Capstone Project

TheiaVision: Object Detection Technology for PMD Safety Alerts

Ng Wei GA-DSI-42 14-May-2024

I am from

Innovation and Design Team

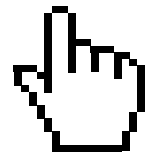
in MaximalSG (PMD Maker)

You are

Darren – Product Manager

in MaximalSG (PMD Maker)

Content



01

Problem Statement & Persona

Articles of PMD

Oct 2019	**"TTSH reports spike in injuries involving PMD riders"** In 2019, PMD accident rate increased **68%** from 2017 Source: The Strait Times
Nov 2019	**"E-Scooters to Be Prohibited on All Footpaths Following Safety Review"** Source: LTA
2020 - 2022	**"PMD-related offences decrease in last 3 years as e-scooter population dwindles further"** From 2020 to 2022, **65%** fall in PMD-related offences Source: The Strait Times
May 2022	**"PMD delivery rider dies after accident with motorcycle in Serangoon"** Source: The Strait Times

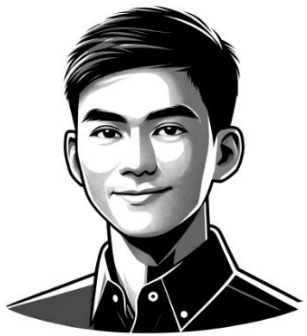
PMD Accident Footage

PMD Accident Footage

Problem Statement

How can we **enhance the safety of Personal Mobility Devices (PMDs)** in urban environments by using **object detection** to improve PMD users' ability to perceive and respond to their surroundings?

Persona



Darren
36
Product Manager
MaximalSG (PMD)

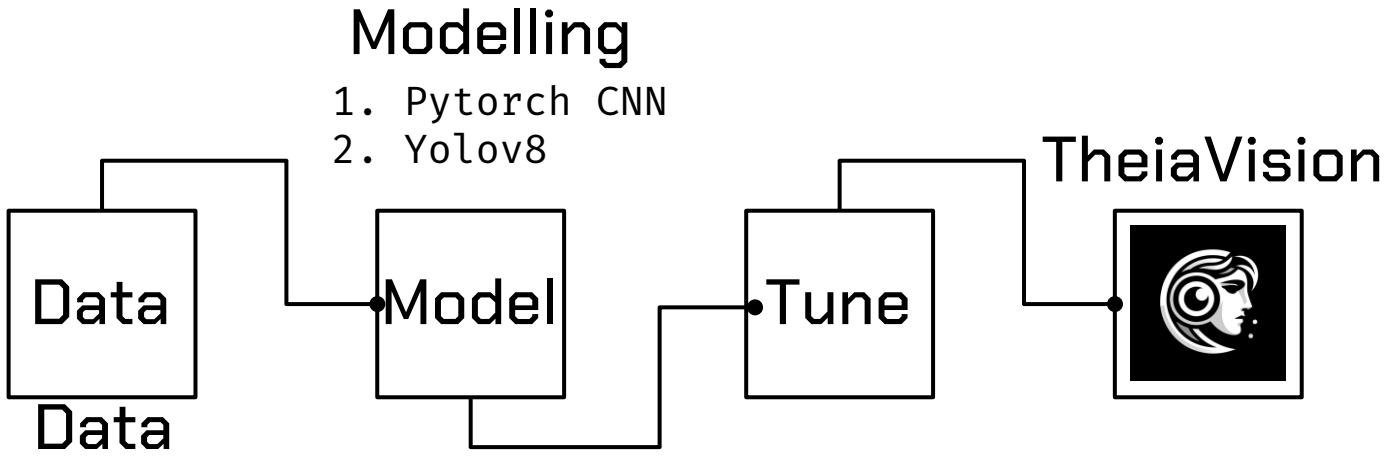
Darren is addressing the critical challenge of **enhancing PMD safety** due to increasing urban accidents. He is leading the development of an alert system with **object detection technology** that identifies obstacles such as **pedestrians, vehicles, and traffic signs**.

The Solution -TheiaVision

TheiaVision – An Eye of PMD that guide your way!

Theia - Greek goddess of sight and vision

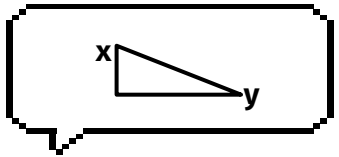
What is behind of TheiaVision?



02

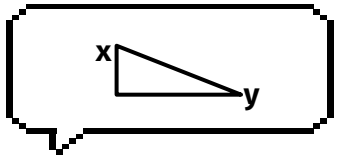
Dataset & Preprocessing

Image Dataset



	Udacity Self Driving Car	Singapore Traffic Sign
Images	15,000 images	851 images
Annotation	97,942 labels	Unlabelled
No. of Class	11	7
Resolution	512 x 512 Pixels	Not fixed Pixel, Low Resolution
Content	Car, pedestrian, truck and traffic lights	7 traffic signs related with PMD/Pedestrian
Sources	roboflow	
Readiness	Ready for modelling	Raw images

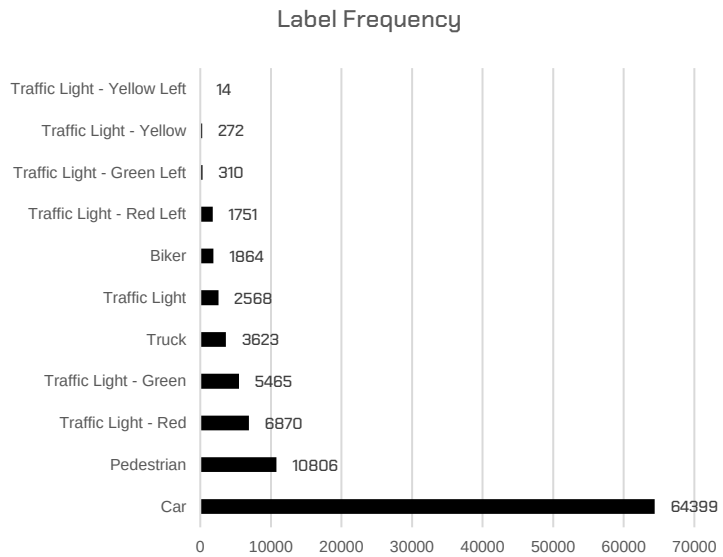
Image Dataset



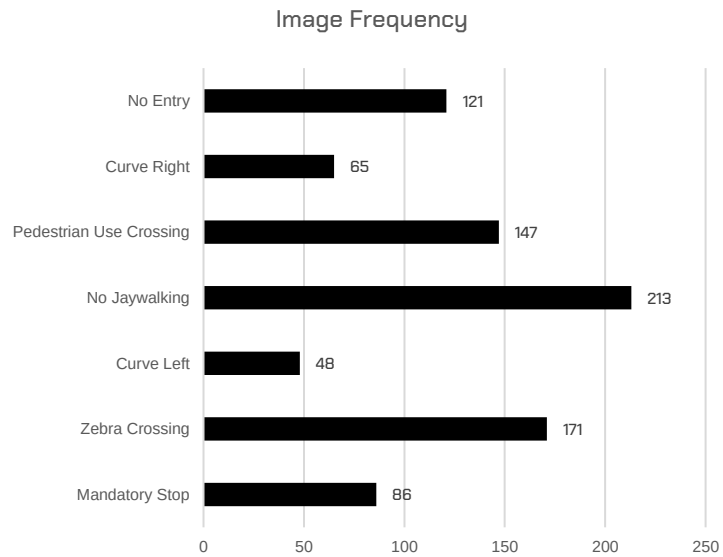
	Udacity Self Driving Car	Singapore Traffic Sign
Images	15,000 images	851 images
Annotation	97,942 labels	Unlabelled
No. of Class	11	7
Resolution	512 x 512 Pixels	Not fixed Pixel, Low Resolution
Content	Car, pedestrian, truck and traffic lights	7 traffic signs related with PMD/Pedestrian
Sources	roboflow	
Readiness	Ready for modelling	Raw images

Label/Image Proportion

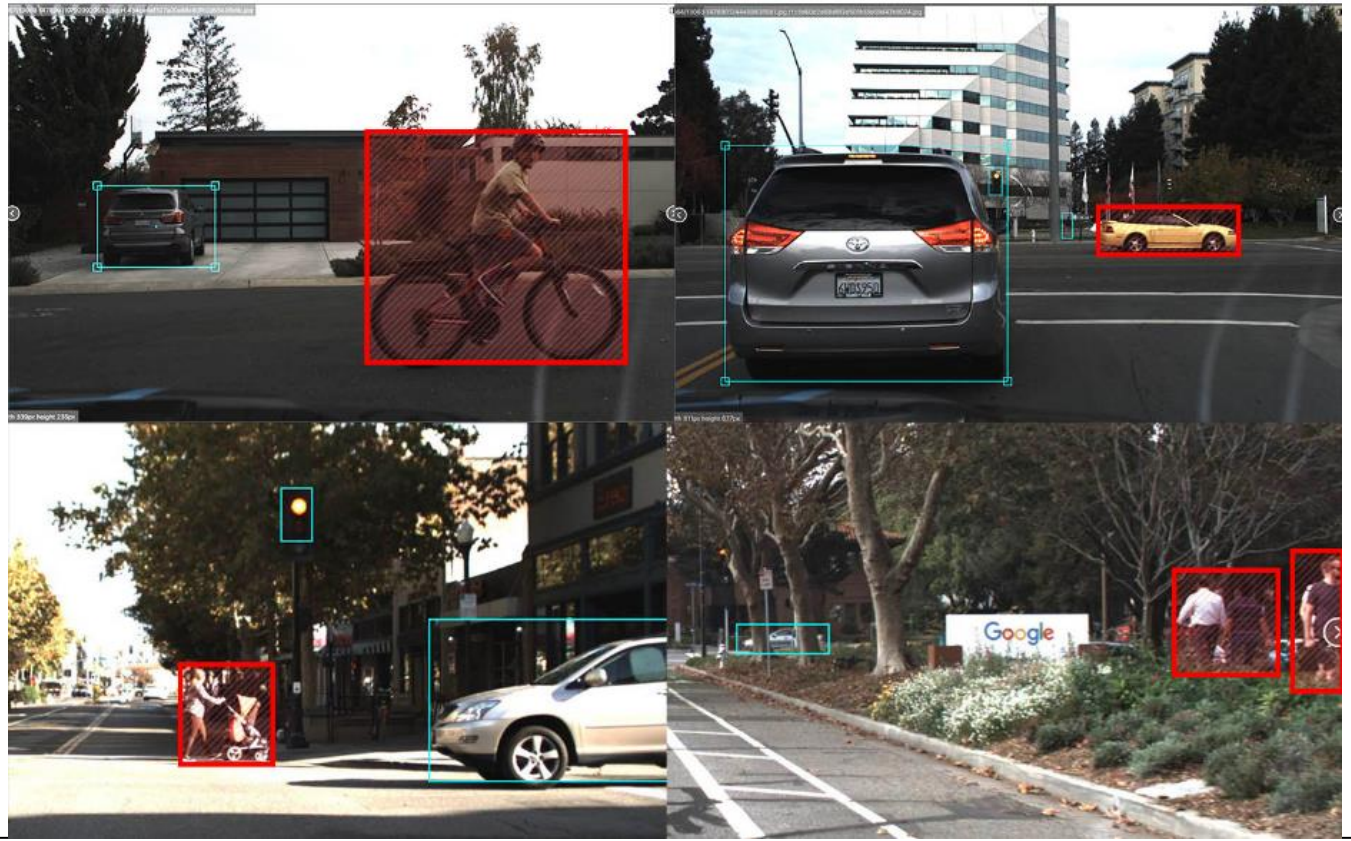
Udacity Self Driving Car



Singapore Traffic Sign



Udacity Self Driving Car Dataset



02.01

Singapore Traffic Sign

Shortlisted Traffic Sign for PMD

Priority Sign - Pedestrian Crossing

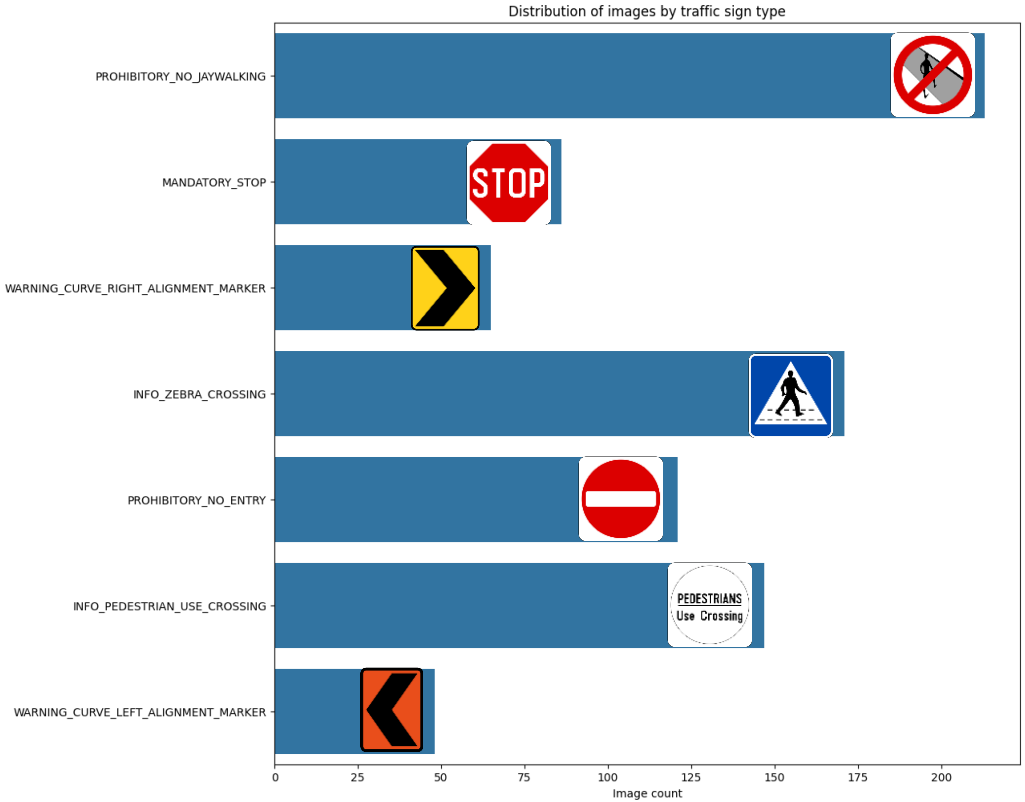
Prohibitory Sign - No Entry & No Jaywalking

Informatory Sign - Pedestrian Crossing

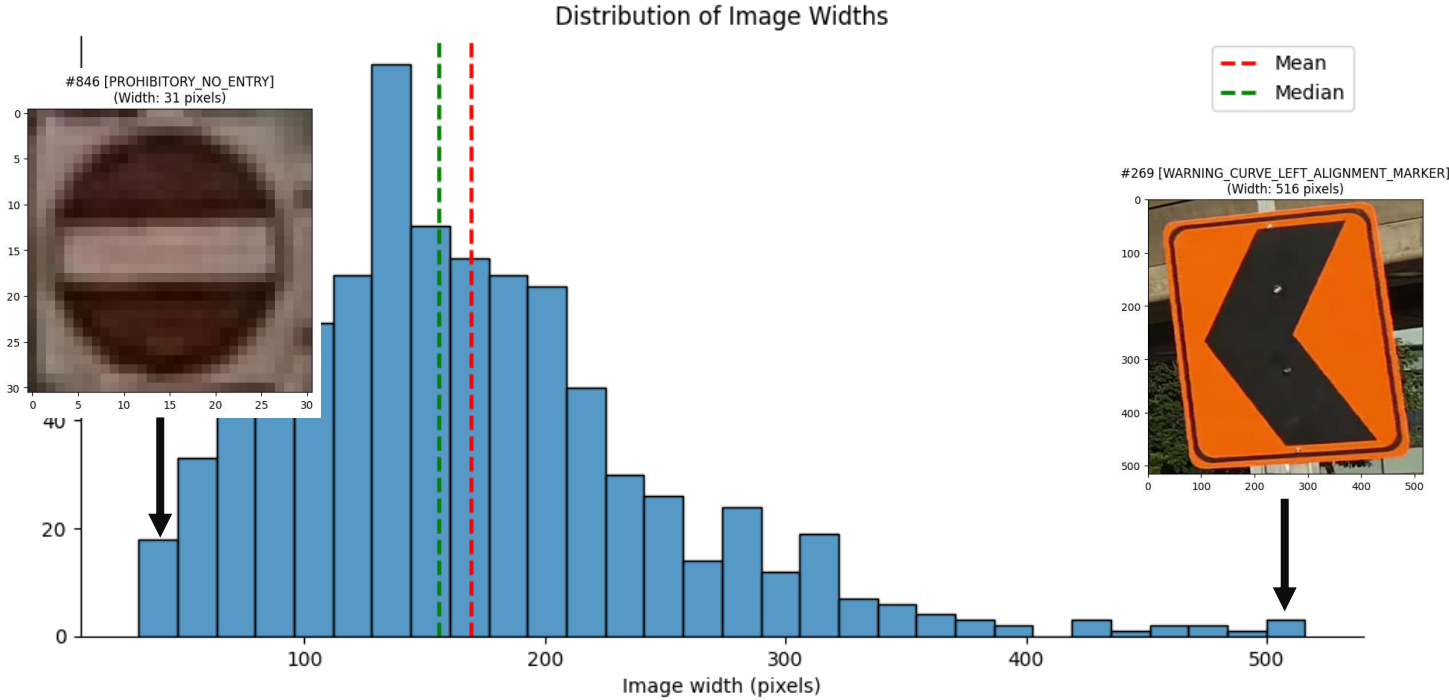
Mandatory Sign - Pedestrian Crossing

Temporary Left / Warning Curve Marker

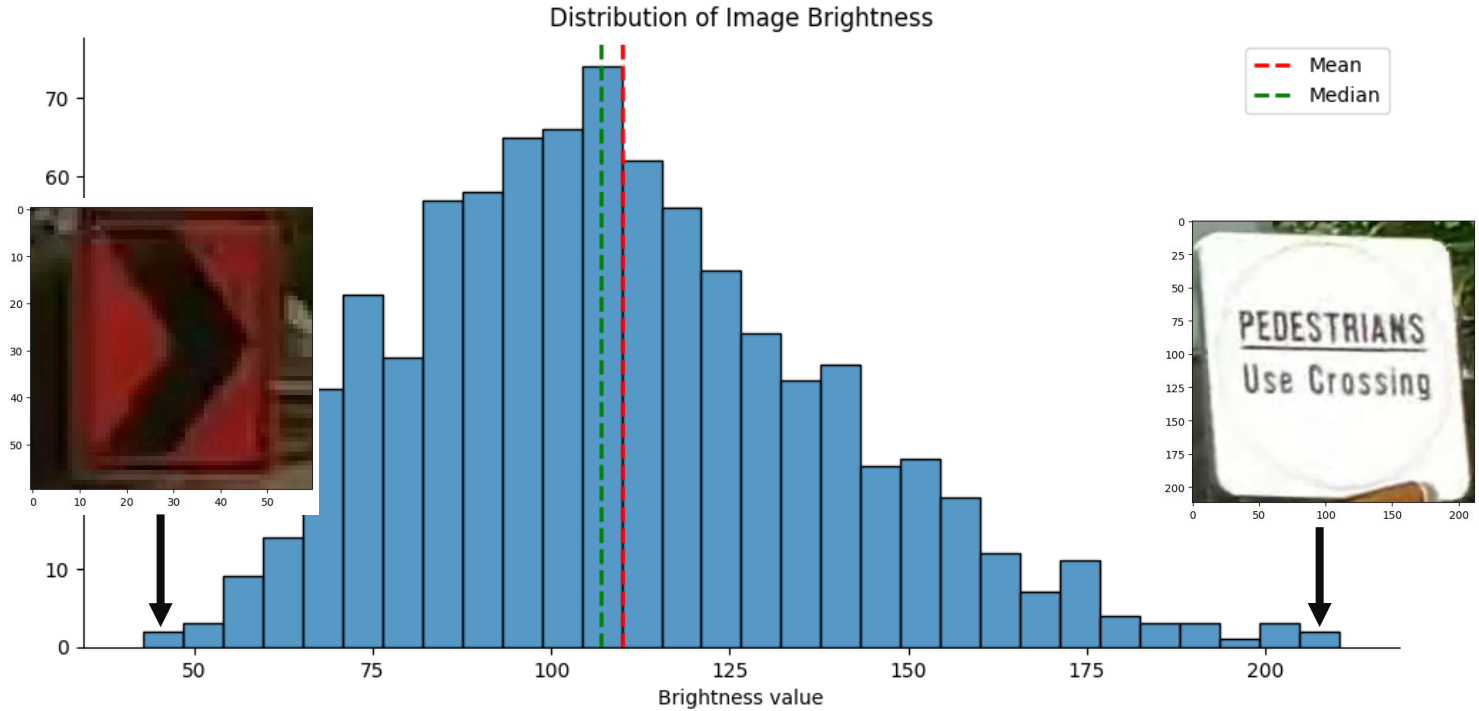
Proportion of Traffic Signs



Resolution of Traffic Sign Images



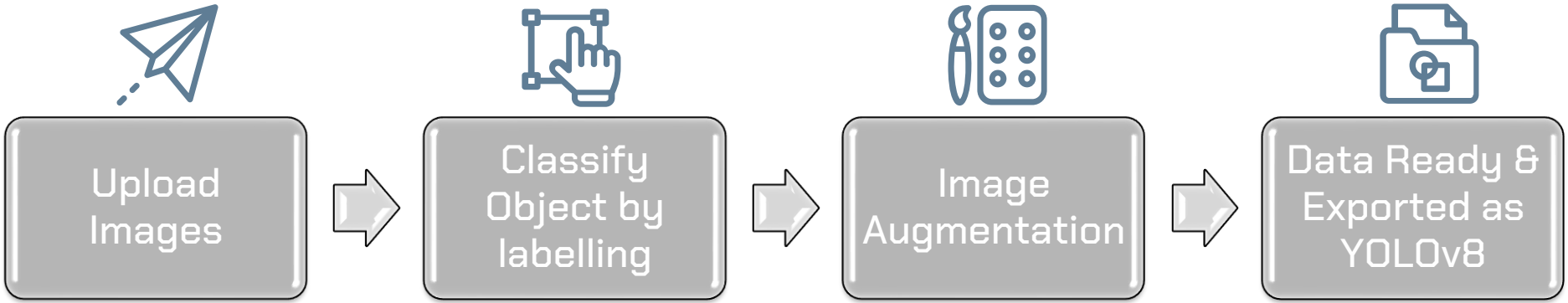
Brightness of Traffic Sign Images



Annotation of Unlabeled Traffic Image

- After data EDA, let's label our data!

Platform for annotation: roboflow



Annotation of Unlabeled Traffic Image

How to annotate?

1. Identify the objects in the image
2. Draw a bounding box for 1st object
3. Classify the object
4. Repeat for the rest of the objects

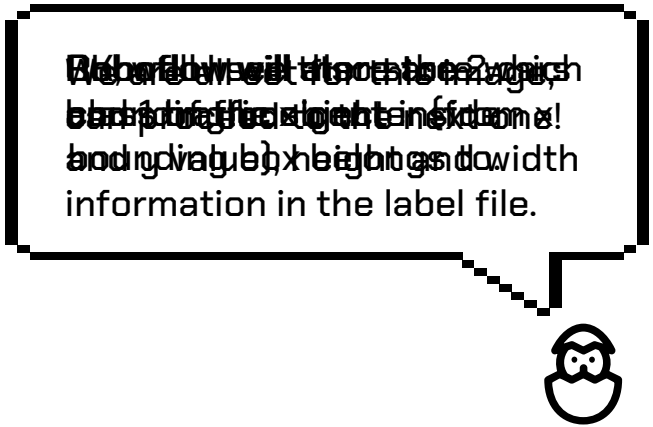
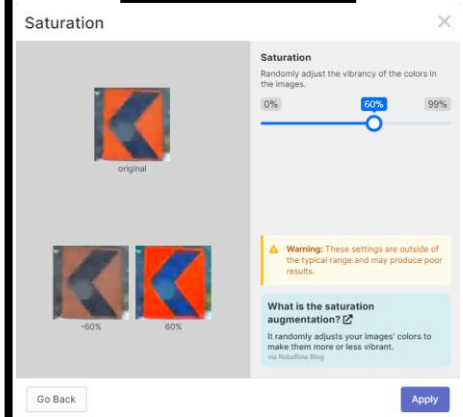


Image Augmentation

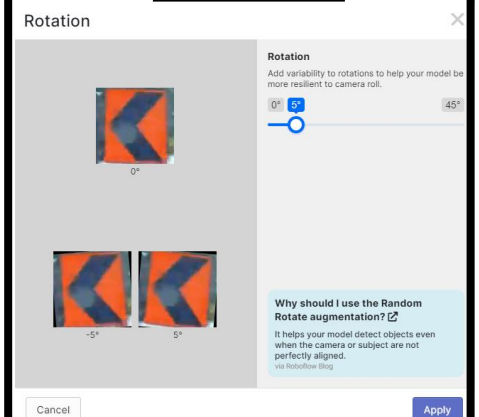
3 methods to boost the image counts!

Saturation



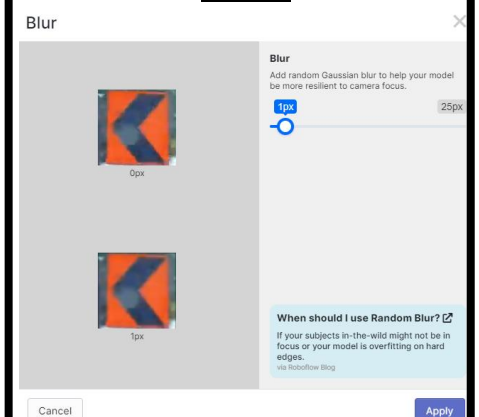
Adjust the saturation ± 60%

Rotation



Rotate the orientation ± 5%

Blur



Apply Gaussian blur with 1px

Image Preprocessing

2613 Total Images

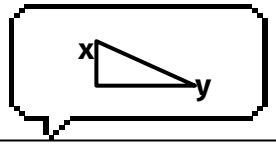
View All Images →

Dataset Split

TRAIN SET	VALID SET	TEST SET
90%	6%	3%
2360 Images	169 Images	84 Images

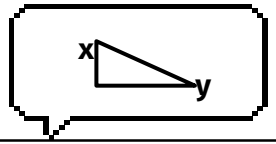
Preprocessing Resize: Stretch to 640x640

BEFORE Augmentation and Preprocessing



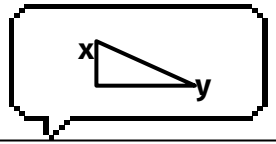
	Udacity Self Driving Car	Singapore Traffic Sign
Images	15,000 images	851 images
Annotation	97,942 labels	Unlabelled
No. of Class	11	7
Resolution	512 x 512 Pixels	Not fixed Pixel, Low Resolution
Content	Car, pedestrian, truck and traffic lights	7 traffic signs related with PMD/Pedestrian
Sources	roboflow	
Readiness	Ready for modelling	Raw images

AFTER Augmentation and Preprocessing



	Udacity Self Driving Car	Singapore Traffic Sign
Images	15,000 images	2853 images
Annotation	97,942 labels	2613 labels
No. of Class	12	7
Resolution	512 x 512 Pixels	Not fixed - 640 x 640 Pixels
Content	Car, pedestrian, truck and traffic lights	7 traffic signs related with PMD/Pedestrian
Sources	roboflow	roboflow
Readiness	Ready for modelling	Ready for modelling

Merging both image dataset



	Udacity Self Driving Car	Singapore Traffic Sign
Images	15,000 images	2613 images
Annotation	97,942 labels	2613 labels
No. of Class	11	7
Resolution	512 x 512 Pixels	640 x 640 Pixels
Content	Car, pedestrian, truck and traffic lights	7 traffic signs related with PMD/Pedestrian
Sources	roboflow	roboflow
Readiness	Ready for modelling	Ready for modelling

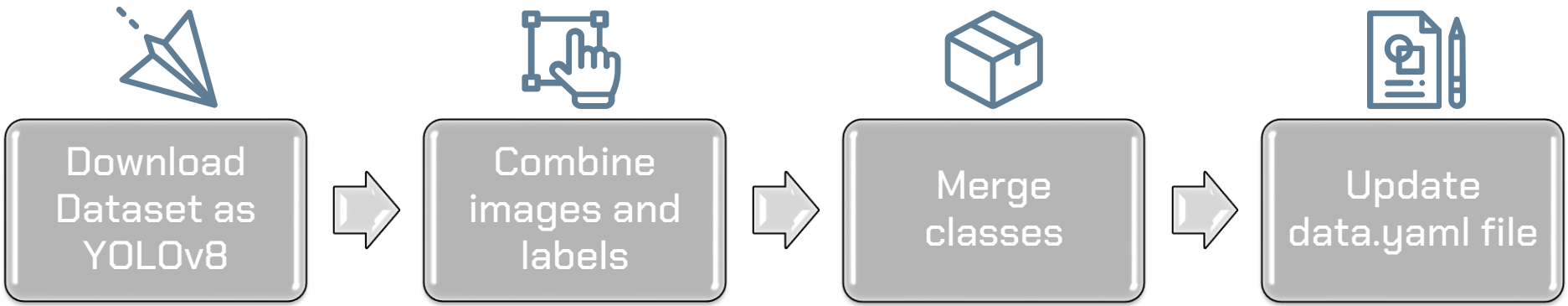
Before that... let's look at the classes

Udacity Self Driving Car	Singapore Traffic Sign

```plaintext
biker                    # Class 0
car                      # Class 1
pedestrian               # Class 2
trafficLight             # Class 3
trafficLight-Green       # Class 4
trafficLight-GreenLeft   # Class 5
trafficLight-Red         # Class 6
trafficLight-RedLeft     # Class 7
trafficLight-Yellow      # Class 8
trafficLight-YellowLeft  # Class 9
truck                    # Class 10
```

```plaintext
curveleft_sign           # Class 0
curveright_sign          # Class 1
mandatorystop_sign       # Class 2
noentry_sign             # Class 3
nojaywalking_sign        # Class 4
pedestriancrossing_sign  # Class 5
zebracrossing_sign       # Class 6
```

Merging both image dataset



Before that... let's look at the classes

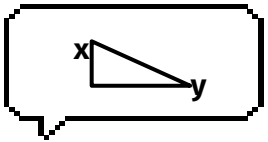
Udacity Self Driving Car	Singapore Traffic Sign	Combined Dataset

```plaintext
biker                    # Class 0
car                      # Class 1
pedestrian               # Class 2
trafficLight             # Class 3
trafficLight-Green       # Class 4
trafficLight-GreenLeft   # Class 5
trafficLight-Red         # Class 6
trafficLight-RedLeft     # Class 7
trafficLight-Yellow      # Class 8
trafficLight-YellowLeft  # Class 9
truck                    # Class 10
```

```plaintext
curveleft_sign           # Class 0
curveright_sign          # Class 1
mandatorystop_sign       # Class 2
noentry_sign             # Class 3
nojaywalking_sign        # Class 4
pedestriancrossing_sign  # Class 5
zebracrossing_sign       # Class 6
```

```yaml
names:
  - nojaywalking_sign          # Class 0
  - mandatorystop_sign         # Class 1
  - curveright_sign            # Class 2
  - zebracrossing_sign         # Class 3
  - noentry_sign               # Class 4
  - pedestriancrossing_sign    # Class 5
  - curveleft_sign             # Class 6
  - biker                      # Class 7
  - car                        # Class 8
  - pedestrian                 # Class 9
  - trafficlight               # Class 10
  - trafficlight_green         # Class 11
  - trafficlight_red           # Class 12
  - trafficlight_yellow        # Class 13
  - truck                      # Class 14
```

Merging both image dataset

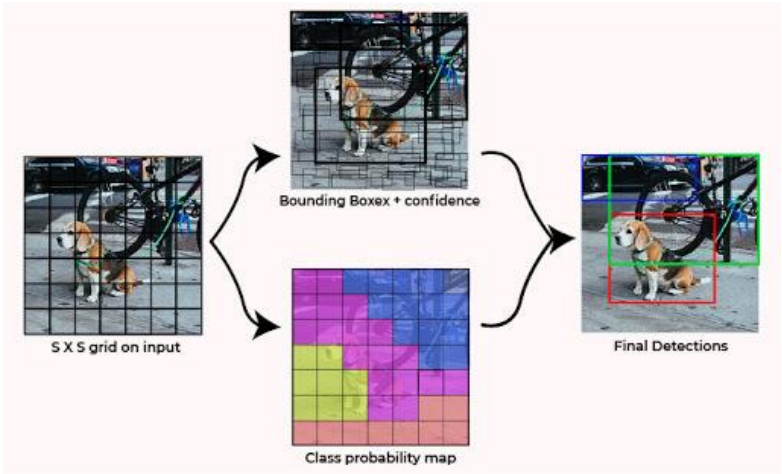


	Udacity Self Driving Car	Singapore Traffic Sign	Combined Dataset
Images	15,000 images	2613 images	17,613 images
Annotation	97,942 labels	2613 labels	100,555 labels
No. of Class	11	7	15 (was 18 before merging traffic light)
Content	Car, pedestrian, truck and traffic lights	7 traffic signs related with PMD/Pedestrian	7 traffic signs, car, pedestrian, truck and traffic lights

03

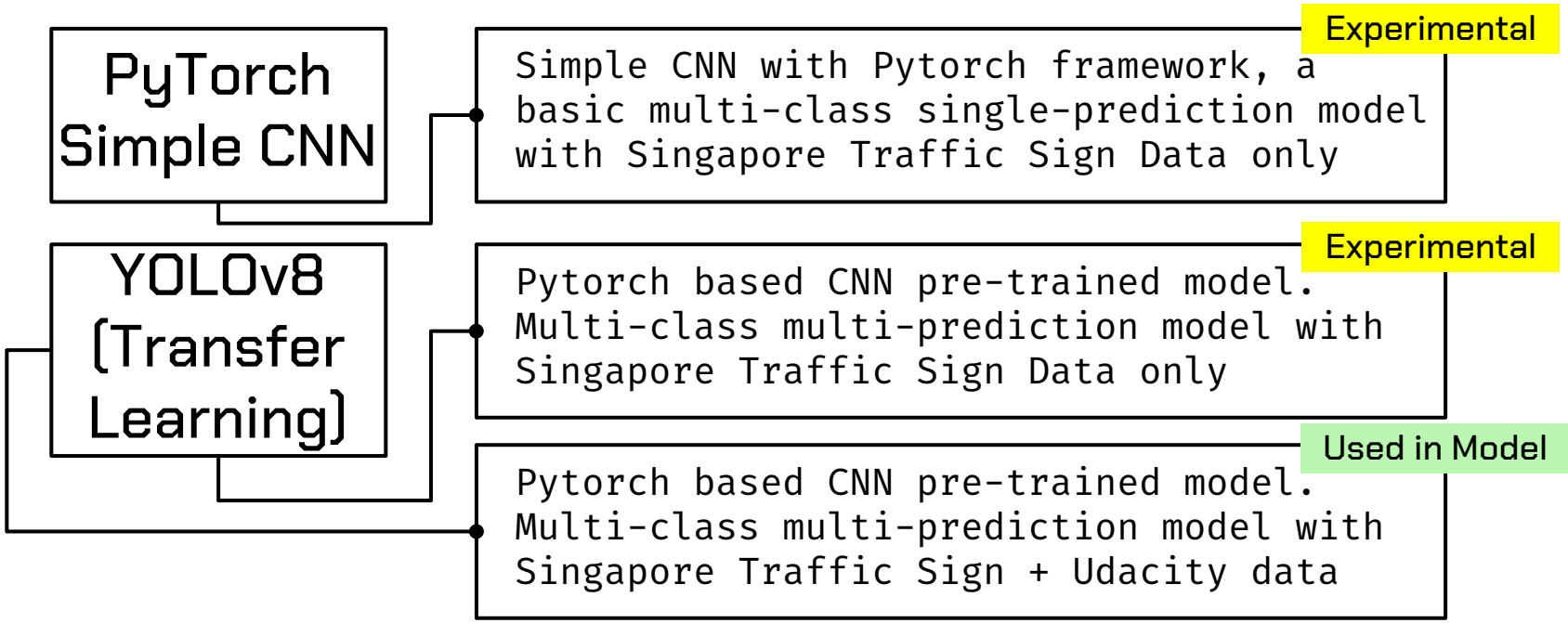
Data Modelling & Hyperparameter tuning

What is YOLO?



YOLO (You Only Look Once) is a fast, efficient real-time object detection system that uses convolutional neural networks (CNNs). It detects multiple objects in images or videos with a single look.

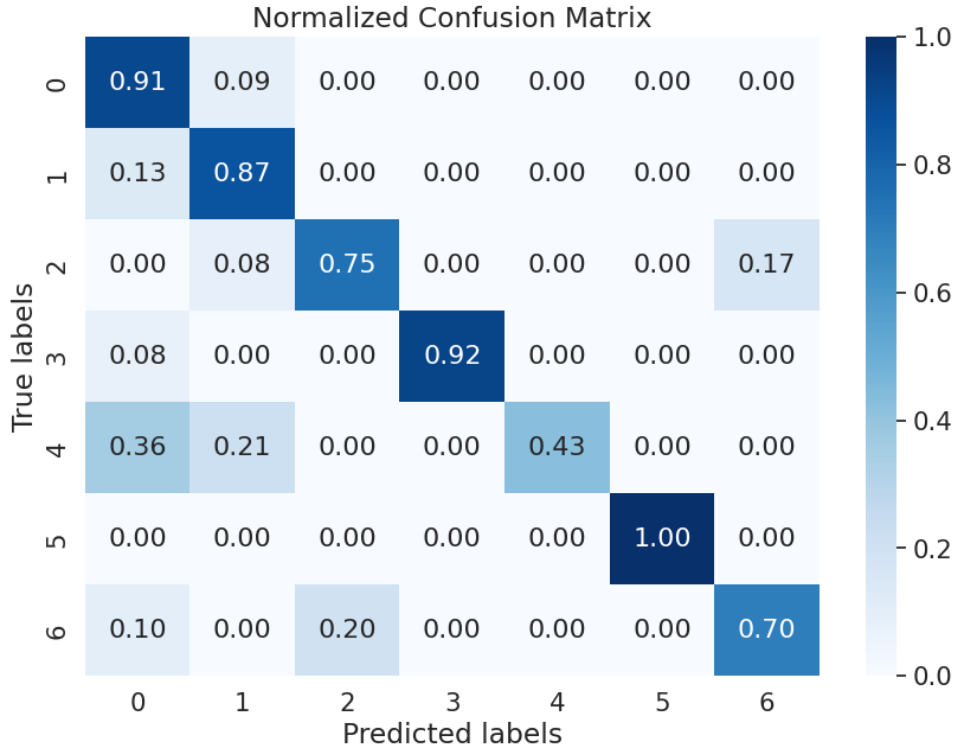
Models


PyTorch Simple CNN
Simple CNN with Pytorch framework, a basic multi-class single-prediction model with Singapore Traffic Sign Data only
Experimental
YOLOv8 (Transfer Learning)
Pytorch based CNN pre-trained model. Multi-class multi-prediction model with Singapore Traffic Sign Data only
Experimental
Pytorch based CNN pre-trained model. Multi-class multi-prediction model with Singapore Traffic Sign + Udacity data
Used in Model

PyTorch Simple CNN Result

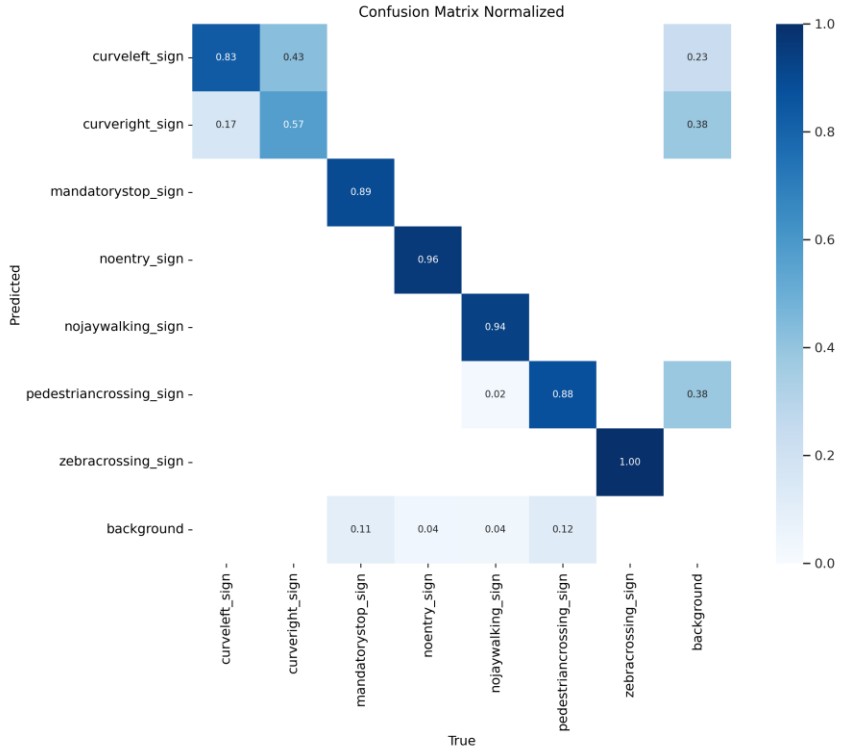
```plaintext
curveleft_sign              # Class 0
curveright_sign             # Class 1
mandatorystop_sign          # Class 2
noentry_sign                # Class 3
nojaywalking_sign           # Class 4
pedestriancrossing_sign     # Class 5
zebracrossing_sign          # Class 6
```

93.6%
Sensitivity



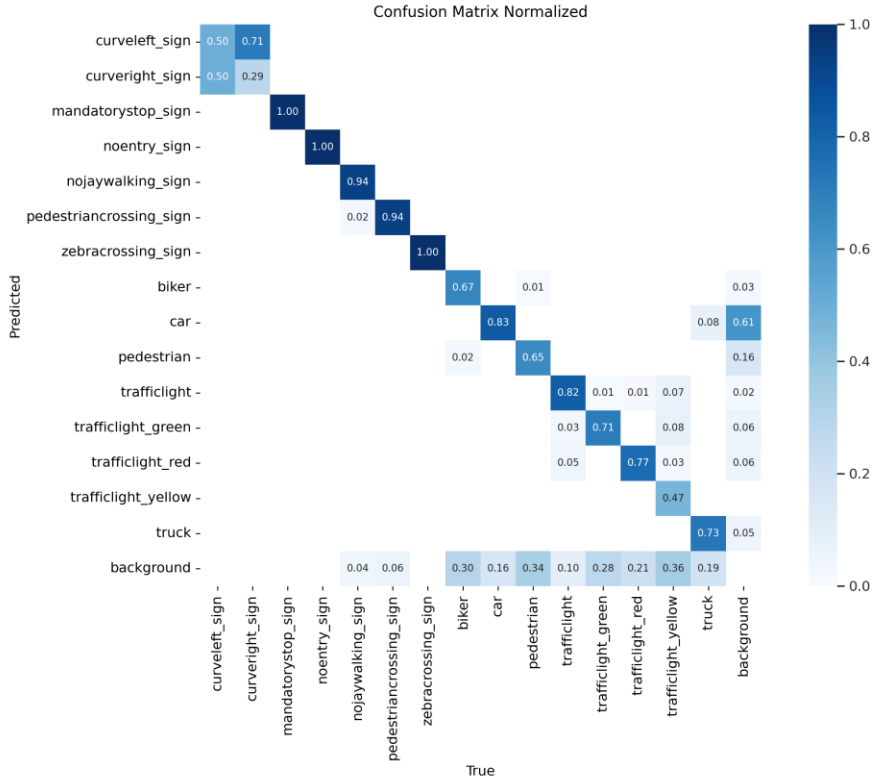
YOLOv8 with Traffic Sign Data

92%
Sensitivity



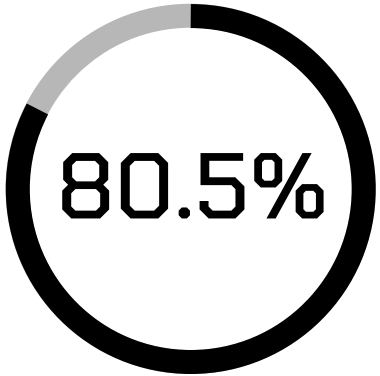
YOLOv8 with Master Data

80.5%
Sensitivity



Yolov8 Hyperparameter Tuning

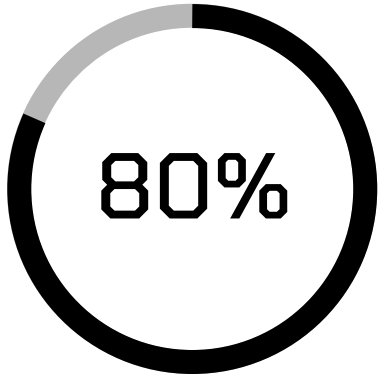
Base Model



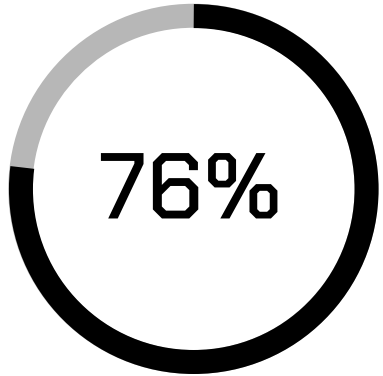
Default

Hyperparameter Tuning

Learning Rate : 0.001, 0.01 and 0.1
Batch Size : 8 and 16



Batch size =16



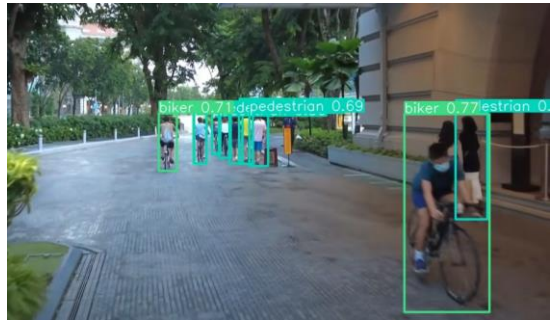
Batch size = 8

Results of different learning rate are similar.
Proceed with Based Model.

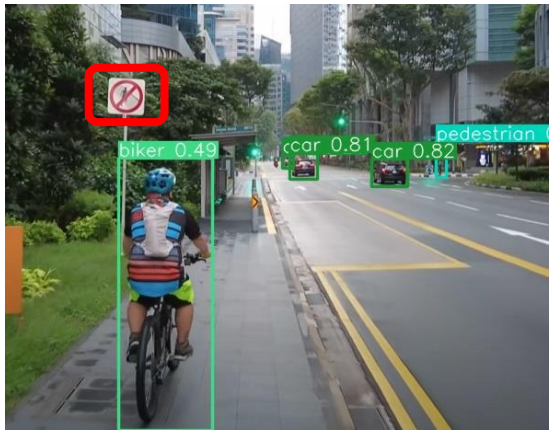
04

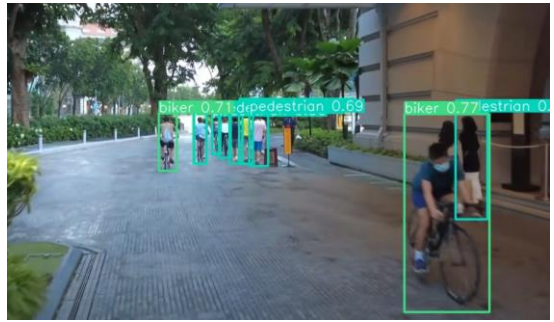
Demonstration

Traffic Sign + Self Drive Car
Yolov8 Multiclass Multi Prediction



Finding – Bad at Recognizing Traffic Signs





Video Demo

05

Conclusion and Recommendation

TheiaVision brings benefit to...

Our Company – MaximalSG	1. Product Differentiation 2. Safety compliance
General Public	1. Increase safety and confidences 2. Better user experience
LTA	1. Safety Compliance 2. Collect data for Urban Planning
SCDF	1. Reduce frequency of deployment 2. Accurate Accident Reconstruction
Insurance Compines	1. Reduction in Claims 2. Risk assessment and management

Cost Benefit Analysis

+ =

10% - New Sale due to new feature, 5000 units increased

30% - $150 Markup from original price ($500 to $650)

13% - Revenue increased

Category	Description	Amount (SGD)
Costs		
Data Scientists (2)	SGD 90,000 each per year	180,000
Developers (2)	SGD 80,000 each per year	160,000
Project Manager	SGD 100,000 per year	100,000
Development Hardware	Cloud services and physical devices	20,000
Software Licensing	Licenses for tools and software	10,000
Marketing Campaign	Promotional activities	50,000
Distribution Costs	Increased logistics costs	20,000
Maintenance and Updates	Annual maintenance	30,000
Total Costs		**570,000**
Benefits		
Increase in Sales Revenue	5,000 units at SGD 650 each	3,250,000
Total Benefits		**3,250,000**
Net Benefit	Total Benefits - Total Costs	**2,680,000**
Revenue Analysis		
Original Revenue	50,000 units at SGD 500 each	25,000,000
New Total Revenue	Original Revenue + Increase in Sales Revenue	28,250,000
Percentage Increase in Revenue	((New Total Revenue - Original Revenue) / Original Revenue) × 100%	**13%**

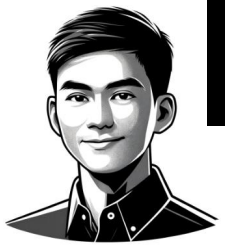
Conclusion

Problem Statement

How can we enhance the safety of Personal Mobility Devices (PMDs) in urban environments by using object detection to improve PMD users' ability to perceive and respond to their surroundings?

Conclusion

TheiaVision helps to detect object with 80.5% sensitivity



[Darren]
I have more confidence to convince the management team for this project.

Recommendation

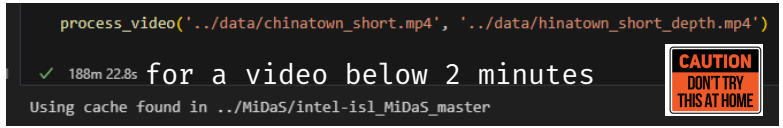
1. Handle imbalance datasets (SMOTE)
2. Include Diverse Image of Traffic Sign (Currently Data from DashCam Only, low resolution)
3. Stereo Video for depth estimation (Monocular video is used in this project)
4. Including hazard detection (pavement condition, construction, etc.)
5. Install Speaker for Voice Feedback to PMD Users

Depth Estimation Attempt...

Attempt #1:
"Did research on how to utilize my **iPhone's LiDAR sensor/duo-camera (Stereo Video)** to capture the **"Depth" information**. However, this requires a **Mac machine and coding with Swift programming**. Due to time constraints, I cannot afford to do that."

Attempt #2:
"I am finding **third-party apps** that utilize **LiDAR sensor**, but most of the apps are used for **3D object modeling and mapping**."

Attempt #3:
"Researched any algorithm/model that can handle Depth estimation with **monocular RGB video** and found **MiDaS (Monocular Depth Estimation)** a CNN architecture model."




THANK YOU

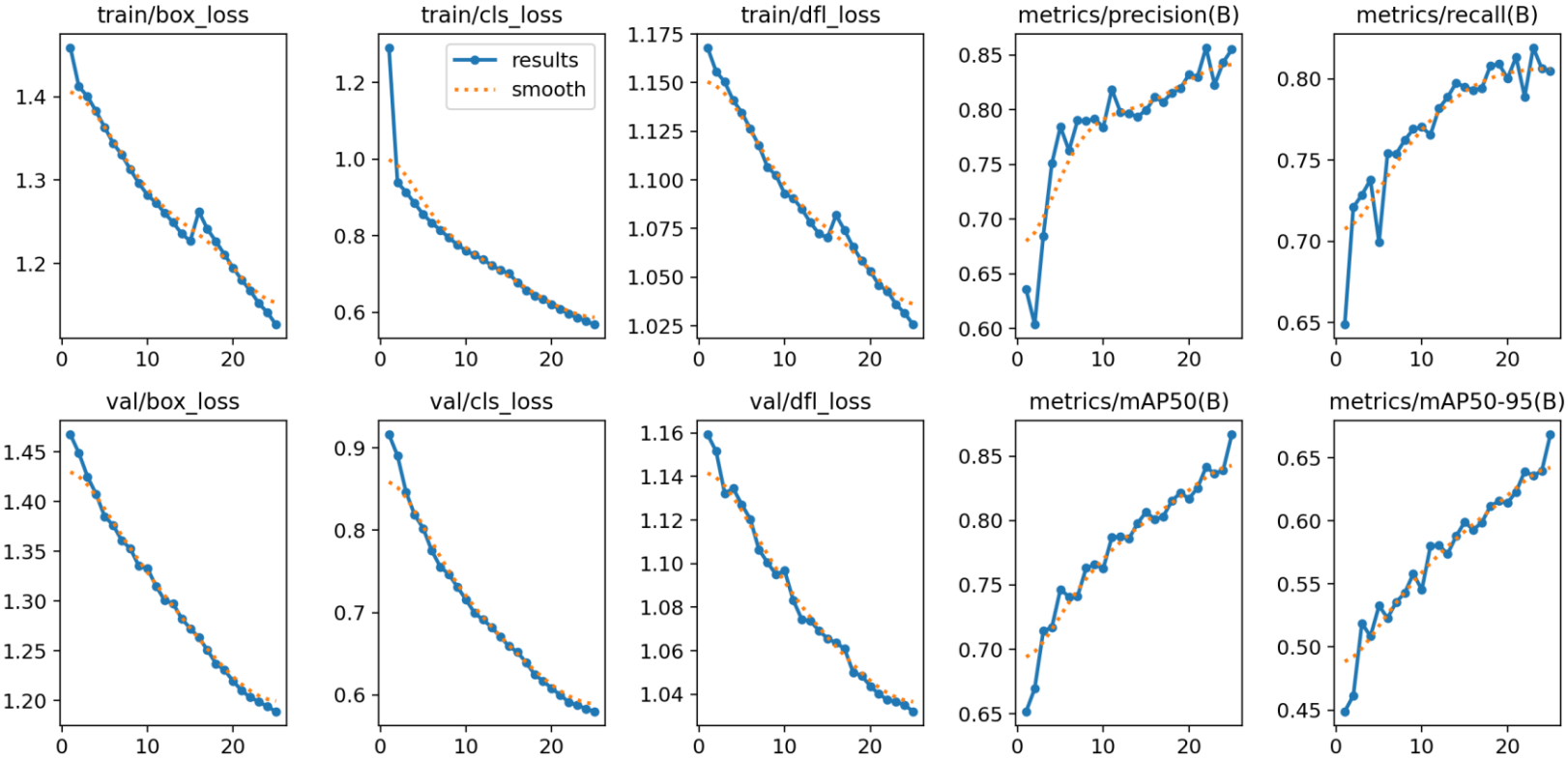
Running Model on my own PC
Really computationally expensive

Name	Status	18% CPU	96% GPU	70% Memory	1% Disk	0% Network
Virtual Machine Worker Process		0%	95.3%	3.7 MB	0 MB/s	0 Mbps
Visual Studio Code (17)		0.9%	0.2%	492.3 MB	0 MB/s	0 Mbps
Client Server Runtime Process		0%	0.1%	1.1 MB	0 MB/s	0 Mbps
Desktop Window Manager		0.4%	0.1%	84.1 MB	0 MB/s	0 Mbps
wsappx		0%	0%	3.2 MB	0 MB/s	0 M
SnippingTool.exe		0%	0%	3.5 MB		
System		1.0%	0%	0.1 MB		
Task Manager		0.2%	0%	80.1 MB		
Windows Explorer (2)		0.2%	0%	203.5 MB		
iCloud (9)		0.5%	0%	52.8 MB		
Application Frame Host		0%	0%	5.0 MB	0	
Windows Defender SmartScre...		0%	0%	1.4 MB	0 M	
VmmemWSL		14.9%	0%	11,319.8 ...	0.1 MB/	0 Mbps

Specification of models

Feature	SimpleCNN Model	YOLOv8 Model (Singapore Traffic Sign)	YOLOv8 Model (Singapore Traffic Sign + Udacity data)
Framework	PyTorch	YOLO	YOLO
Model Type	Custom simple CNN	Pre-trained YOLOv8s model	Pre-trained YOLOv8s model
Input Size	50x50 pixels	640x640 pixels	640x640 pixels
Batch Size	128 for training and validation; 1 for testing	8	8
Epochs	50	20	25
Main Operations	Convolution, ReLU activation, pooling, fully connected layers	Convolutional operations optimized for object detection	Convolutional operations optimized for object detection
Optimization Algorithm	RMSprop with a learning rate of 0.001	Not specified, likely configurable in YOLO setup	Not specified, likely configurable in YOLO setup
Loss Function	CrossEntropyLoss	Typically a composite loss suitable for object detection tasks	Typically a composite loss suitable for object detection tasks
Output	Class scores for 7 classes	Bounding boxes with class probabilities and objectness scores	Bounding boxes with class probabilities and objectness scores
Use Case	Basic image classification	Advanced object detection in images	Advanced object detection in images

Model Result



Merging Roboflow Yolov8 Dataset

Merging Roboflow Yolov8 Dataset

1. Consolidate the data

Step 1: Consolidate the Data

First, you'll need to combine the image and label files from both datasets. This involves physically moving or copying the image and corresponding label files from both datasets into new directories for training, validation, and testing.

Directory Structure:

- **Merged Dataset**
  - **train**
    - images
    - labels
  - **val**
    - images
    - labels
  - **test**
    - images
    - labels

Merging Roboflow Yolov8 Dataset

2. Update Yaml file
Merge traffic Light Left and traffic Light

You'll need to update the `data.yaml` file to reflect the new structure and combined class list. Here's how you might structure your updated `data.yaml`:

yaml — Copy code

```yaml
names:
- curveleft_sign
- curveright_sign
- mandatorystop_sign
- noentry_sign
- nojaywalking_sign
- pedestriancrossing_sign
- zebracrossing_sign
- biker
- car
- pedestrian
- trafficLight
- trafficLight-Green
- trafficLight-GreenLeft
- trafficLight-Red
- trafficLight-RedLeft
- trafficLight-Yellow
- trafficLight-YellowLeft
- truck
nc: 18
train: merged/train/images
val: merged/valid/images
test: merged/test/images
```

Merging Roboflow Yolov8 Dataset

3. Adjust label indices

Step 3: Adjust Label Indices

If the label files (often `.txt` files in YOLO format) contain class indices, you need to ensure that the indices in the label files for one dataset do not overlap or conflict with those from the other dataset. You will need to write a script to adjust these indices in the label files of the second dataset:

python — Copy code

```python
import os

def adjust_label_indices(label_dir, adjustment):
    for label_file in os.listdir(label_dir):
        path = os.path.join(label_dir, label_file)
        with open(path, 'r+') as file:
            lines = file.readlines()
            file.seek(0)
            for line in lines:
                parts = line.split()
                parts[0] = str(int(parts[0]) + adjustment)  # Adjust the class index
                file.write(" ".join(parts) + "\n")

# Example usage: Adjust labels for the second dataset (starting index would be 7)
adjust_label_indices('path/to/self-driving-car/train/labels', 7)
# Repeat for validation and test sets
```

Merging Roboflow Yolov8 Dataset

Step 4: Verify Data Integrity

After merging, check some of the combined images and their corresponding label files to ensure that everything aligns correctly. This is crucial to prevent training issues later.

Step 5: Train Your Model

With your merged dataset and updated configuration file, you can now proceed to train your model using this larger, unified dataset.

By following these steps, you should be able to effectively merge and utilize both datasets for comprehensive model training.

Traffic Sign Only
Pytorch CNN Multiclass Single Prediction