40	97.93	98.09	1.61	0.31	4.1
LLDNet simplified version	96.23	97.23	96.71	2.35	0.076	1.3

Models - Vehicle Detection

- YOLOv7
- Transfer Learning of YOLOv7
- SVM

Model (100 epoch)	Model size	Precision	Recall	mAP@.5	mAP@.5:.95
yolov7-tiny_small	208 L, 6 million	0.806	0.642	0.713	0.416
yolov7_small	314 L, 36 million	0.819	0.708	0.745	0.472
yolov7-tiny_full	208 L, 6 million	0.821	0.781	0.823	0.525

Test dataset: Udacity Self Driving Car Dataset: 3000 test images

Dataset

- Training data:
  - Lane Detection
    - Udacity Machine Learning Nanodegree Project Dataset
    - Cracks and Potholes in Road Images Dataset
  - Vehicle Detection
    - GTI vehicle image database (SVM)
    - COCO 2017 Dataset (YOLO_v7, 143575 images)
    - Udacity Self Driving Dataset (YOLO_v7 Transfer, 15000 images)

Dataset

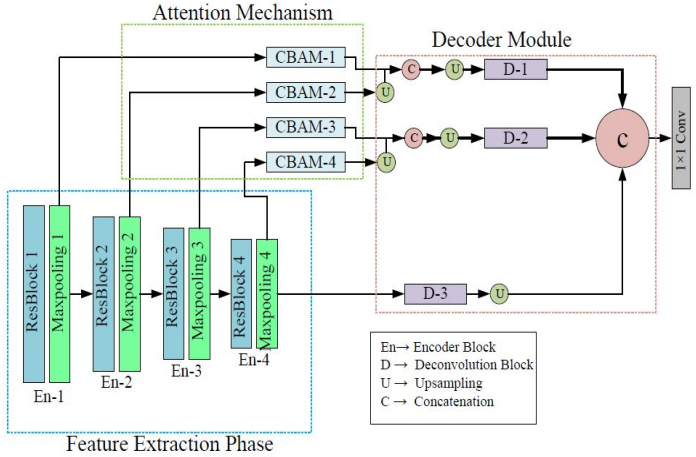
- Testing data:
  - Lane Detection
    - Udacity Machine Learning Nanodegree Project Dataset
    - Cracks and Potholes in Road Images Dataset
  - Vehicle Detection
    - Udacity Self Driving Dataset
  - Pipeline
    - Frames extracted from a video taken on highway

Data Preprocessing - Lane Detection

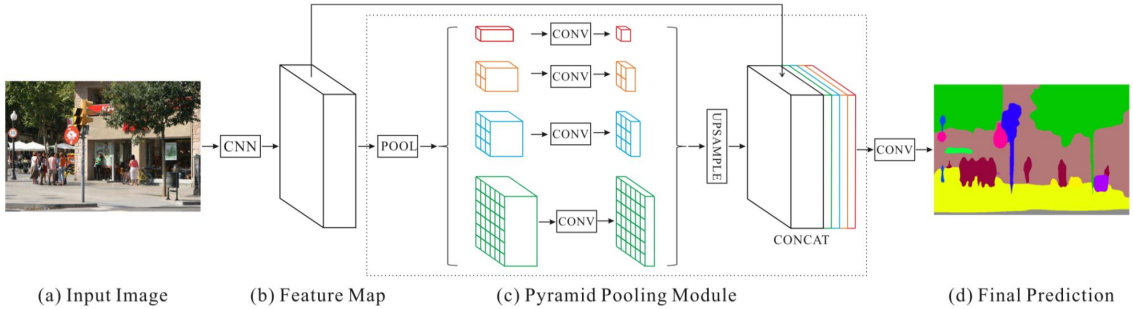
- PSPNet, LLDNet, and the two models we presented
  - Convert Images into NumPy files
  - Resize all the images into the same size
  - Merged the files for generating a mixed dataset

Network architectures - Lane

- **PSPNet**
- **LLDNet**
- **LLDNet with deeper ResBlock**
- **LLDNet simplified version**



LLDNet Architecture



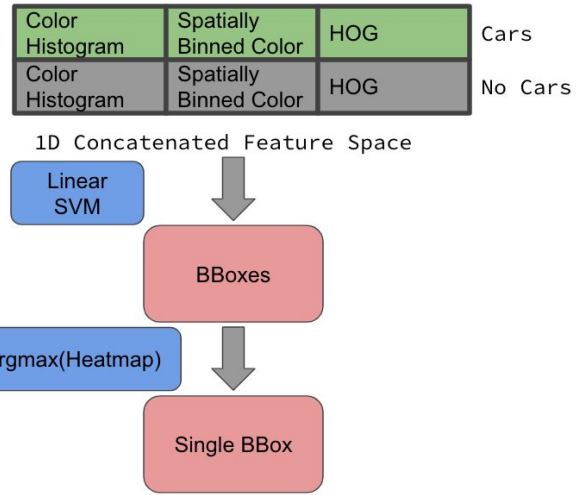
PSPNet Architecture

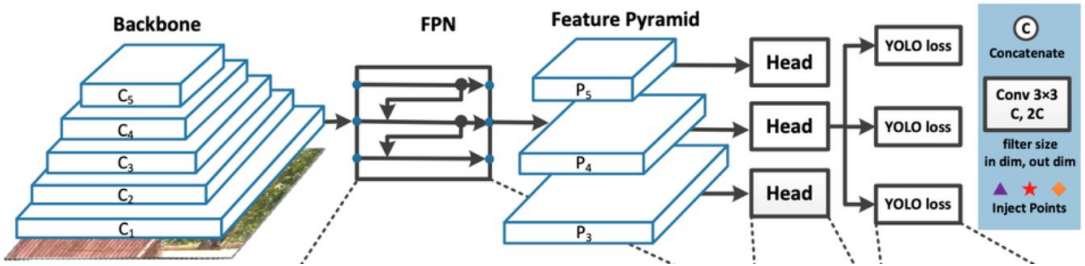
Data Preprocessing - Vehicle Detection

- Linear SVM
  - Image normalization
  - Y-crop: remove sky and hood area
  - Convert to YCrCb
- YOLOv7
  - Convert to RGB
  - Random HSV
  - Random Perspective + Rotation + Mosaic

Network architectures - Vehicle

Linear SVM





Similar YOLO architecture from PP-YOLO

- **Linear SVM**
  - Small
- **YOLO v7**
  - E-ELAN Layer Aggregation
  - Fast (155 FPS)
  - Accurate
  - High Res (608)

Transfer Learning - YOLOv7 - Vehicle Detection

- Transfer Learning of YOLOv7
  - Pretrained weights: [YOLOv7, YOLOv7-tiny] on COCO 2017, 80 classes
  - Transfer to the dataset focused on vehicle detection
    - Two label classes: car, truck ← from 11 classes
  - Two different data size: [15000 full set, 1500 small set]
  - 100 epochs, 640x640, same augmentation

Lane Detection Result

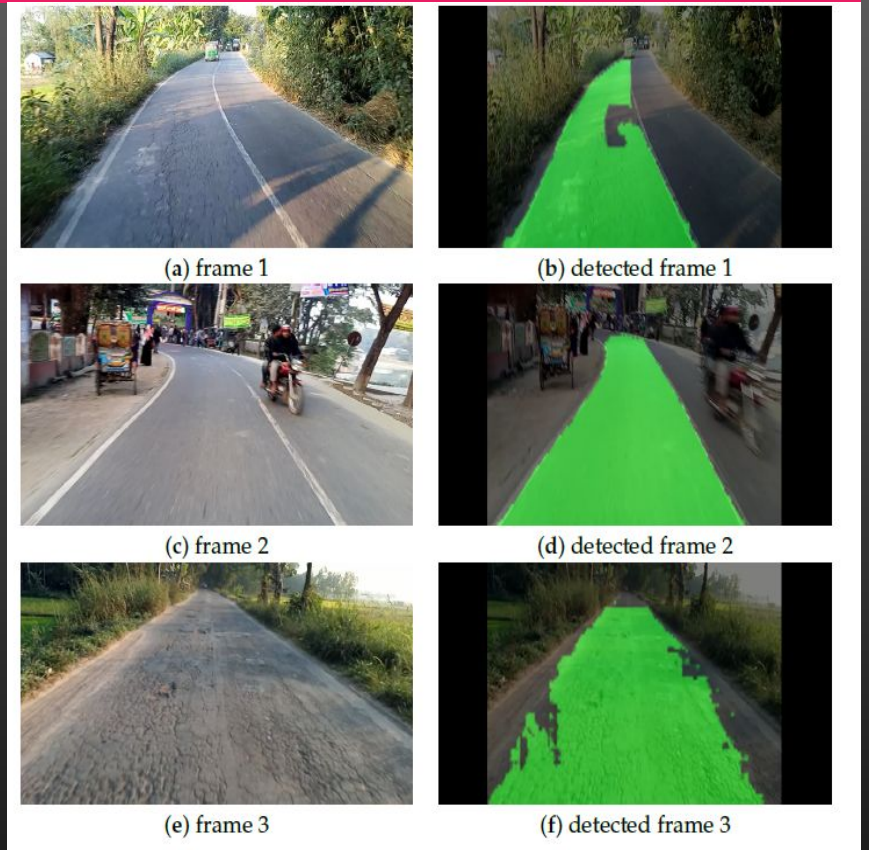
- PSPNet

  Test results

Lane Detection Result

- LLDNet

  Test results



Lane Detection Result

- LLDNet (deeper ResBlocks)

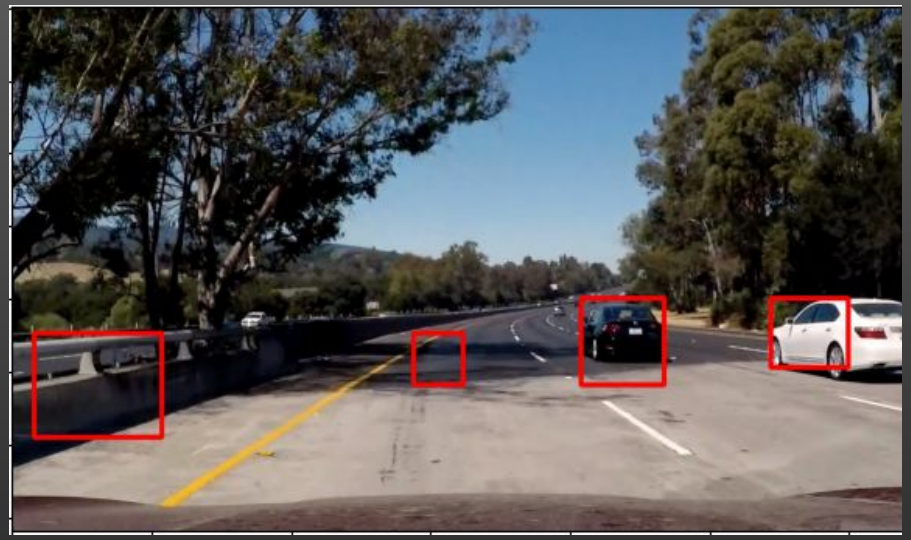
  Test results

Lane Detection Result

- LLDNet (simplified ResBlocks)

  Test results

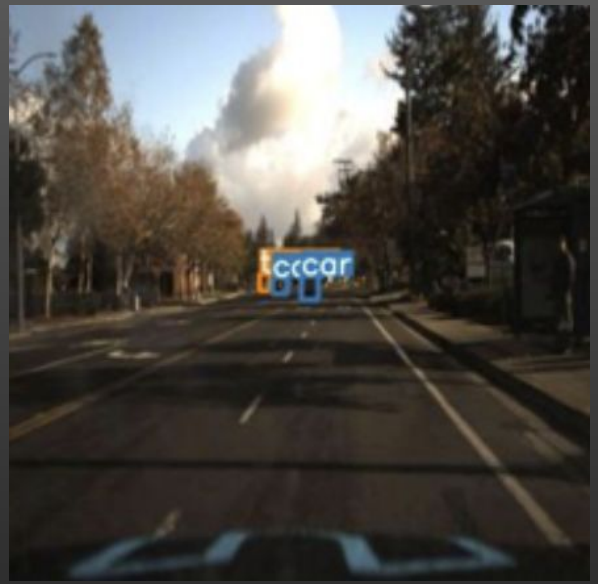
Vehicle Detection Result



SVM Result

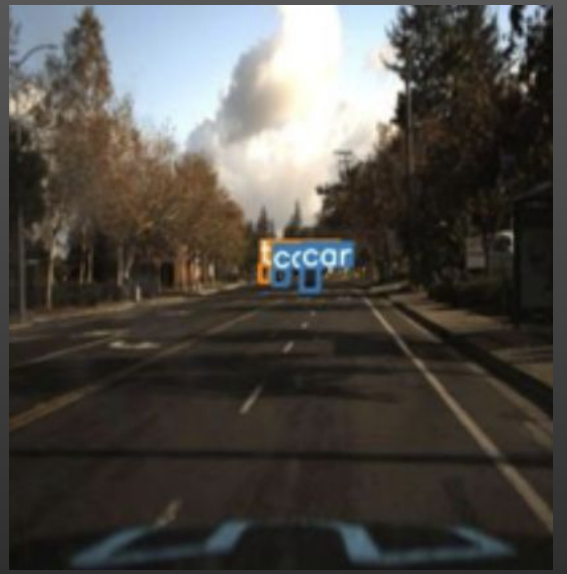
Best Transfer Model: v7-tiny, 15000 images

Vehicle Detection Result



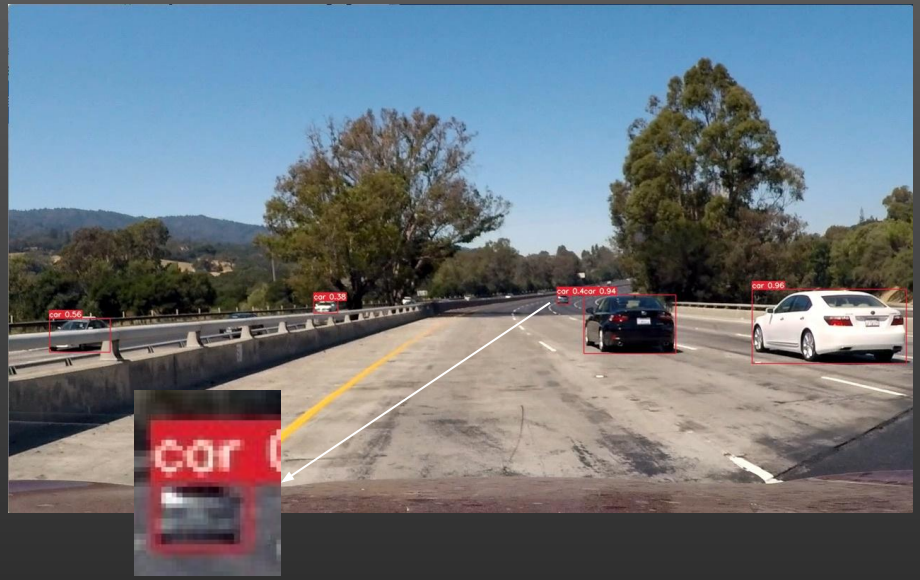
Ground Truth

YOLOv7 original

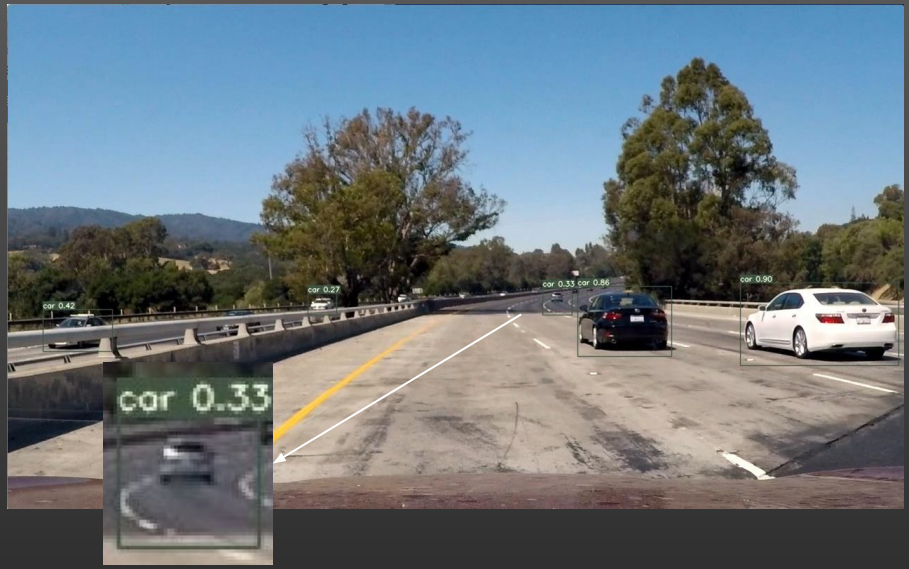


Transfer YOLOv7-tiny

Vehicle Detection Result

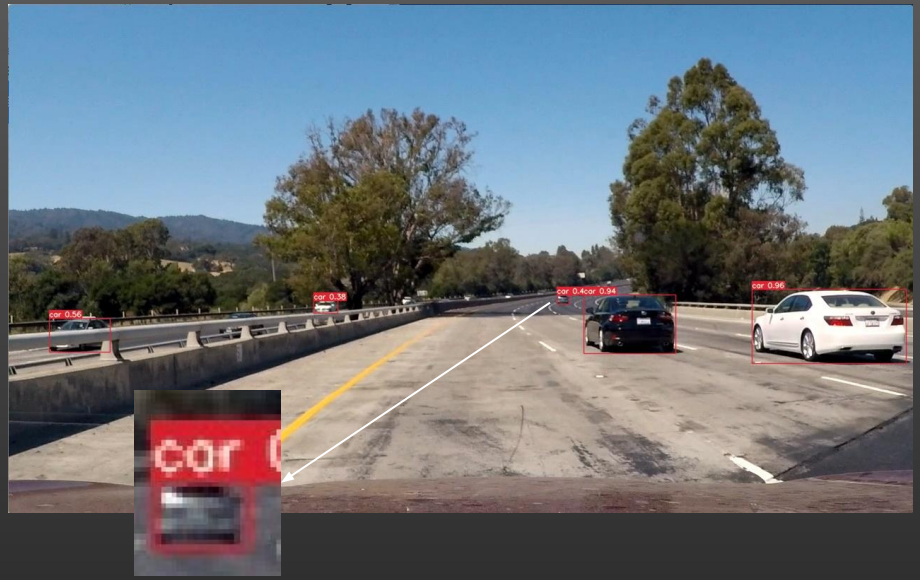


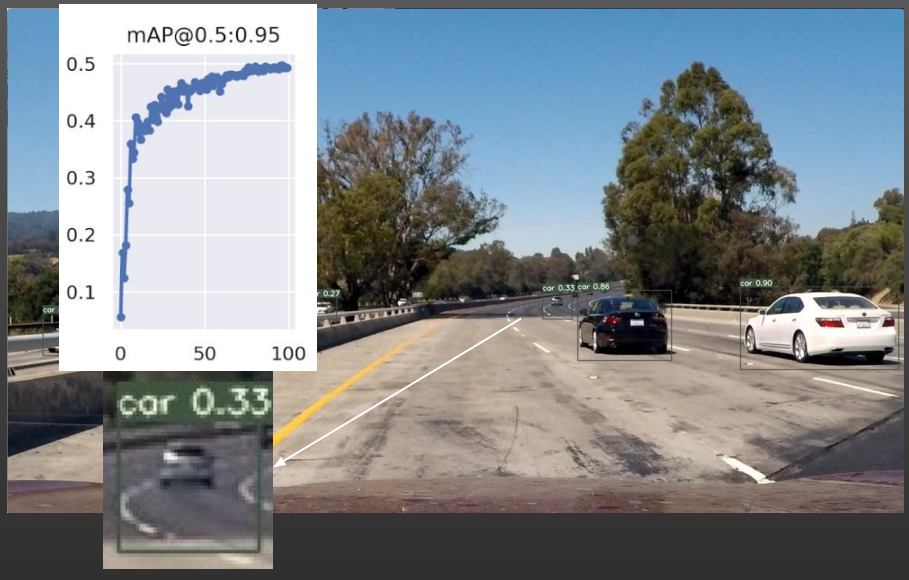
Model: YOLOv7 original, COCO 143575 images



Best Transfer Model: v7-tiny, 15000 images

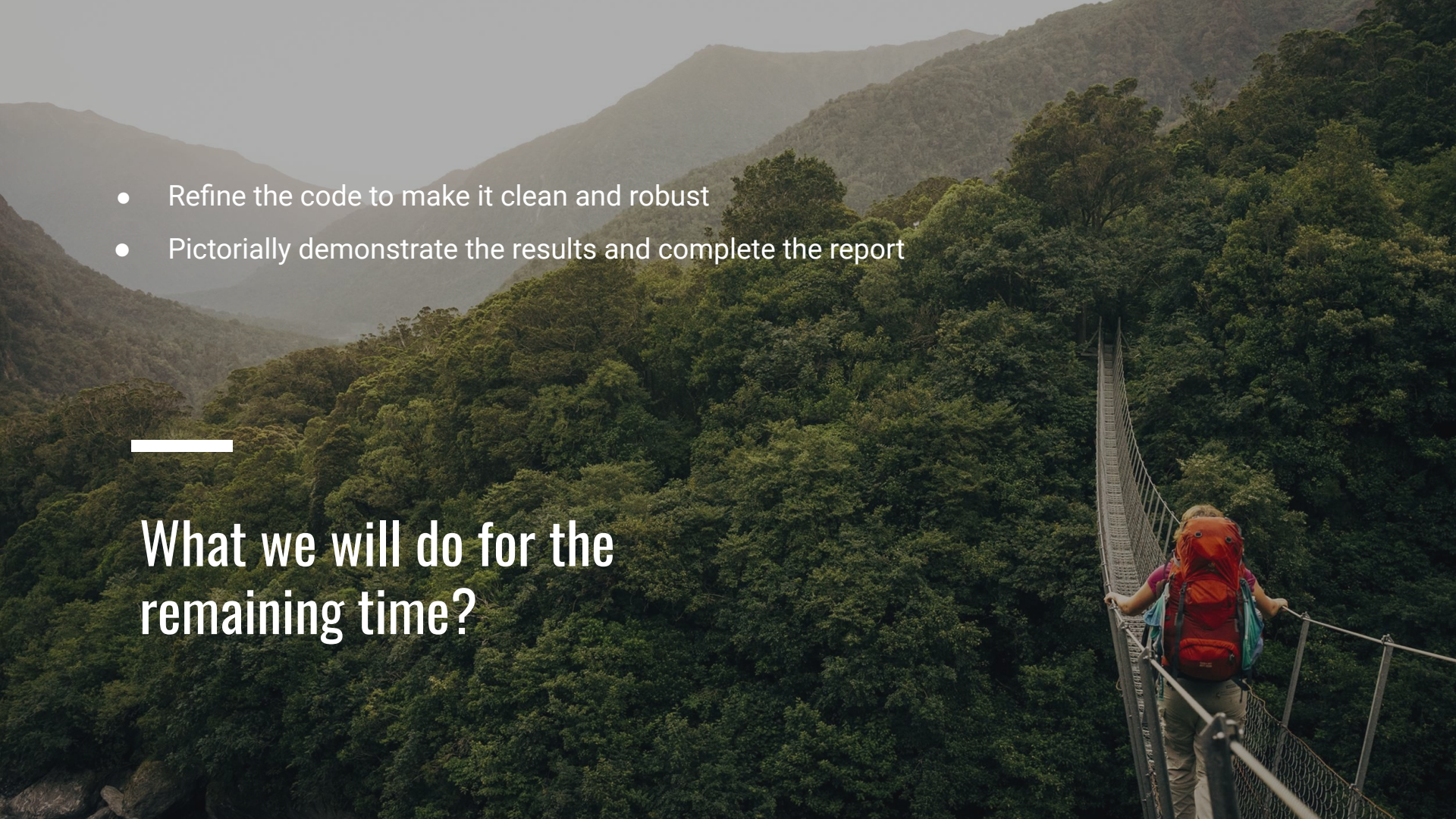
Vehicle Detection Result





Model: YOLOv7 original, COCO 143575 images

Best Transfer Model: v7-tiny, 15000 images



- Refine the code to make it clean and robust
- Pictorially demonstrate the results and complete the report

What we will do for the remaining time?

Thank you!