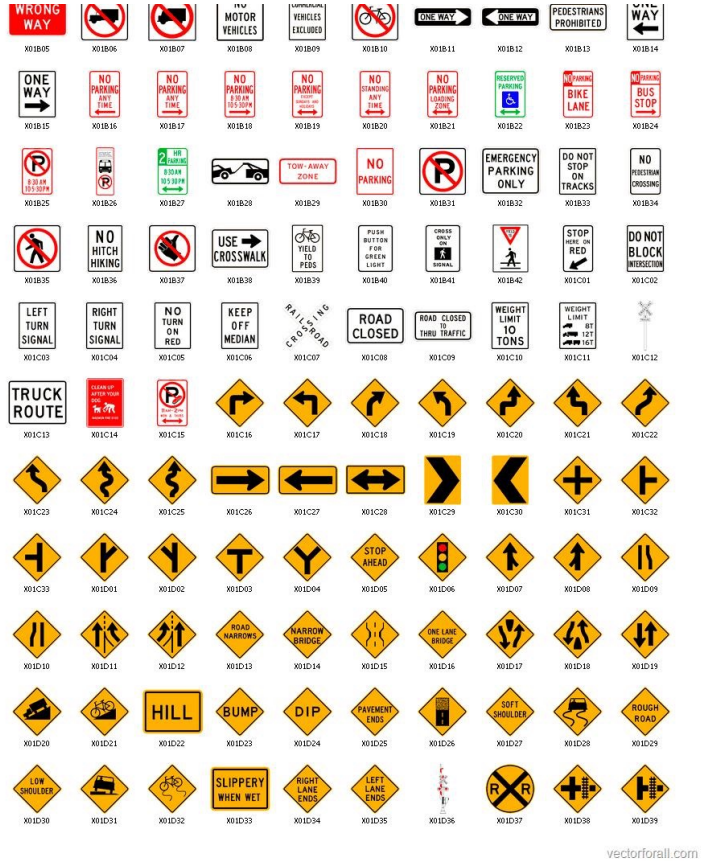
Object Detector for Road Signs

By Ka Hung Lee

Motivation



- Roads and motorways are essential infrastructure that physically connects places together
- They provide ease of assess for vehicles to travel
- Road signs or traffic signs are constructed to guide or provide instructions for road users

Where is that sign?

- However, there could be times when driver wish to:
  - Locate/Identify specific road signs at ease
  - Be made aware of upcoming road signs
- To solve this problem, we could create a program that could detect specific road signs

Project Plan

1. Collect data
2. Analyze Data
3. Preprocess Data
4. Model Data
5. Evaluate Model
6. Test Model

Project Plan

1. Collect Data
2. Analyze Data
3. Preprocess Data
4. Model Data
5. Evaluate Model
6. Test Model

Original Dataset

- Source: https://www.kaggle.com/andrewmvd/road-sign-detection
- 877 images with 4 classes (trafficlight, stop, speedlimit, crosswalk)

Number of Labels

- Original dataset

			
Label Name			
trafficlight	stop	speedlimit	crosswalk
Total			
170 labels	91 labels	783 labels	200 labels

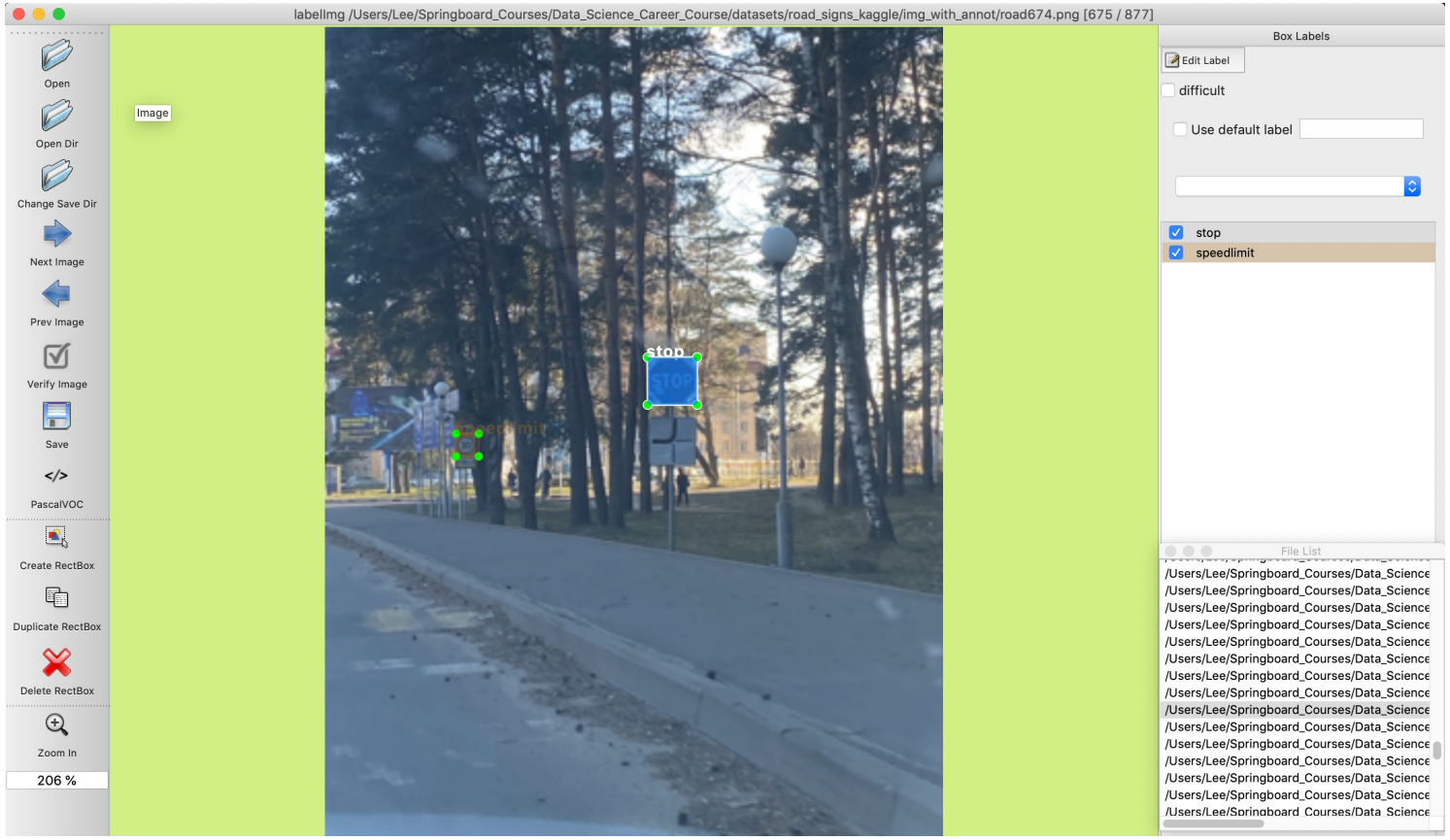
Project Plan

1. Collect Data
2. Analyze Data
3. Preprocess Data
4. Model Data
5. Evaluate Model
6. Test Model

Preprocessing Data

- Although the dataset already came with “ground-truths”, they need to be verified
- All labels/annotations were checked using LabelImg

LabelImg



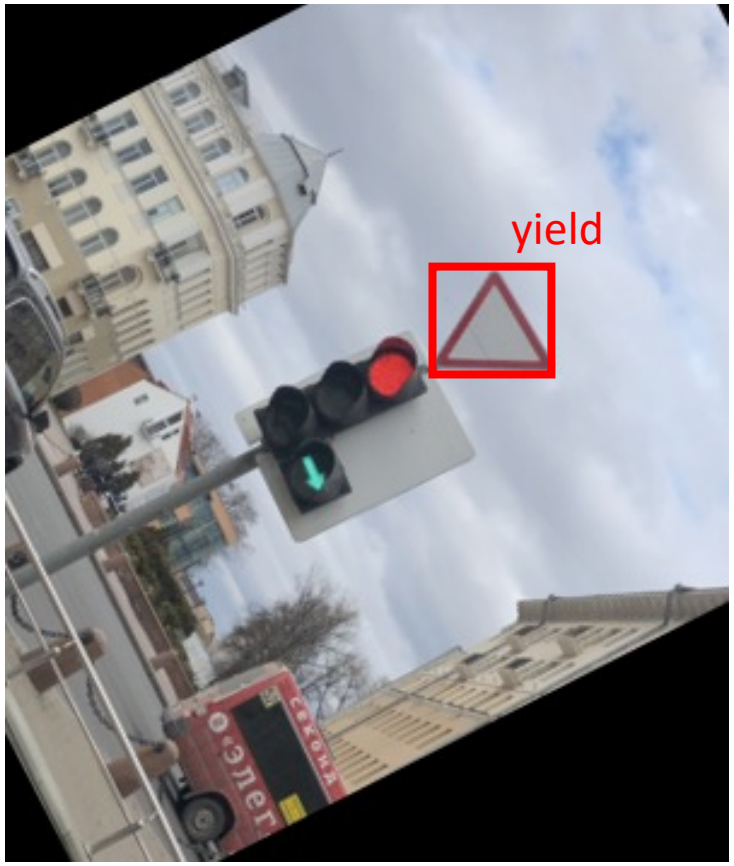
- Labeling Tool used: https://github.com/tzutalin/labelImg
- Can be used to add/modify existing labels/annotations
- Existing labels were modified with 2 additional classes using LabelImg (nostop, yield)

Number of Labels

- Dataset after adding new labels (nostop and yield) and modifying old labels

					
Label Name					
trafficlight	stop	speedlimit	crosswalk	nostop	yield
Total					
155 labels	93 labels	788 labels	218 labels	107 labels	15 labels
-15	+2	+5	+18	+107 (new)	+15 (new)

Additional Data and Augmentation



- 100 images were added through image augmentation to increase the sample size of yield
  - Rotation
  - image blur
  - noise

Number of Labels

- Dataset w/augmented images

					
Label Name					
trafficlight	stop	speedlimit	crosswalk	nostop	yield
Total					
176 labels	93 labels	862 labels	298 labels	107 labels	111 labels
+21	+0	+74	+80	+0	+96

Number of Labels

- The dataset was partitioned at random roughly 9:1 (training/validation split)
- tfrecord files were created for model training and evaluation

					
Label Name					
trafficlight	stop	speedlimit	crosswalk	nostop	yield
Training Size					
162 labels	77 labels	780 labels	263 labels	97 labels	94 labels
Validation Size					
14 labels	16 labels	82 labels	35 labels	10 labels	17 labels

Project Plan

1. Collect Data
2. Analyze Data
3. Preprocess Data
4. Model Data
5. Evaluate Model
6. Test Model

What this entails?

1. Object Classification

Classification: Traffic Light

What this entails?

1. Object Classification
2. Object Localization

Classification

+

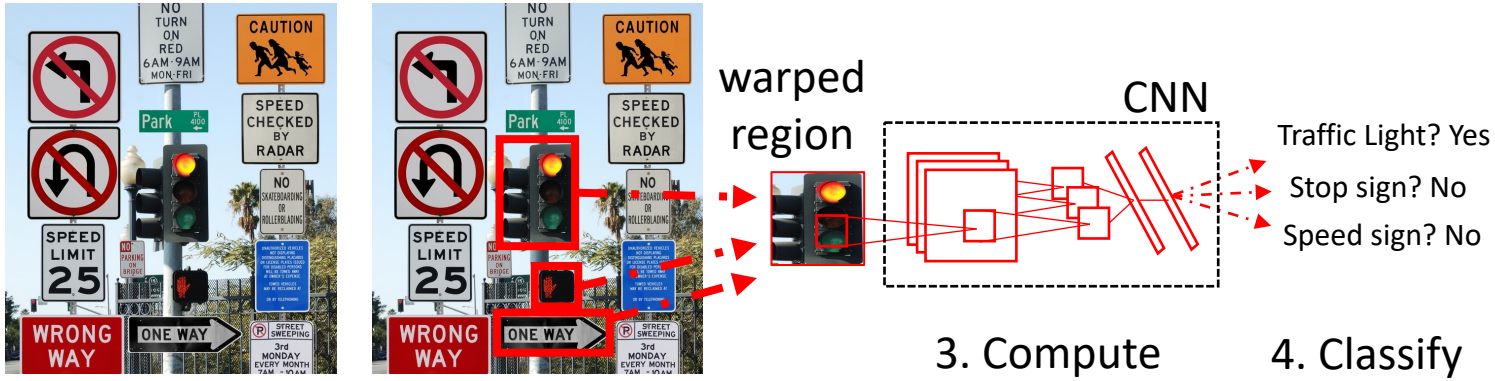
Localization

=

Detection!

Classification: Traffic Light

Common Approaches



- Region-based Convolutional Neural Networks (R-CNNs)
  - Pros:
    - Accurate
  - Cons:
    - External search algorithm is a performance bottleneck
    - Slow training time
    - Memory intensive
    - Slow detection time

Common Approaches

1. Input Image

CNN

2. Compute CNN Features

Feature map of image

Region Proposal Network

CNN

Proposals

RoI Pooling Layer

CNN

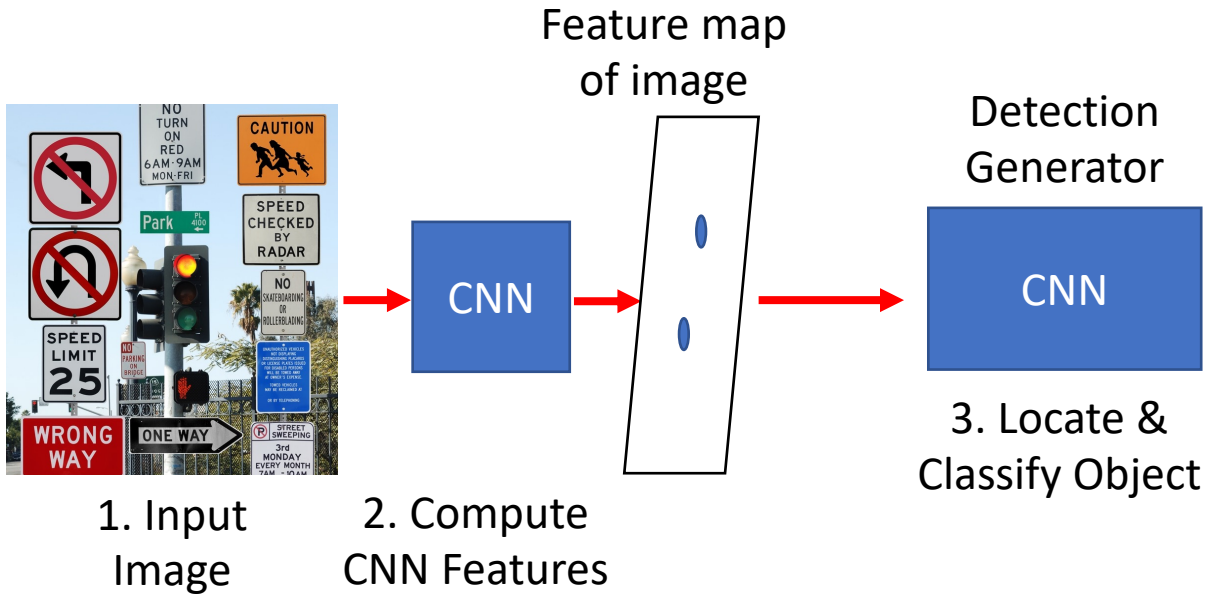
Object Classifier

3. Create Proposed Regions

4. Classify Regions

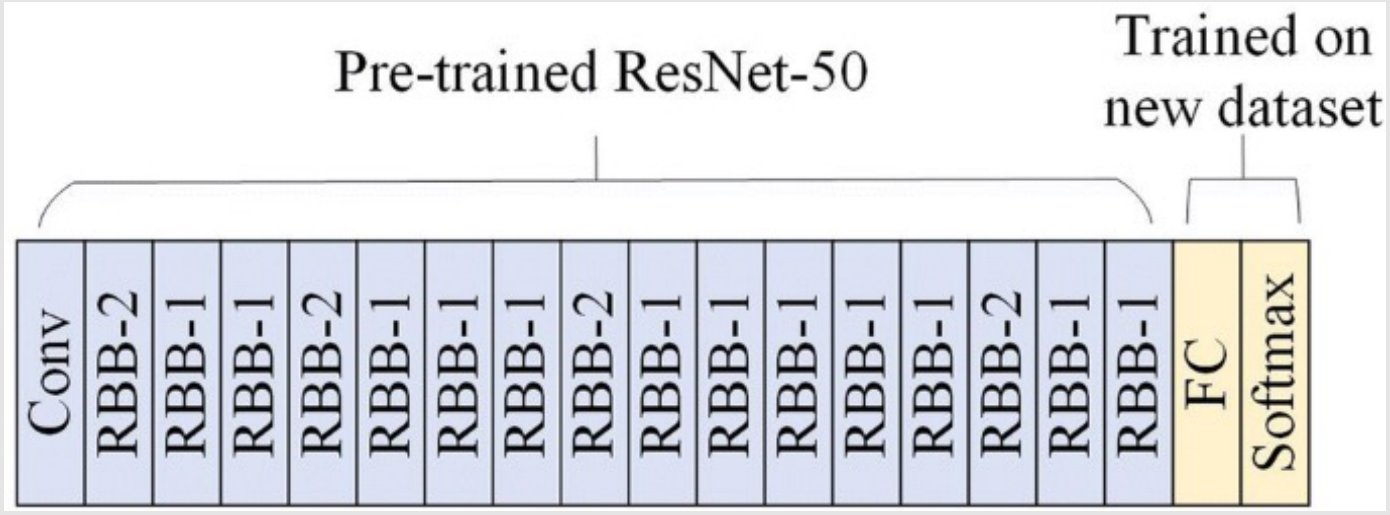
- "Faster" R-CNNs
  - Pros:
    - Reduced training time
    - Faster detection
    - Still accurate
  - Cons:
    - Run time depends on number of proposed region

Common Approaches



- Single-Shot Detector (SSD)
  - Pros:
    - Direct detection
    - Very fast detection
    - Designed for real-time
    - Reduced training time
  - Cons:
    - Accuracy drops for smaller objects

Our Model: SSD Approach

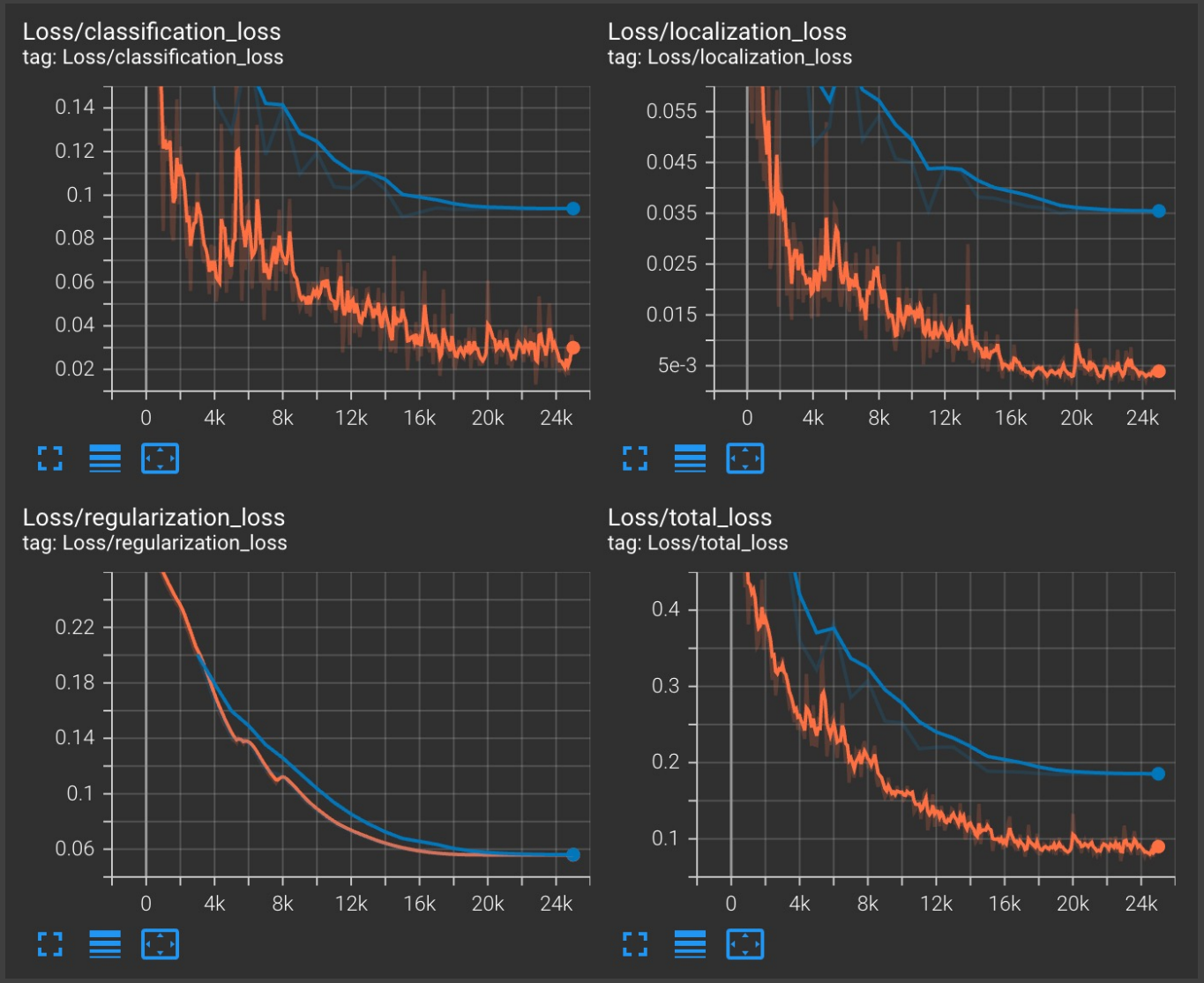


- Our model utilizes the saved weights of a pre-trained Retinanet Object Detection model (SSD with Resnet 50 v1) to train our customized model
- Utilizes technique known as transfer learning
- The model was trained for 25000 steps (2000 warm-up steps)
- Total training took roughly half a day with GPU from Google Colab Pro

Model Evaluation

Training scores: orange

Validation scores: blue



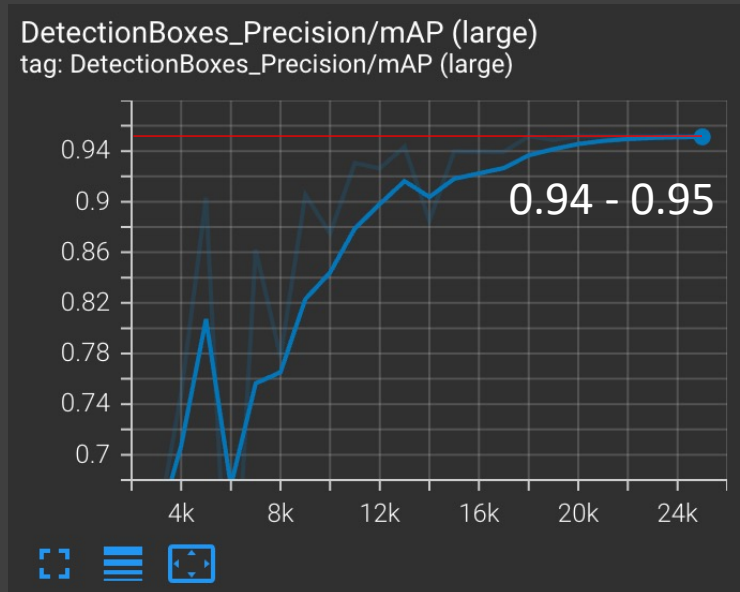
Validation Set

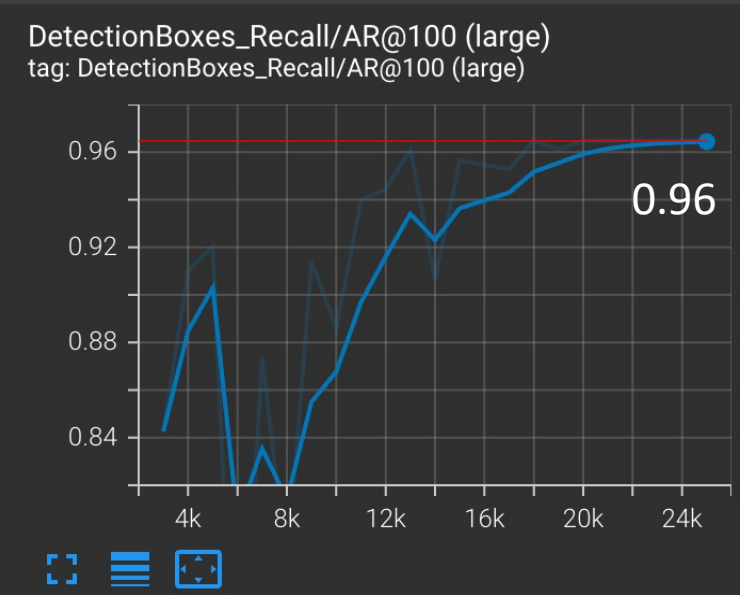
Model Performance (Validation Set)

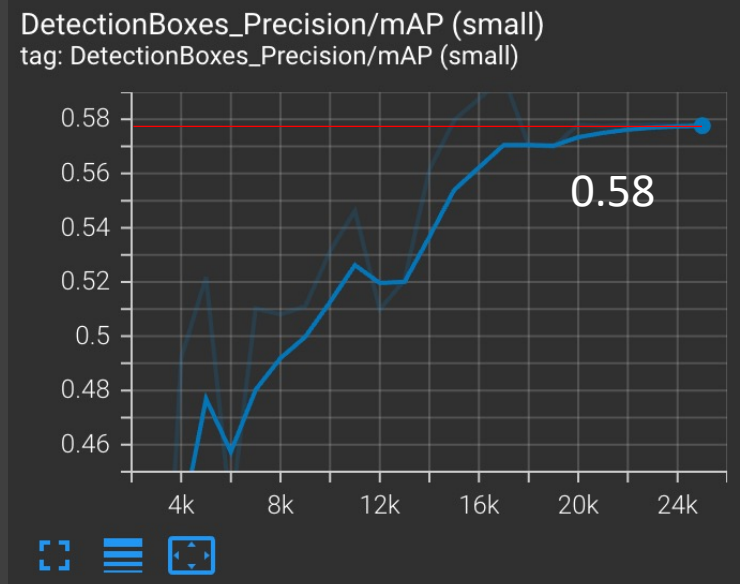
For static images, the model perform well for large objects, and less so for small objects

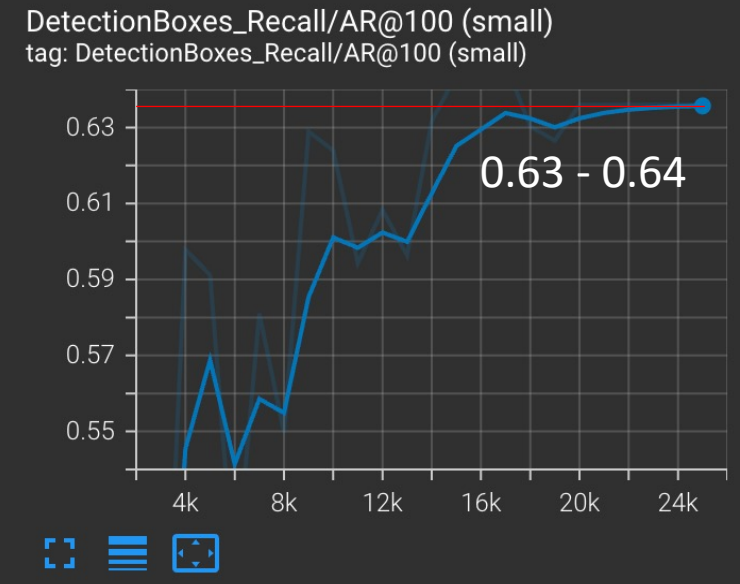
Precision

Recall





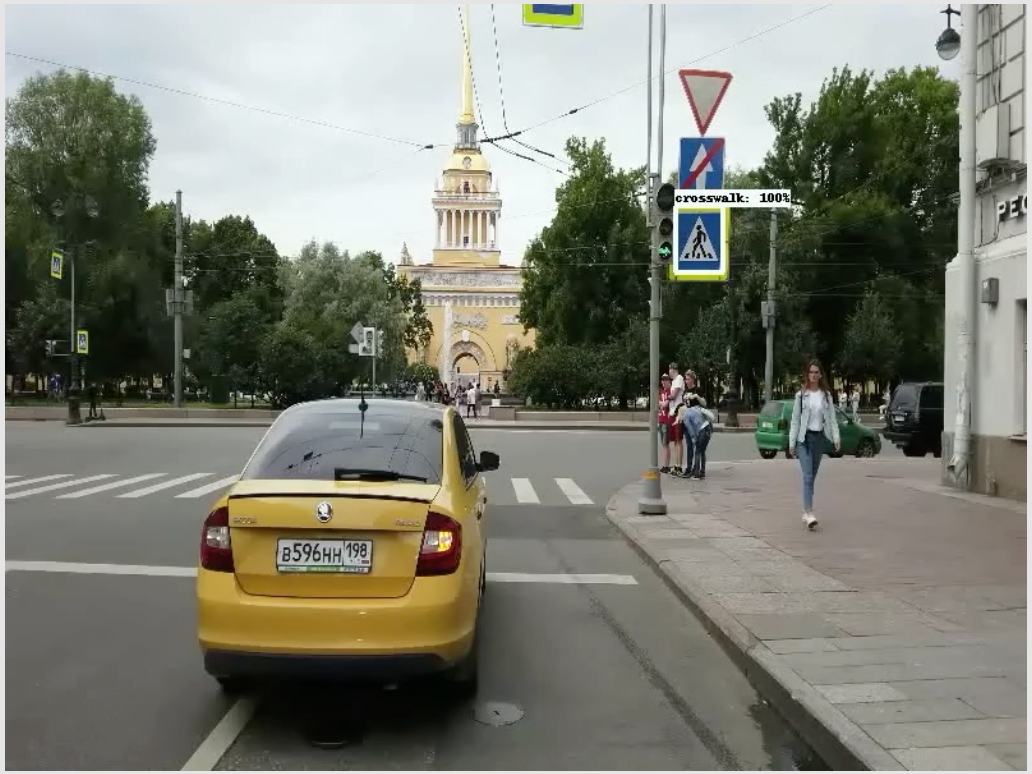




Project Plan

1. Collect Data
2. Analyze Data
3. Preprocess Data
4. Model Data
5. Evaluate Model
6. Test Model

Video Test and Conclusions



- The model was able to detect objects on video
- Apparent issues:
  - Detection for yield is spotty
  - Cars are mistaken for trafficlights
  - Certain road signs can only be detected close up

Potential Improvements

- Include more size variation for labels
- Include more images of labels in traffic settings
- Increased sample size

Thank you

- Questions?
- Capstone link: https://github.com/leekahung/object_detector_roadsigns
- Profile
  - LinkedIn: https://www.linkedin.com/in/ka-hung-lee-090634a7/
  - GitHub: https://github.com/leekahung
  - Google Scholar: https://scholar.google.com/citations?user=VfIHNYIAAAAJ&hl=en