NEED FOR SPEED : Singapore Edition

Presented by: Nicholas Ong Jing

TABLE OF CONTENTS

01

Overview

- Problem Statement
- Method(ology)

02

Data Collection

- Kaggle Dataset
- Recorded Videos

03

EDA & Visualization

- HOG + SVM

04

Image Classification

- CNN
- VGG16
- ResNet50

05

Object Detection

- Darknet + YOLOv4
- PyTorch + YOLOv5

06

Conclusion

- Areas for Improvement
- Business Recommendations

01

Overview

Background Information

Autonomous vehicles fueling up in Singapore

Problem Statement

Data scientist for CETRAN assigned with a two-fold task on pedestrian identification in an urban setting:

(i) Develop an image classification model to determine whether there are pedestrian(s) in an image; and

(ii) Develop an object detection model to spatially demarcate pedestrian(s) in an image or video, if any.

Model Framework

Predicted variable (y): 1 = Person | 0 = Not Person

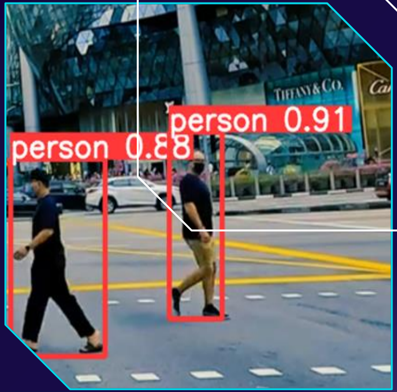


IMAGE CLASSIFICATION

- Self-built CNN models
- Pre-built models (VGG16, ResNet50)

OBJECT DETECTION

- Pre-built models (Darknet + YOLOv4, PyTorch + YOLOv5)

Scoring Metrics

Image Classification (Primary)

Accuracy

Basic indicator: Ratio of correct predictions to total predictions

Loss (Binary Cross-entrpoy)

Negative average of the log of corrected predicted probabilities

Object Detection

Mean Average Precision (mAP)

Mean of average precisions for all classes at a given IoU

Image Classification (Secondary)

Sensitivity

Ratio of true positives over true positives and false negatives

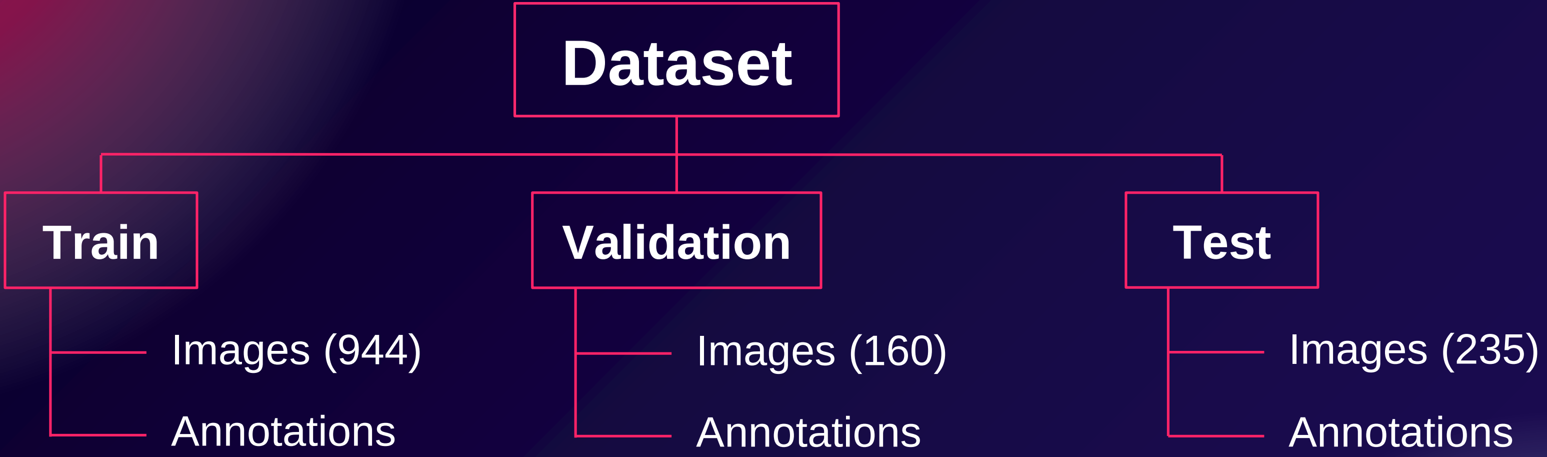
F1-Score

Confidence proxy of a model's predicted positive values

02

Data Collection

Kaggle Pedestrian Dataset



Dataset

Train
- Images (944)
- Annotations

Validation
- Images (160)
- Annotations

Test
- Images (235)
- Annotations

Images contain person and person-like objects:
-Person objects depict actual people
-Person-like objects include statues, mannequins, scarecrows and robots etc.

Recorded Images/Videos

✓

NTU

Ideal testbed for autonomous vehicles

✓

Orchard Road (Day)

Highly urbanized environment with heightened footfall and volume of distractions

✓

Orchard Road (Night)

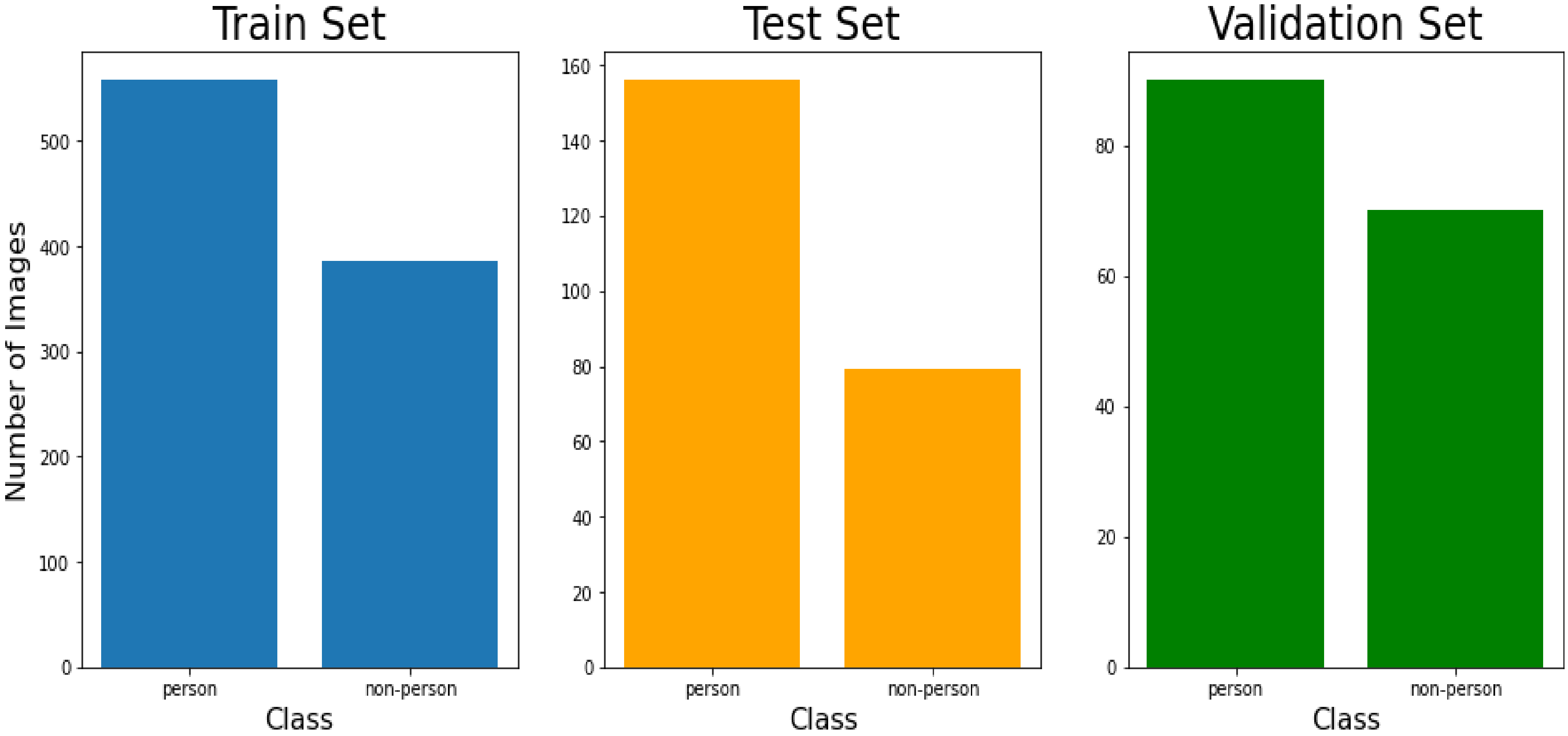
Test under night time conditions to ascertain if model performance deviates significantly

03

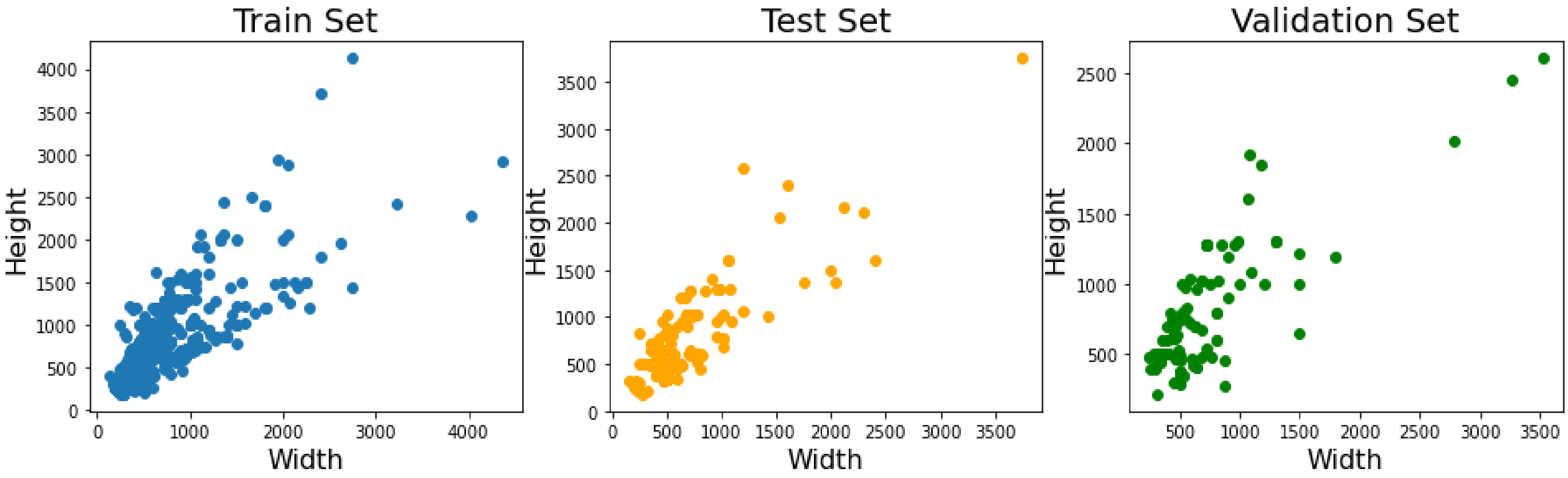
EDA & Visualization

Verify Labels

Check Class Imbalance

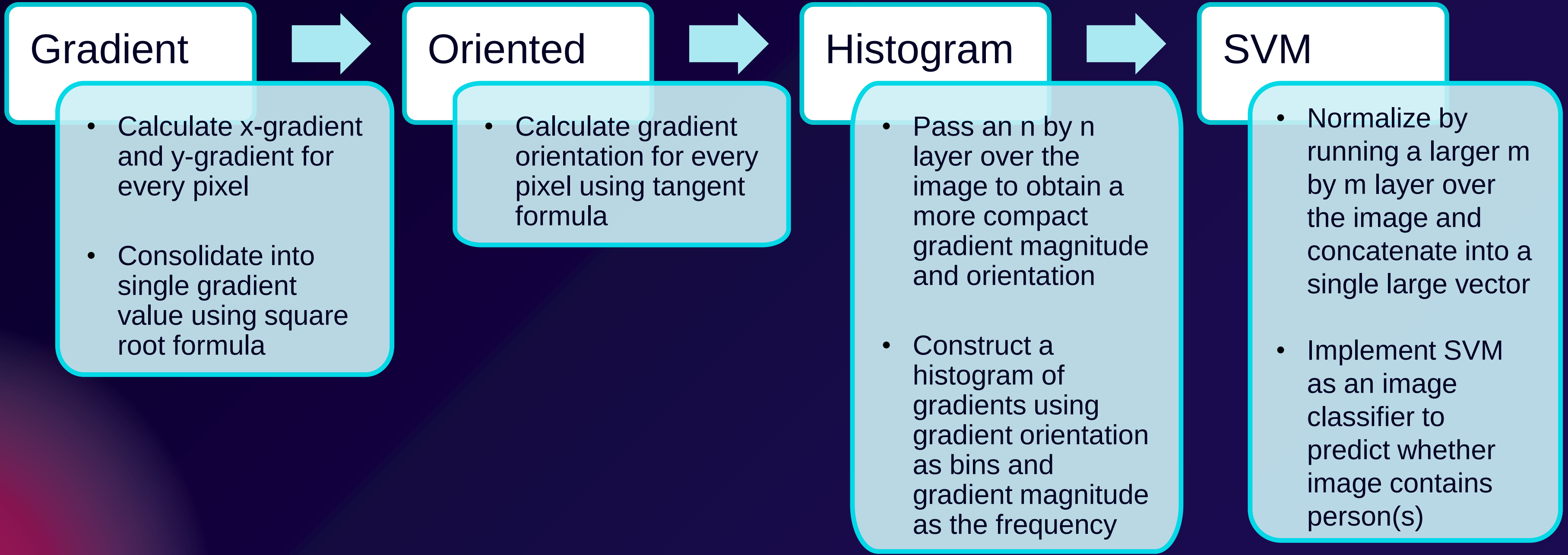


Plot Image Dimensions



People Detection (HOG + SVM)

Histogram of Oriented Gradients (HOG) (Dalal & Triggs, 2005): A feature descriptor providing a simplified representation of an image to allow for easy identification by containing only the most critical details while removing “extra” elements in the image.



Gradient

- Calculate x-gradient and y-gradient for every pixel
- Consolidate into single gradient value using square root formula

Oriented

- Calculate gradient orientation for every pixel using tangent formula

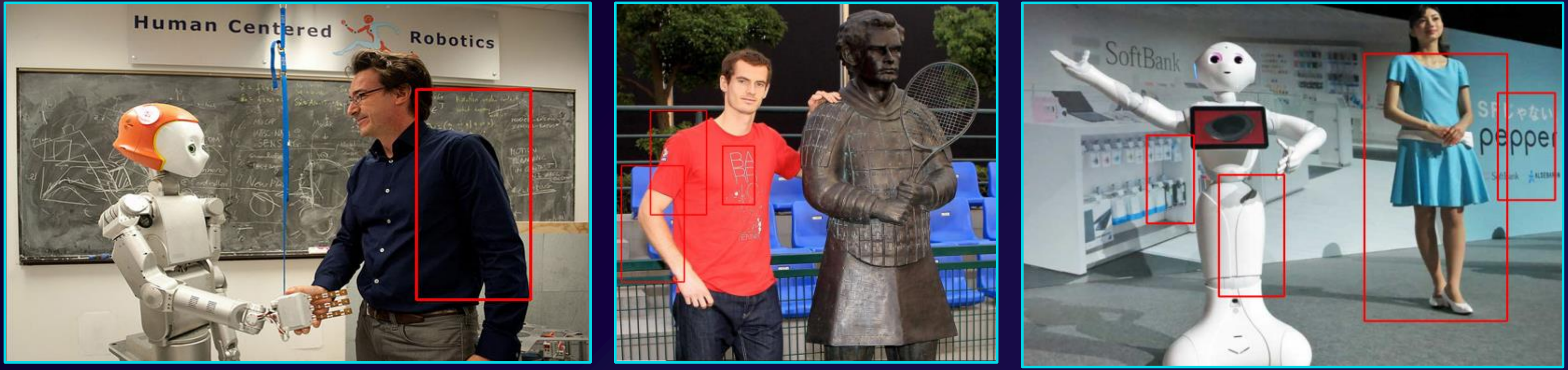
Histogram

- Pass an n by n layer over the image to obtain a more compact gradient magnitude and orientation
- Construct a histogram of gradients using gradient orientation as bins and gradient magnitude as the frequency

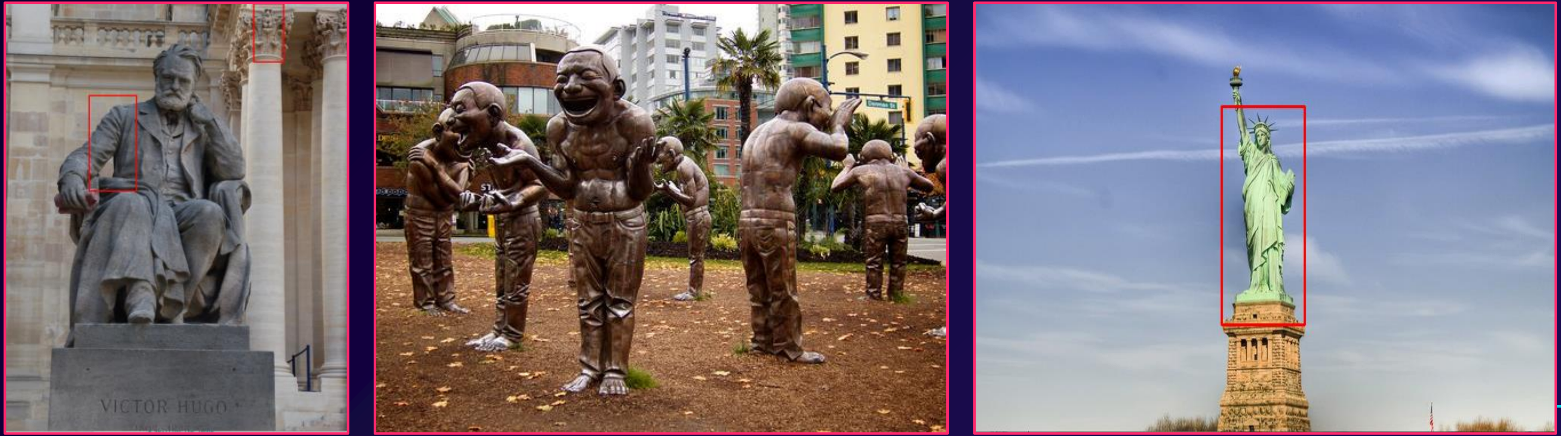
SVM

- Normalize by running a larger m by m layer over the image and concatenate into a single large vector
- Implement SVM as an image classifier to predict whether image contains person(s)

Data Visualization (HOG on Kaggle Dataset)

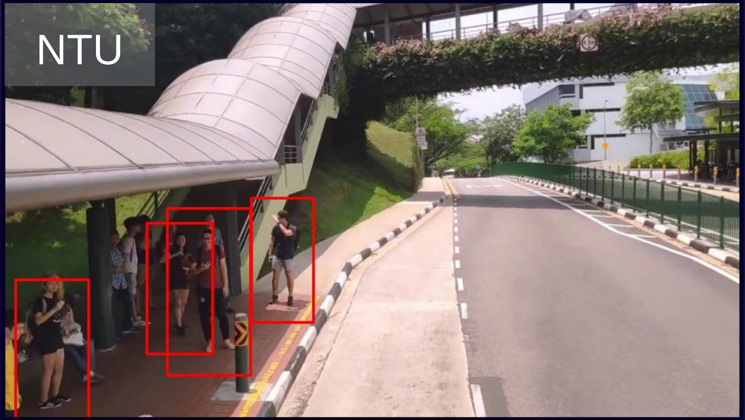


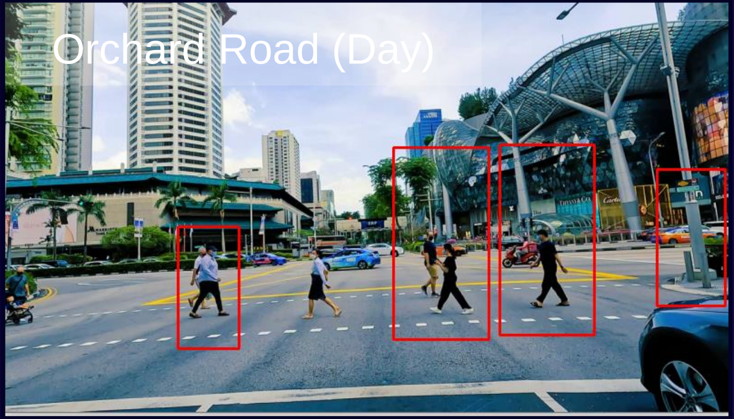
Person Images

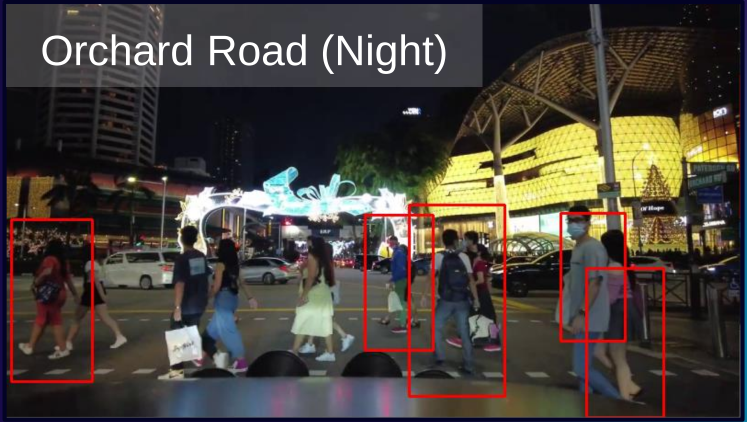


Non-Person Images

Data Visualization (HOG on Recorded Video)



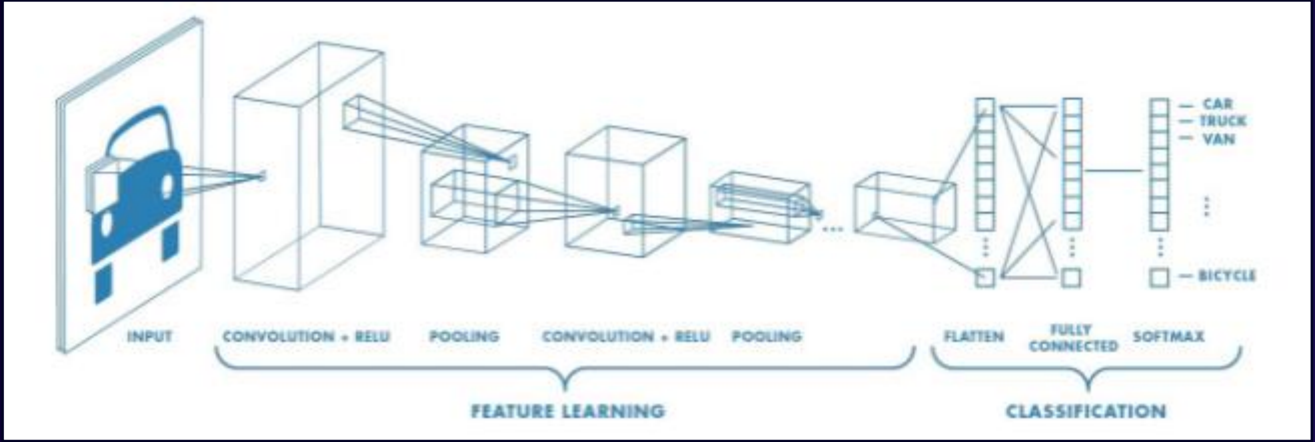




04

Image Classification

Convolutional Neural Network (CNN)

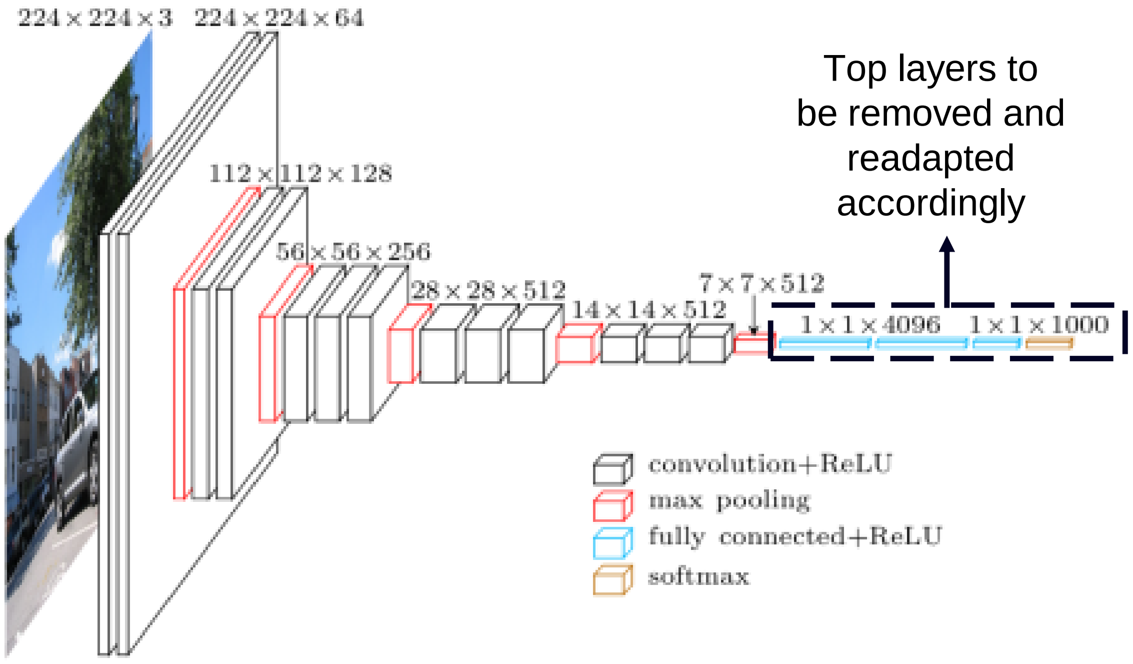


Preprocessing via Keras’ ImageDataGenerator

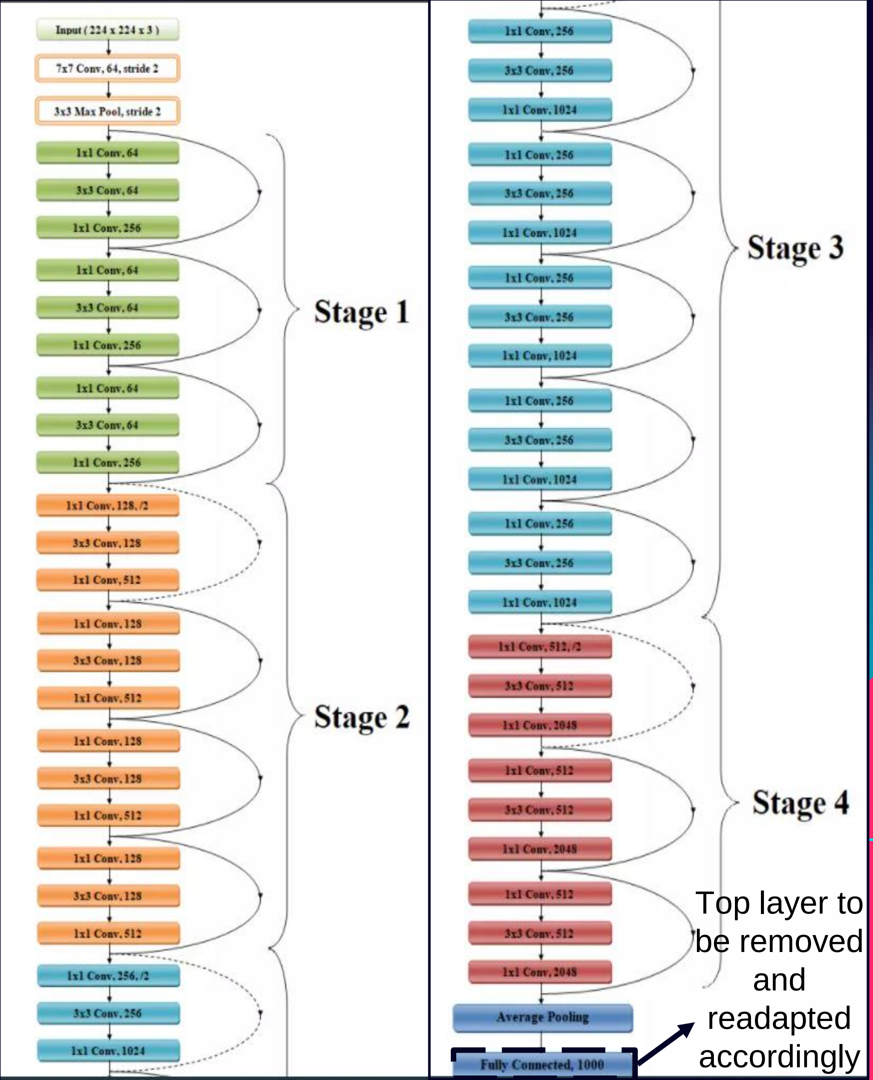
- Architecture: Additional Hidden Layers, Additional Convolutional Stacks
- Tuning: Neurons, Filters, Kernel Size, Learning Rate, Epochs etc.
- Regularization: L2, Dropout, Early Stopping

Transfer Learning: VGG16

Popular CNN model encompassing 16 layers total, including 3 convolutional/pooling stacks



Top layers to be removed and readapted accordingly

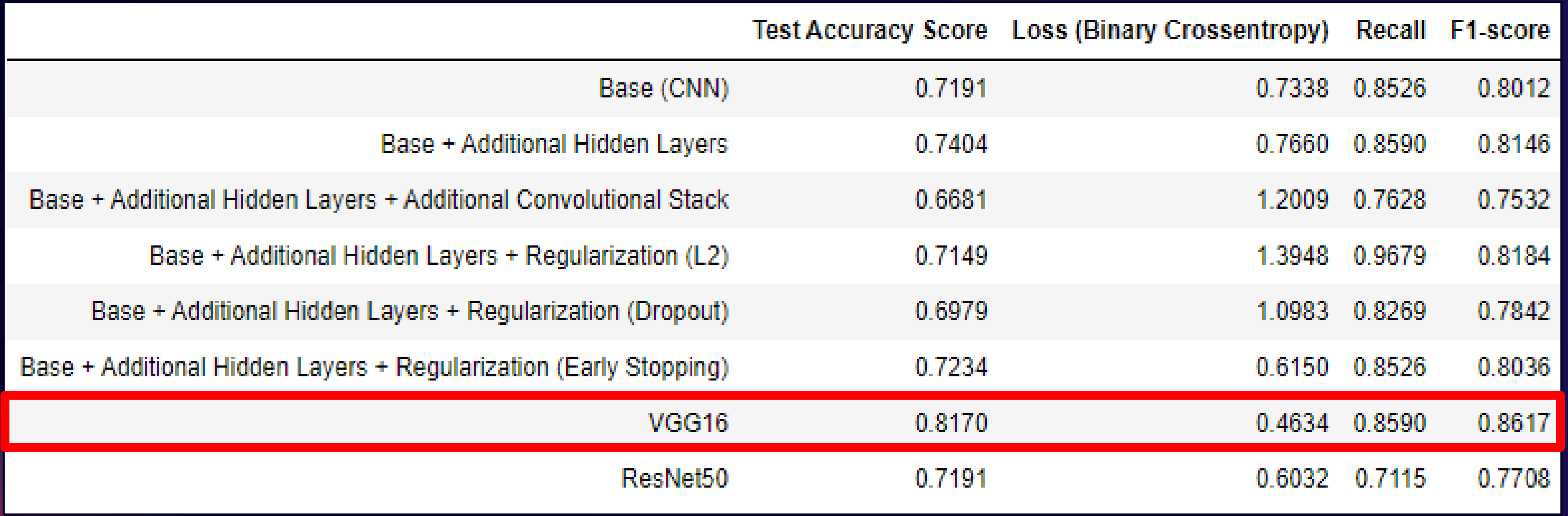


Top layer to be removed and readapted accordingly

Transfer Learning: ResNet50

Gained fame for its ‘skip connection’ ability, which enabled deep neural networks without “vanishing gradients”

□□□ Model Evaluation (Image Classification)

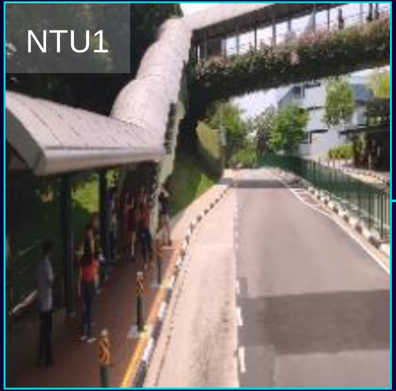


	Test Accuracy Score	Loss (Binary Crossentropy)	Recall	F1-score
Base (CNN)	0.7191	0.7338	0.8526	0.8012
Base + Additional Hidden Layers	0.7404	0.7660	0.8590	0.8146
Base + Additional Hidden Layers + Additional Convolutional Stack	0.6681	1.2009	0.7628	0.7532
Base + Additional Hidden Layers + Regularization (L2)	0.7149	1.3948	0.9679	0.8184
Base + Additional Hidden Layers + Regularization (Dropout)	0.6979	1.0983	0.8269	0.7842
Base + Additional Hidden Layers + Regularization (Early Stopping)	0.7234	0.6150	0.8526	0.8036
VGG16	0.8170	0.4634	0.8590	0.8617
ResNet50	0.7191	0.6032	0.7115	0.7708

- VGG16 adopted as production model

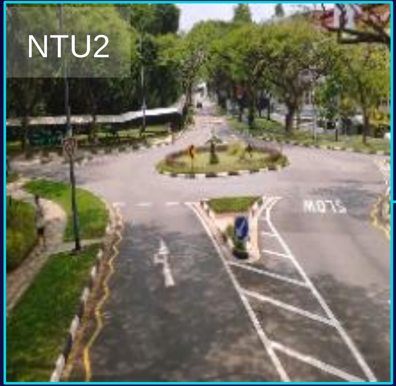
Image Classification Predictions (NTU)

Images with Pedestrians



NTU1 Prediction: There are no pedestrian(s) in the image

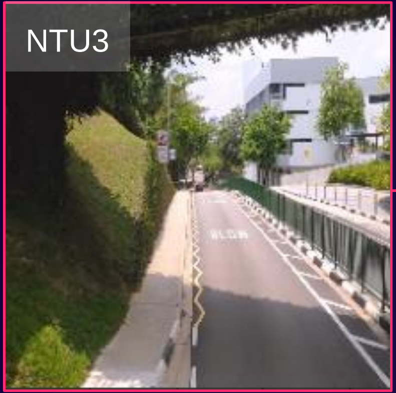
☹



NTU2 Prediction: There are pedestrian(s) in the image

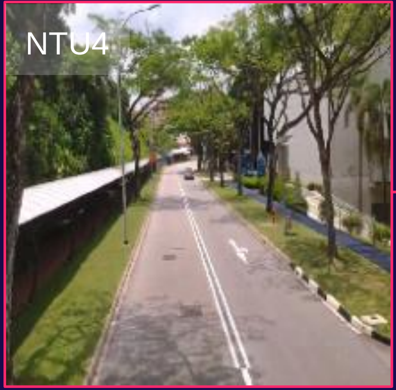
😃

Images without Pedestrians



NTU3 Prediction: There are no pedestrian(s) in the image

😃

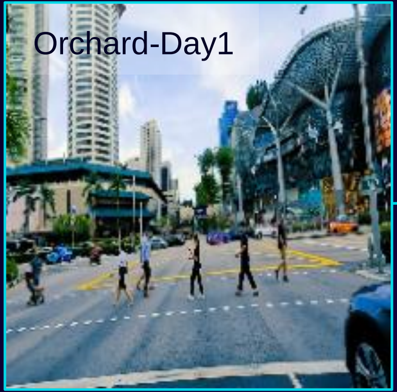


NTU4 Prediction: There are pedestrian(s) in the image

☹

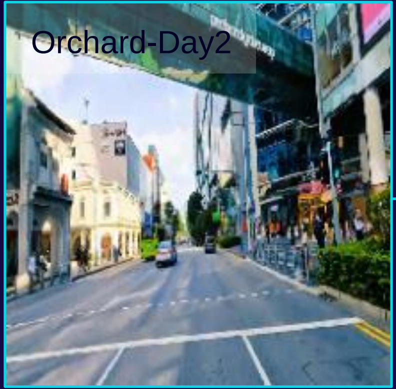
Image Classification Predictions (Orchard Road - Day)

Images with Pedestrians



Orchard-Day1 Prediction: There are no pedestrian(s) in the image

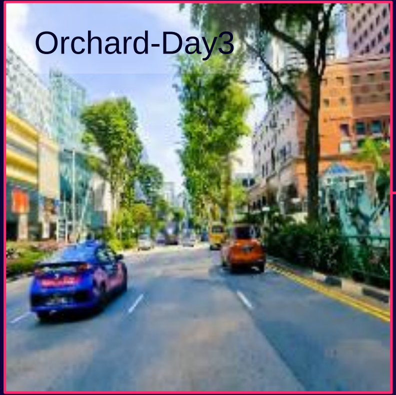
☹



Orchard-Day2 Prediction: There are pedestrian(s) in the image

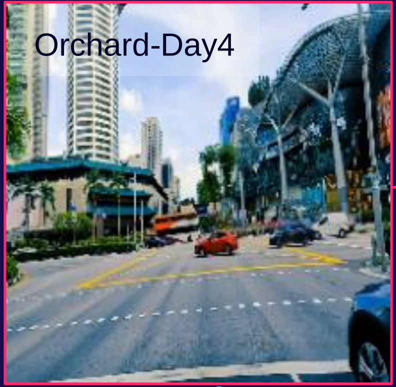
😃

Images without Pedestrians



Orchard-Day3 Prediction: There are pedestrian(s) in the image

☹

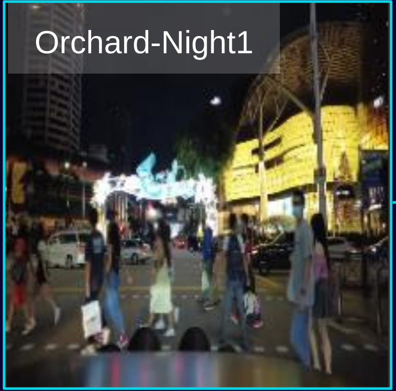


Orchard-Day4 Prediction: There are no pedestrian(s) in the image

😃

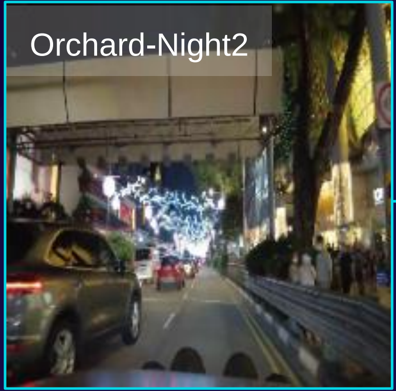
Image Classification Predictions (Orchard Road - Night)

Images with Pedestrians



Orchard-Night1 Prediction: There are no pedestrian(s) in the image

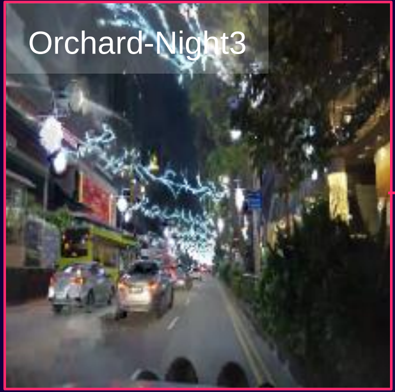
☹



Orchard-Night2 Prediction: There are pedestrian(s) in the image

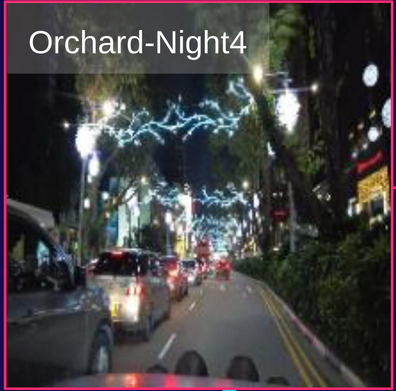
😃

Images without Pedestrians



Orchard-Night3 Prediction: There are no pedestrian(s) in the image

😃

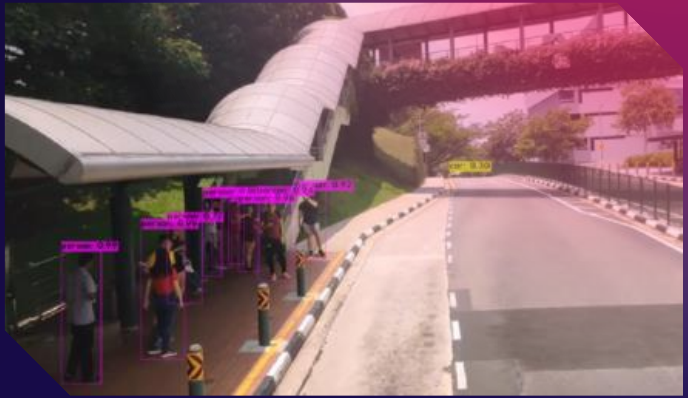


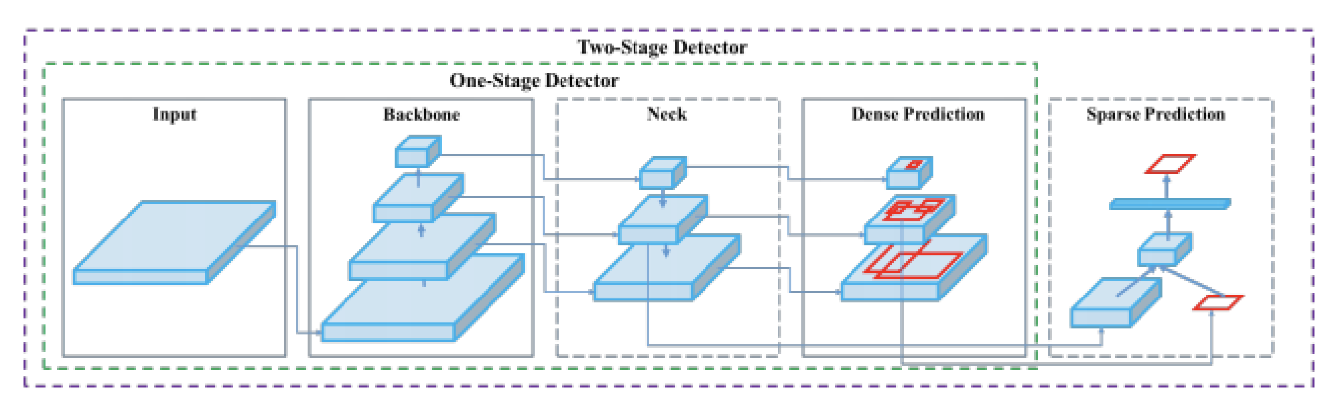
Orchard-Night4 Prediction: There are pedestrian(s) in the image

☹

05

Object Detection



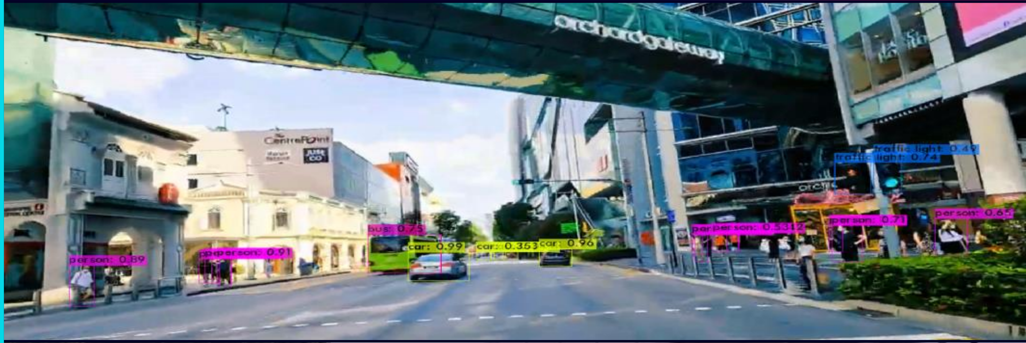


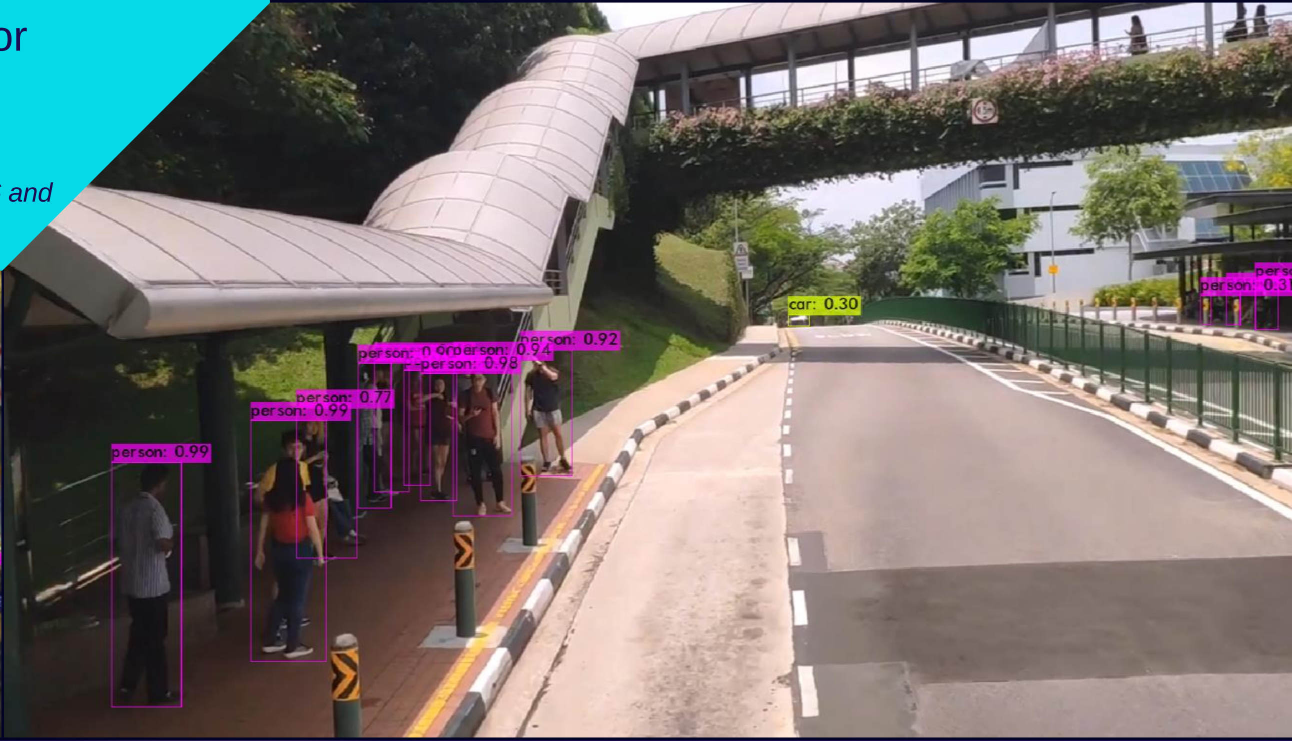
You Only Look Once (YOLO)

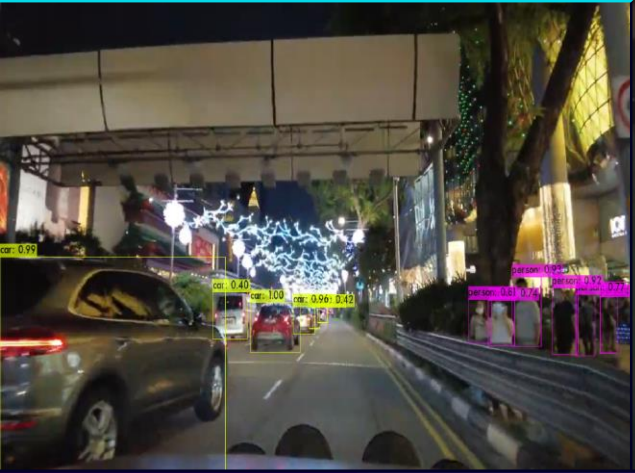
Darknet + YOLOv4

Custom framework tailored for object detection in CV

**Deployed locally via GPU (NVIDIA GeForce GTX1080) with the OpenCV 4.5.5, CUDA 11.6 and CUDNN 8.3.2*





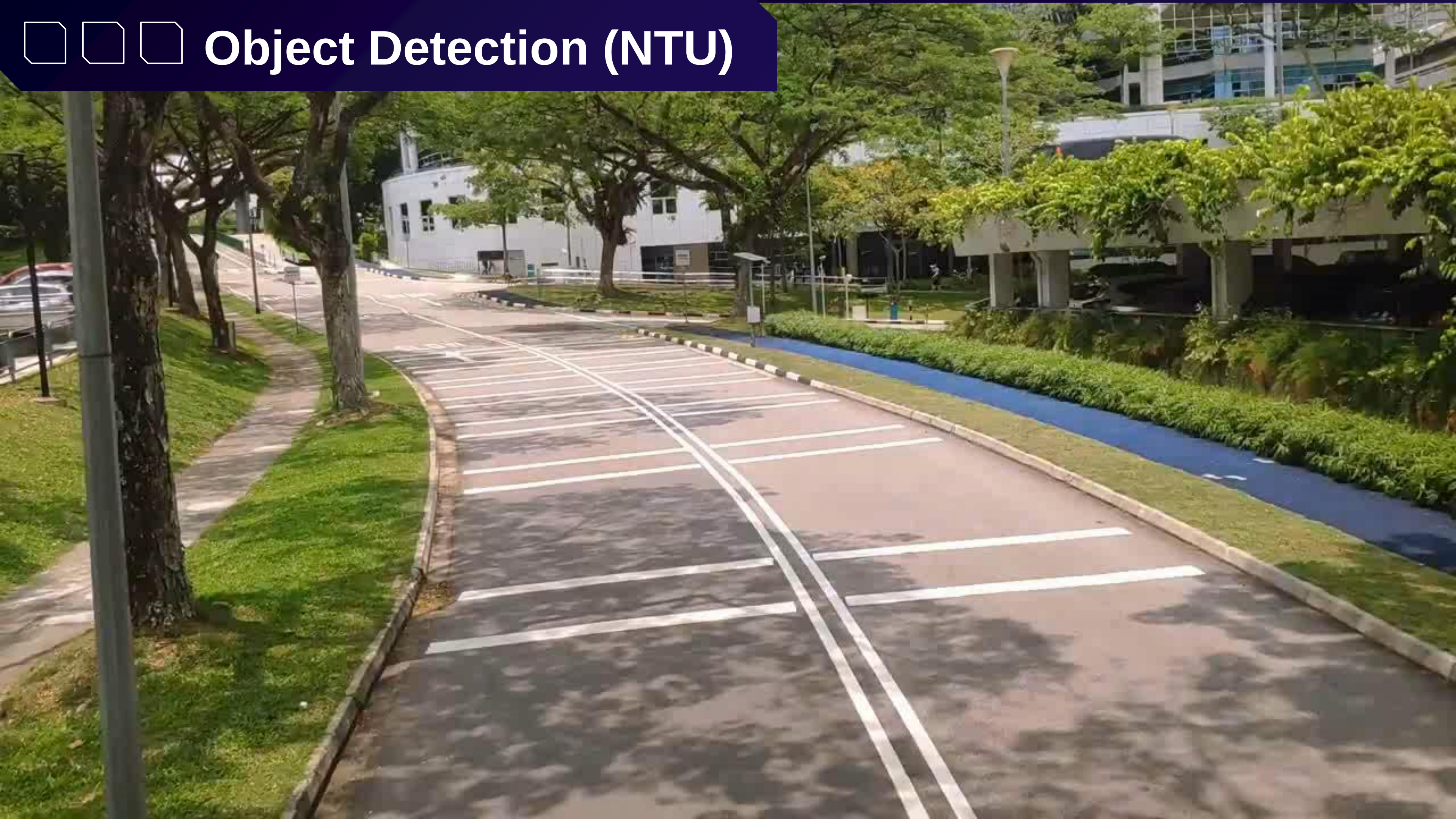


PyTorch + YOLOv5

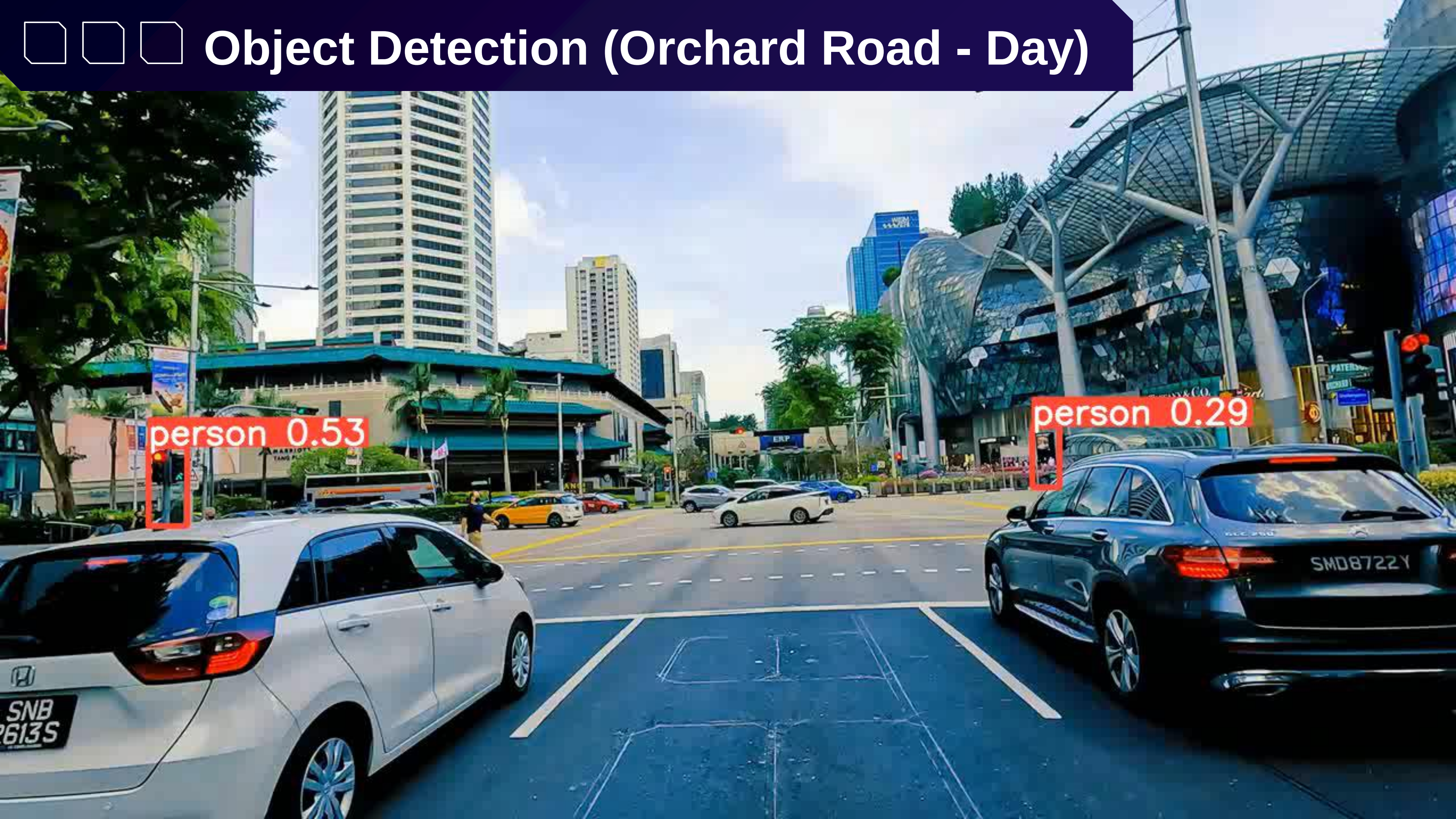
- Main difference vs YOLOv4 is the backbone component
- Trained on custom pedestrian dataset
- Bounding box and object annotation via Roboflow

**Deployed via Google Colab*

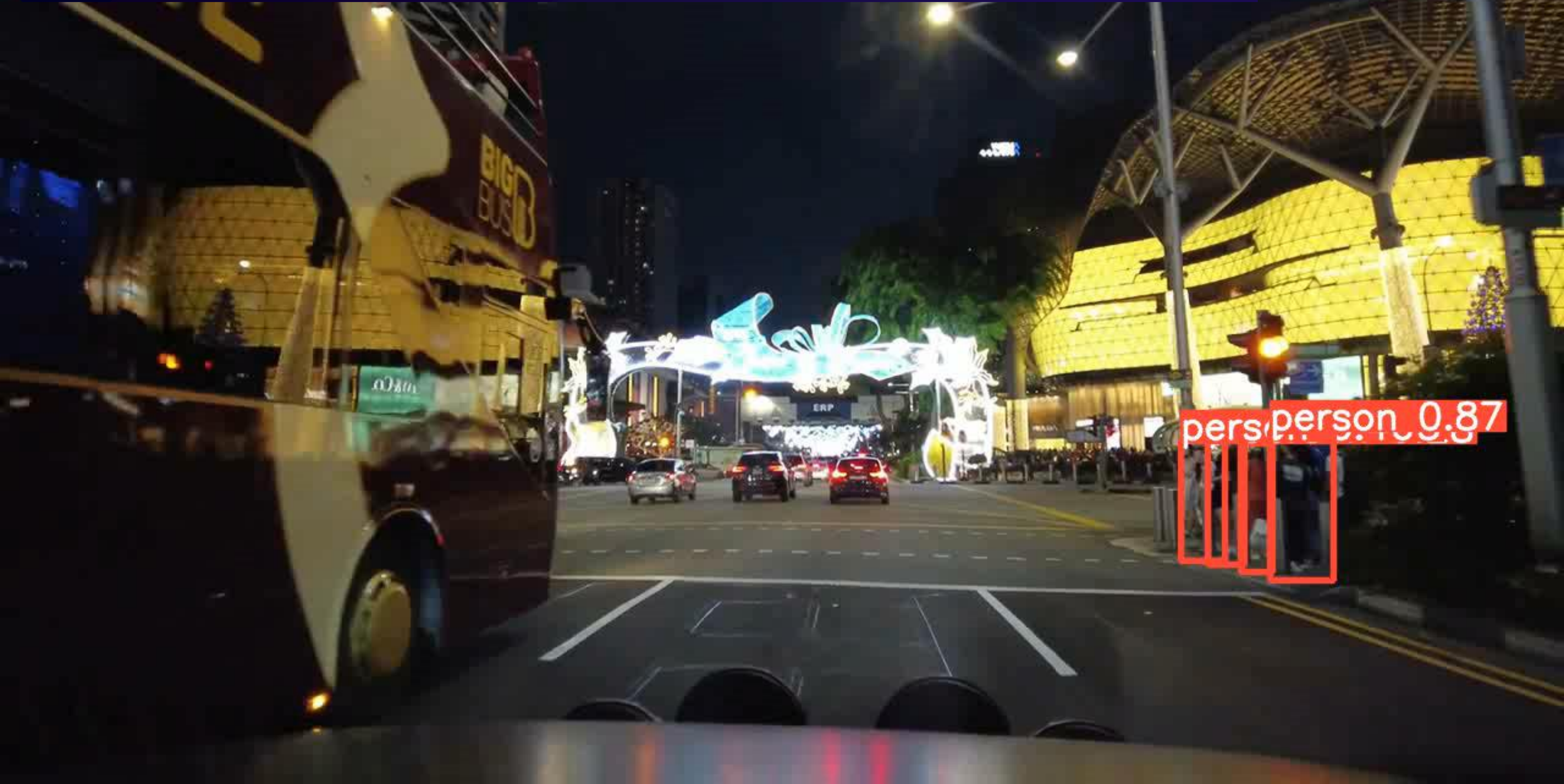
Object Detection (NTU)



Object Detection (Orchard Road - Day)

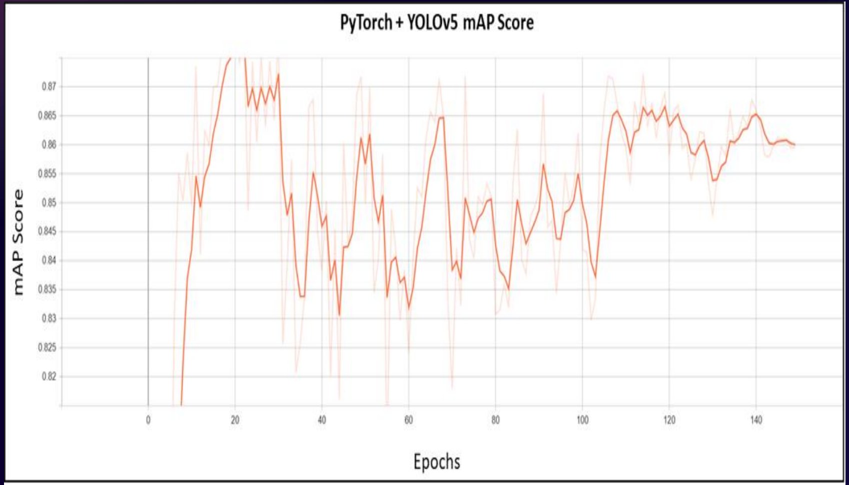


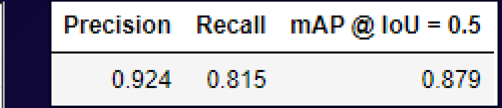
□□□ Object Detection (Orchard Road - Night)



Model Evaluation (Object Detection)

PyTorch + YOLOv5





Precision	Recall	mAP @ IoU = 0.5
0.924	0.815	0.879

- Although model did not meet the 90% criteria set, the mAP of 0.879 signals to a competent model
- Generally, this high average precision translates to a high level of accuracy in predicting actual positives (True Positives)

06

Conclusion

Areas for Improvement

01

DATASET

Use a larger dataset with more variation in person objects

02

MODELLING

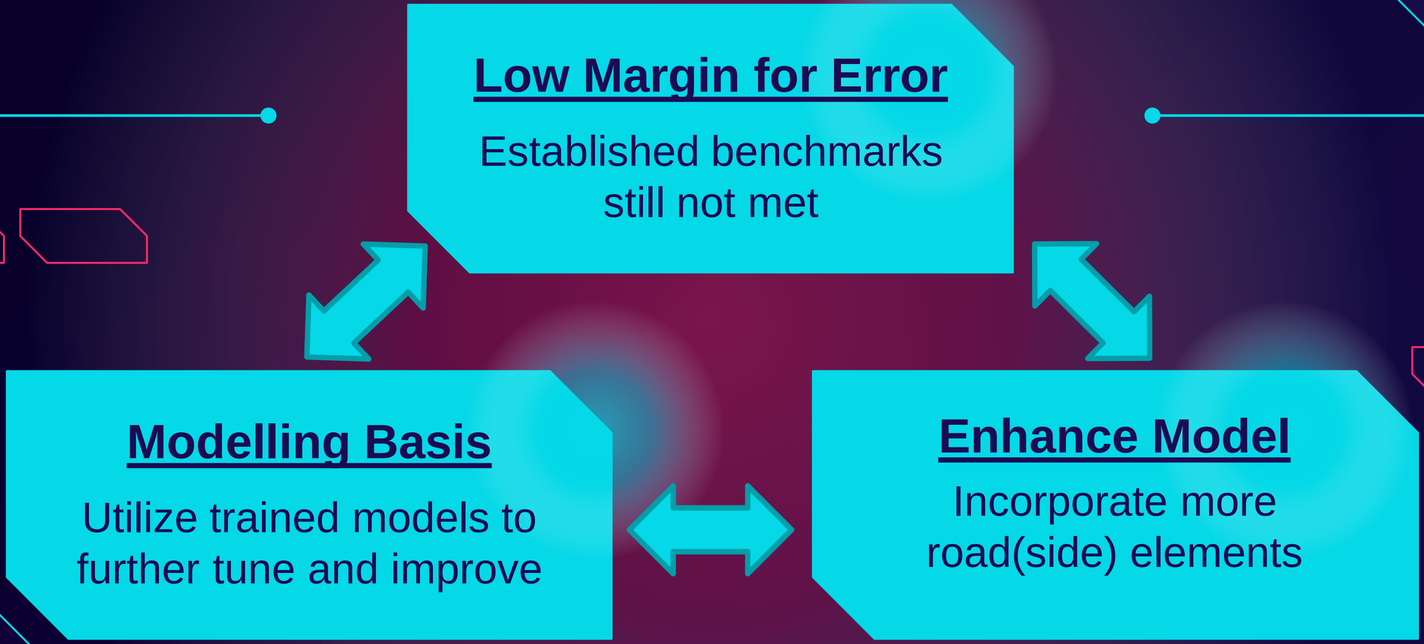
More research on the ‘black box’ to understand how the model classifies/detects

03

ENSEMBLING

Augment performance of CNN models by applying ensembling techniques to form a ‘committee of networks’

Business Recommendations



Low Margin for Error

Established benchmarks still not met

Modelling Basis

Utilize trained models to further tune and improve

Enhance Model

Incorporate more road(side) elements

Thank you!