Object Detection for BDD

By :
Somya Goel

Data Overview

BDD DATASET

As per the BDD paper,

- A good dataset is the driving force for researchers in order to create a model which outperforms the existing methods.
- The data set has 100,000 video clips collected from more than 50,000 rides across diverse weather condition.

Coming to the assignment, the dataset statistics will be further displayed.

Understanding the BDD100K Dataset

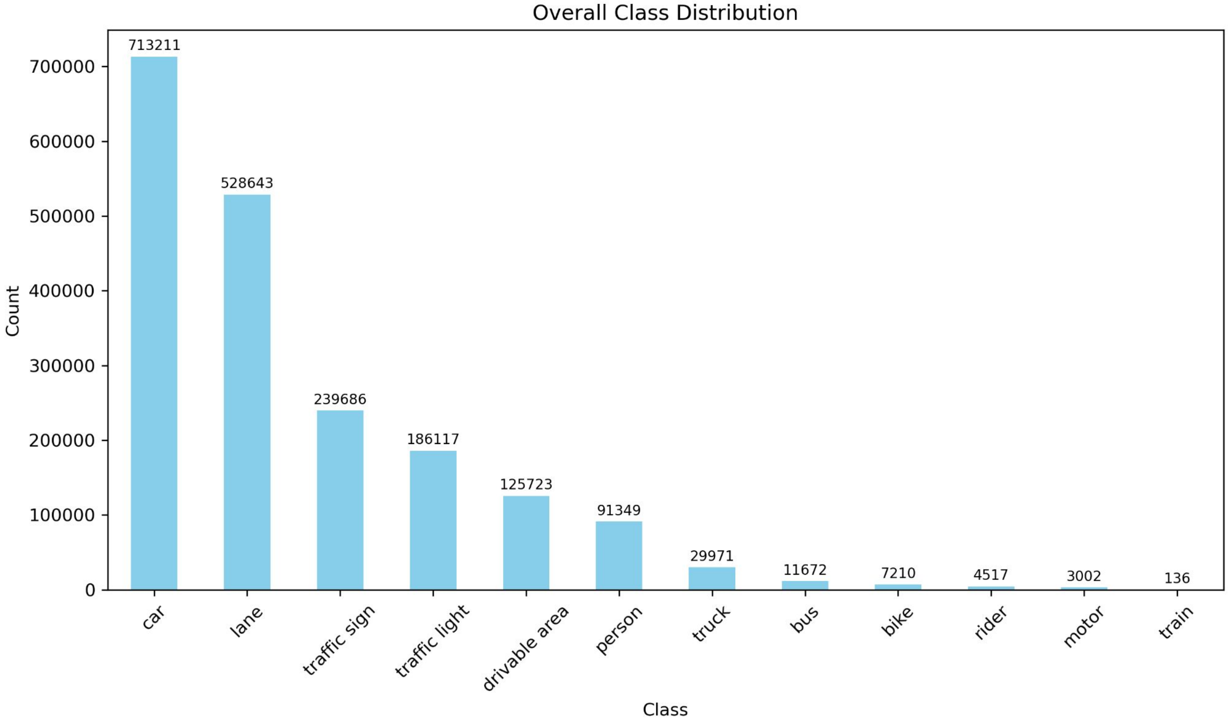
The Berkeley DeepDrive (BDD100K) dataset is one of the most diverse autonomous driving datasets.

- Size: 100,000 videos.
- Diversity: Contains data captured under different weather conditions (sunny, overcast, rainy), time of day (daytime, night), and driving scenarios (highway, urban, rural).
- Annotations: Provides labels for object detection (bounding boxes), drivable area, lane markings, and image segmentation.
- Classes: The object detection labels include 10 classes: `bus`, `traffic light`, `traffic sign`, `person`, `bike`, `truck`, `motor`, `car`, `train`, `rider`.

STATISTICS

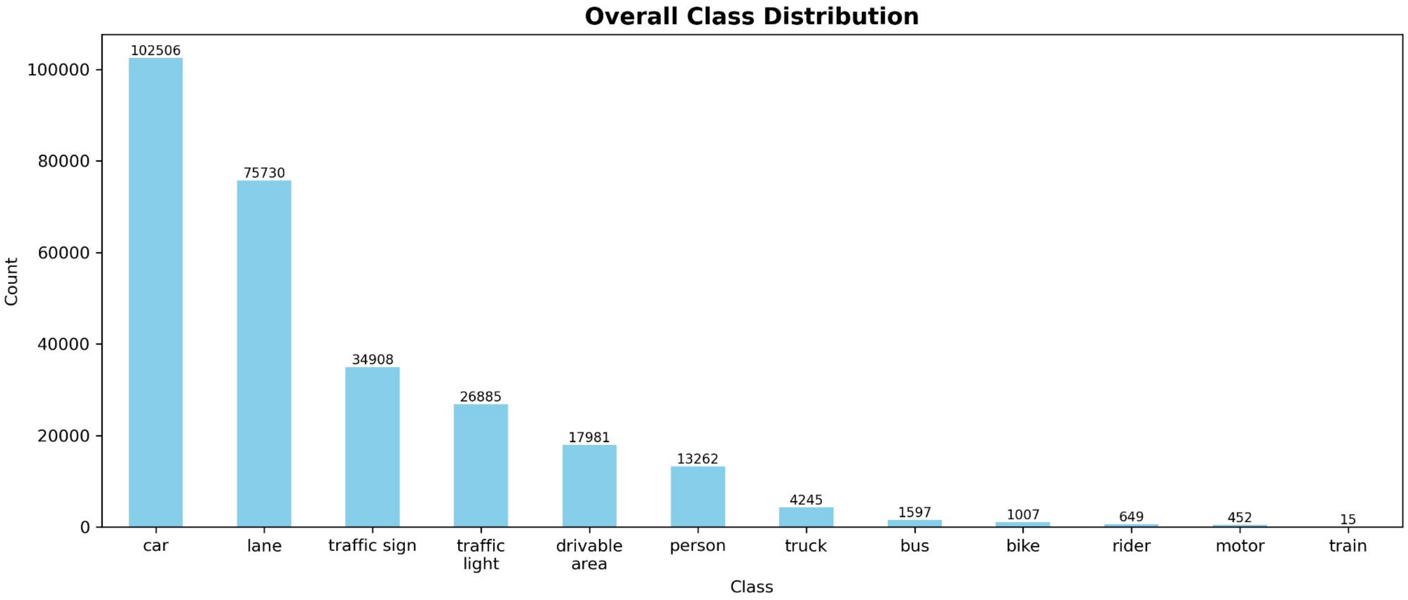
Training Dataset:

- **Total Images: 69863**
- **Total Annotations: 1941237**
- **Unique Classes: 12**



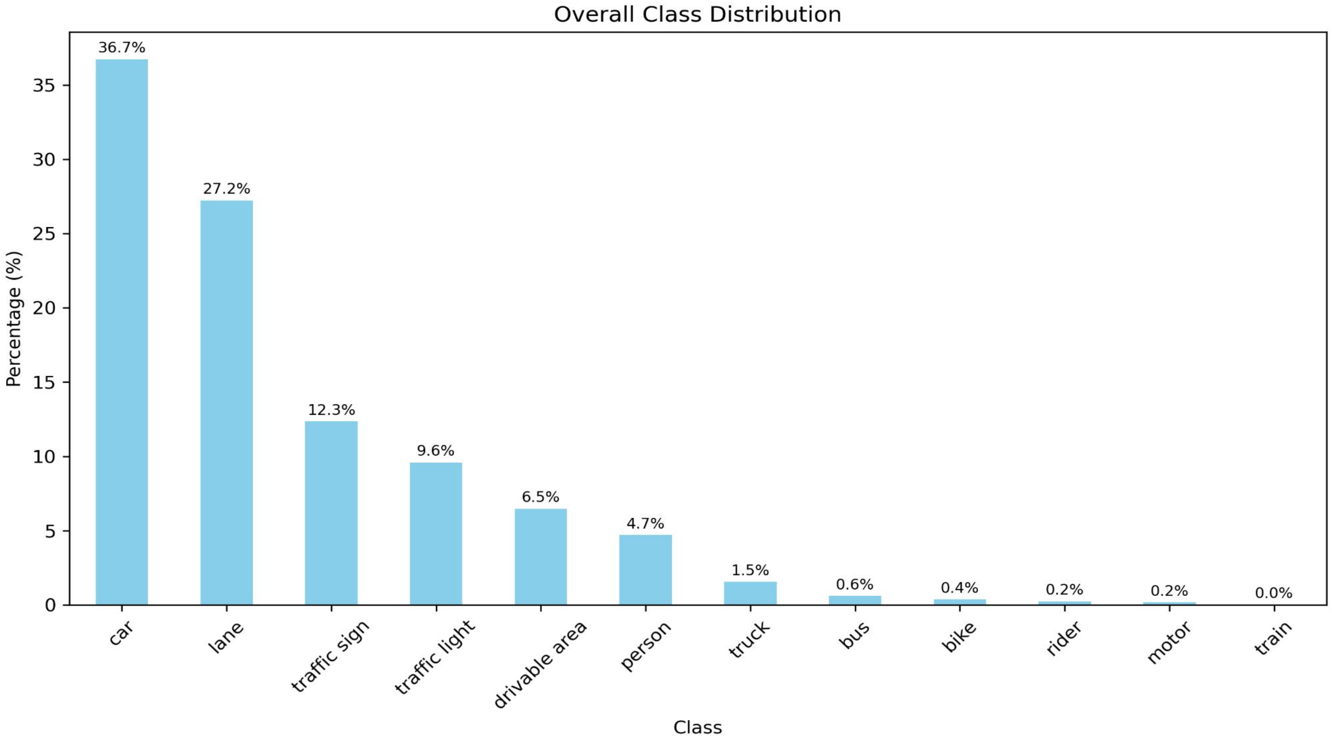
Validation Dataset:

Total Images: 10000
Total Annotations: 279237
Unique Classes: 12

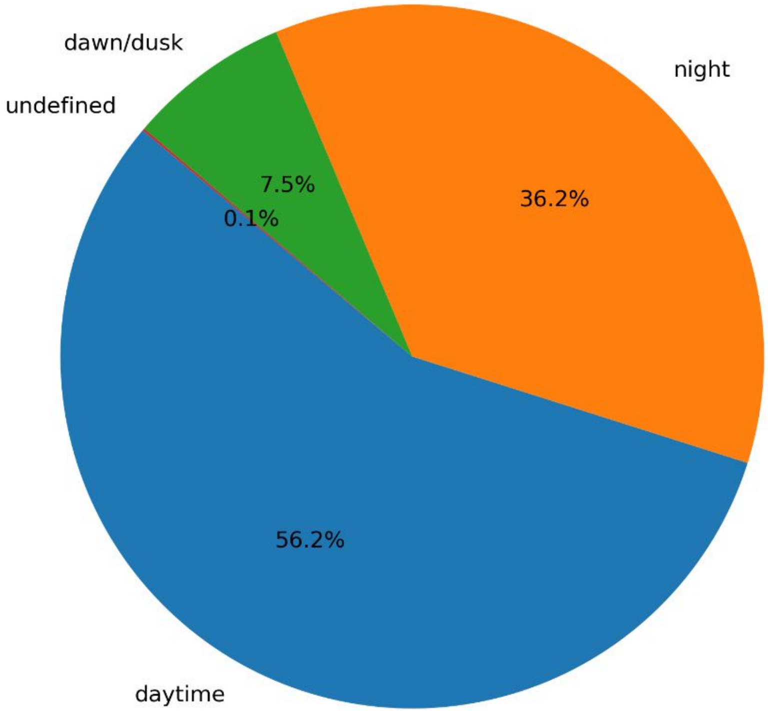


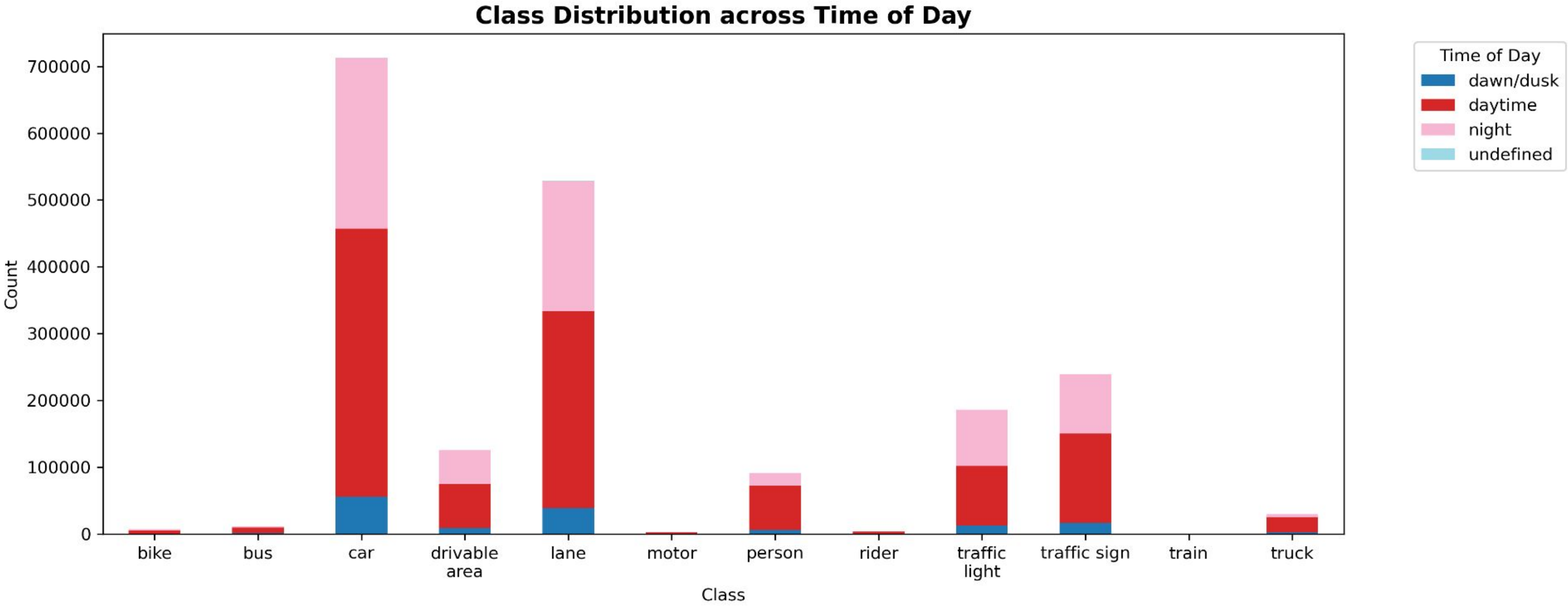
OBSERVATIONS

- Highly imbalance data
- Car and Lane are dominant classes
- Bus, bike , rider , motor , train are less than 1%



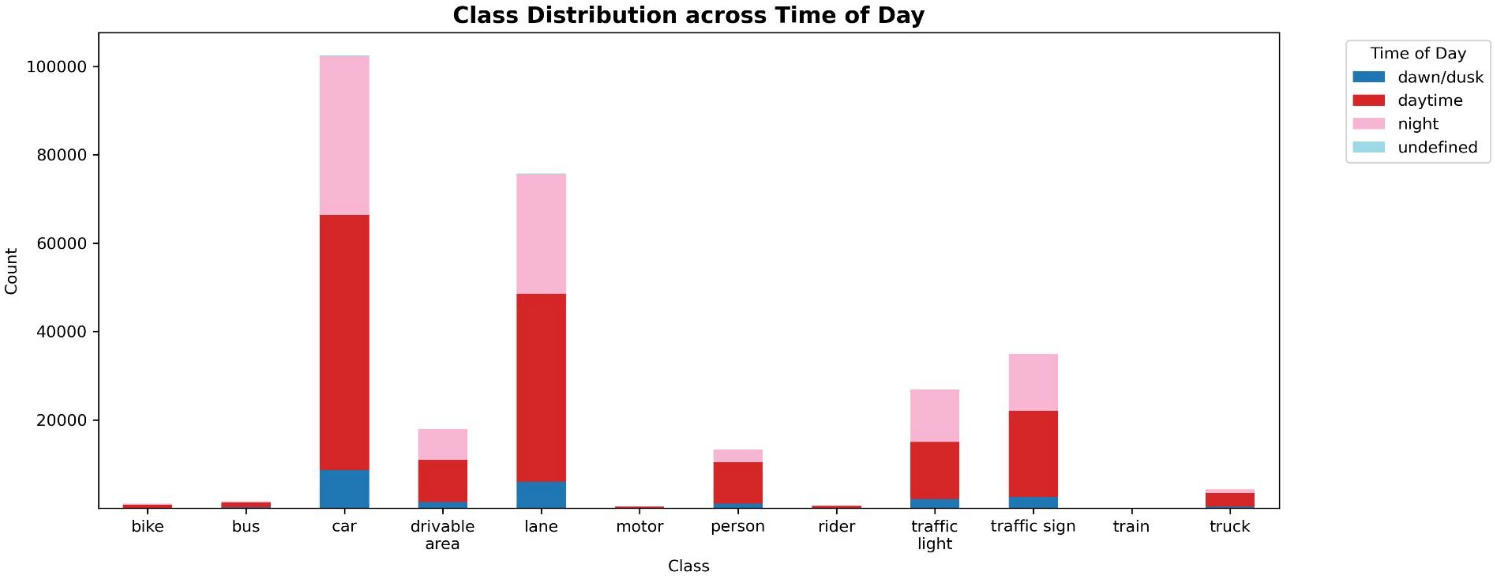
Class Distribution across Time of Day (Normalized) (Pie)

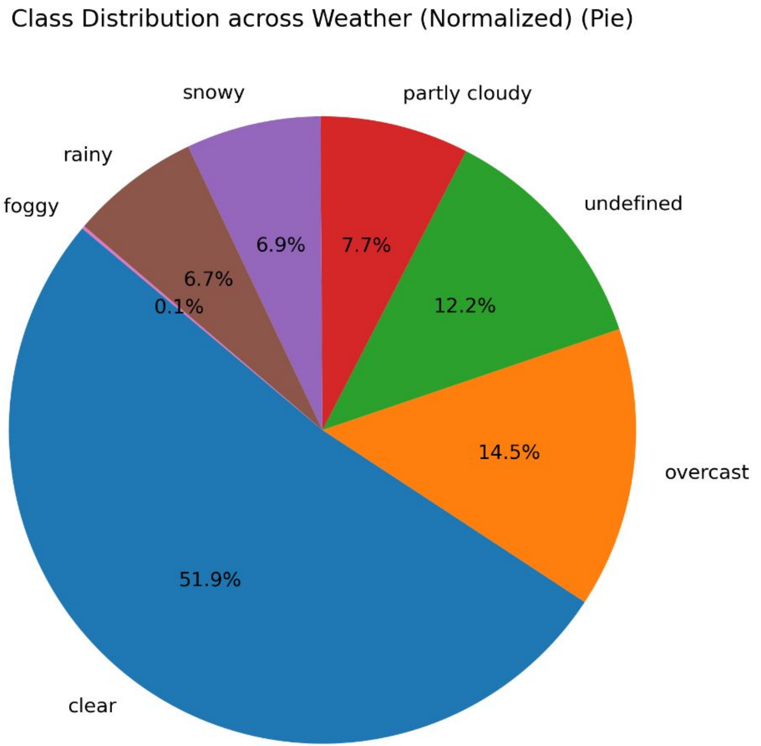




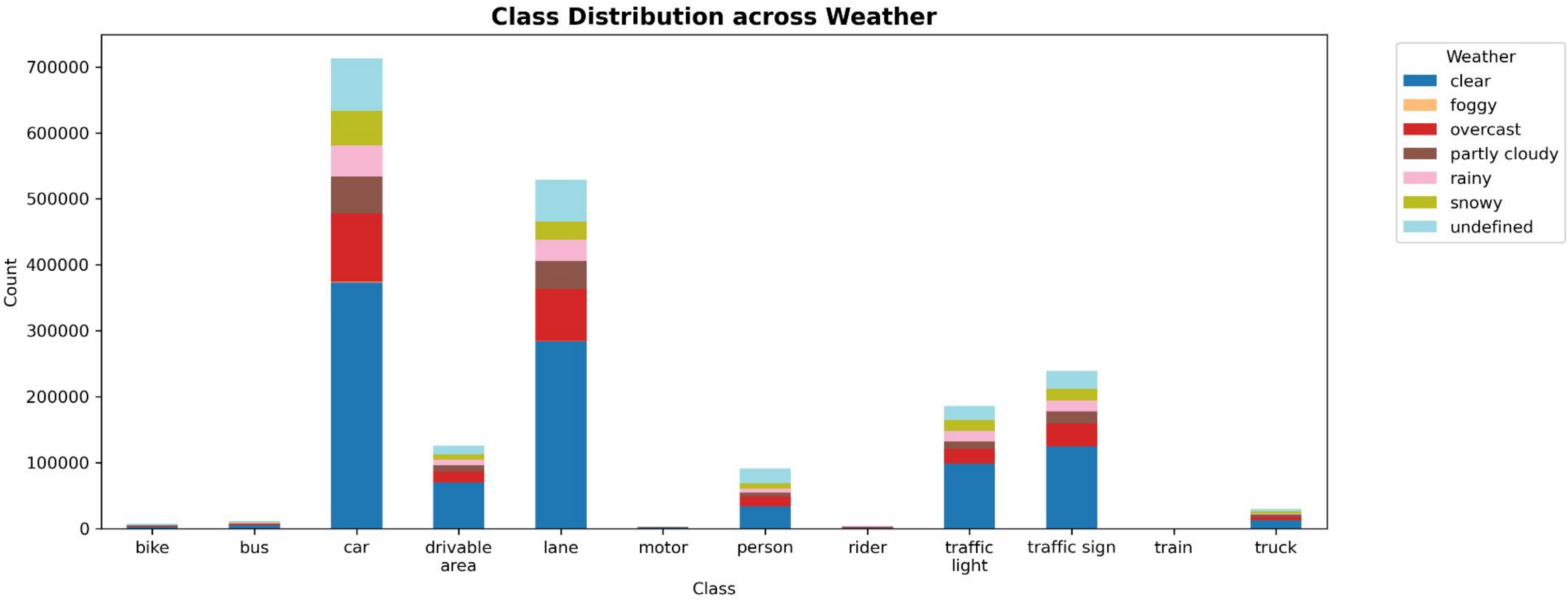
Since the data has been taken from real life scenario, the statistics depict the same. We can see maximum data during the day followed by night. As the dawn/dusk are of shorter span so is the data. The distribution is replicated in validation data too.

Validation Dataset Distribution

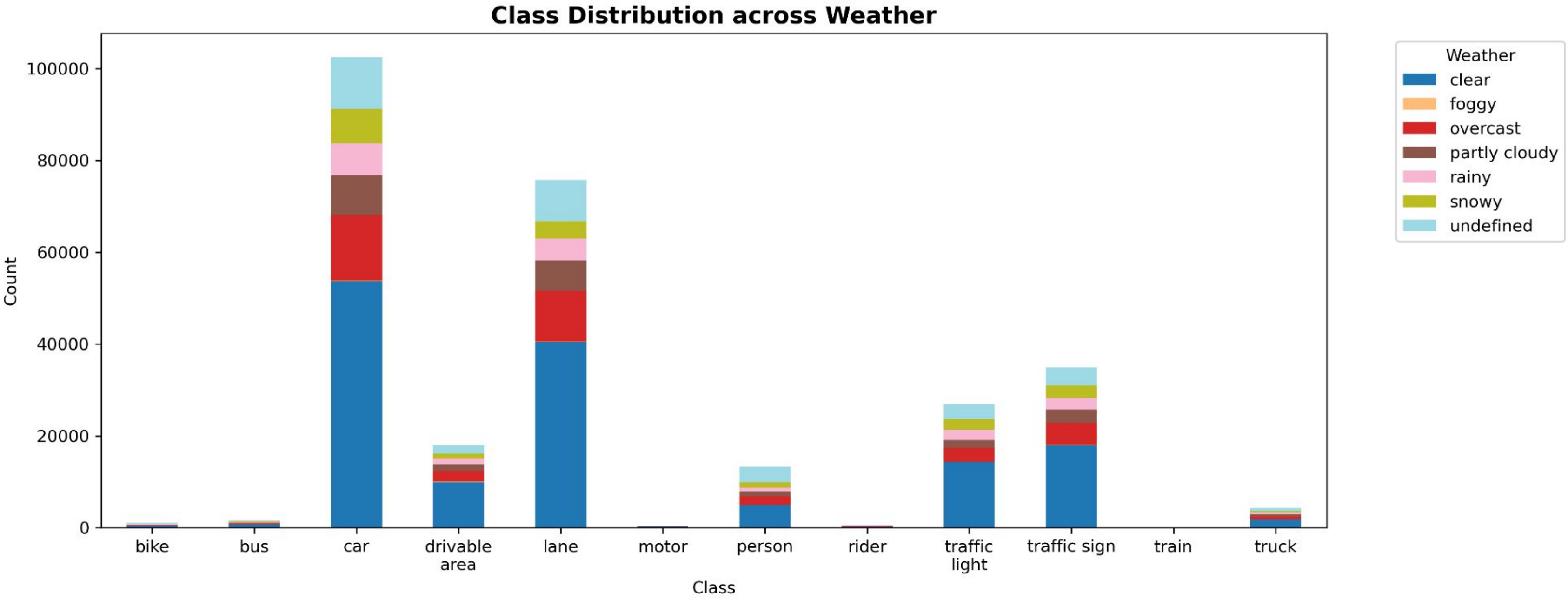


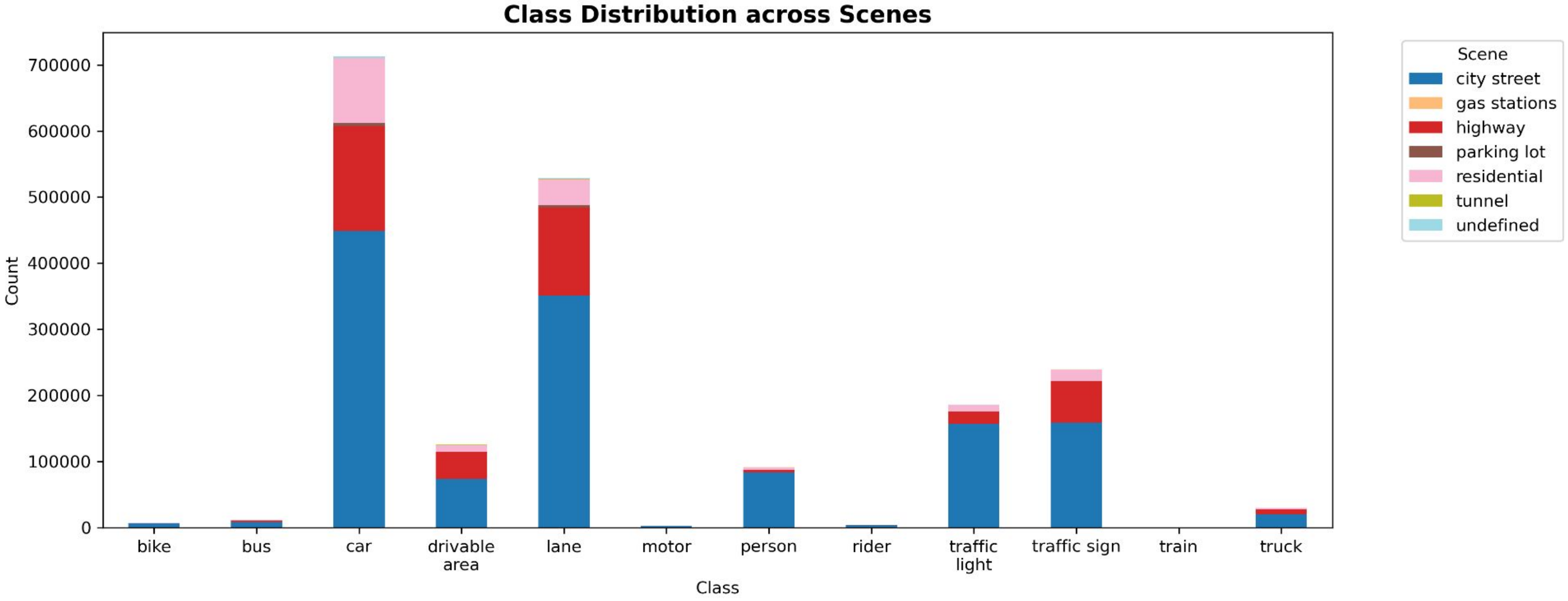


The percent of transactions across various weather conditions can be seen here. Half of the data is in clear weather which is ideal for detection, however, rest 48% cases should also be taken into account which might occur less but are still highly important for the overall product's accuracy.


Class Distribution across Weather
Count
700000
600000
500000
400000
300000
200000
100000
0
bike
bus
car
drivable area
lane
motor
person
rider
traffic light
traffic sign
train
truck
Class
Weather
clear
foggy
overcast
partly cloudy
rainy
snowy
undefined

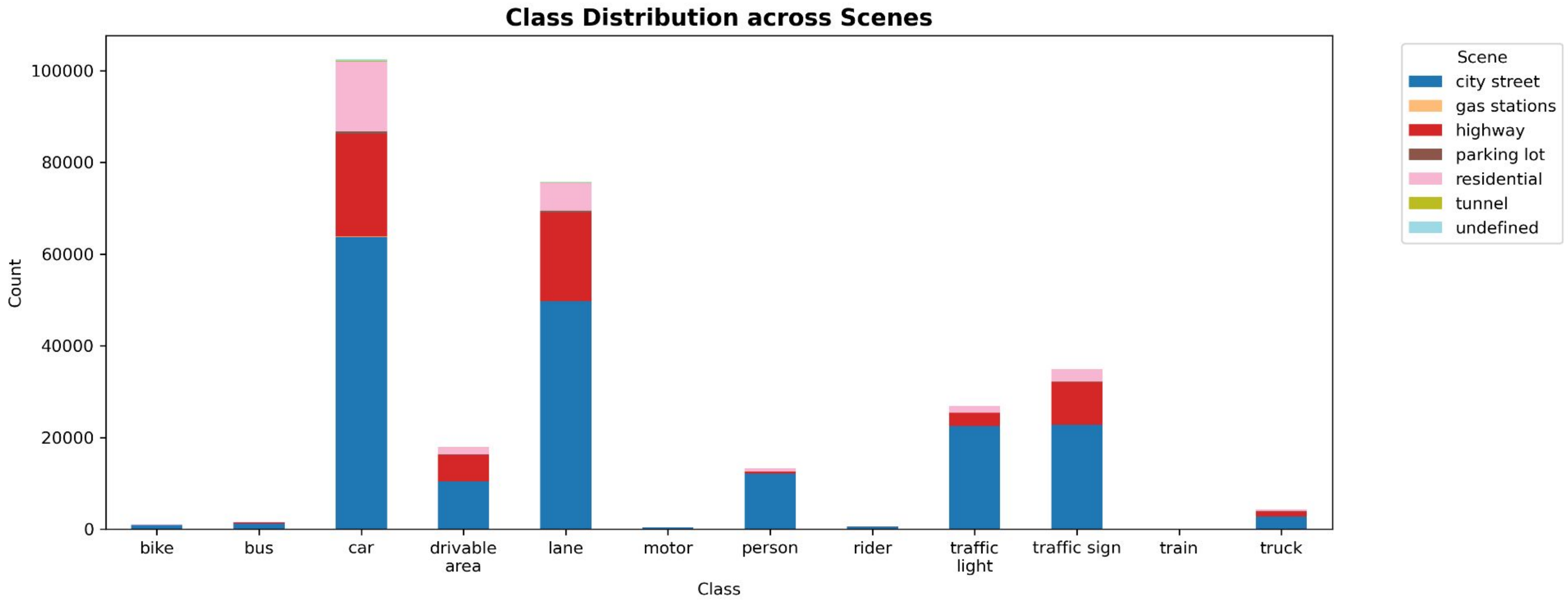
Validation Dataset





The data is biased on city streets followed by highways. However some classes are specific to particular scene.

Validation Dataset



OBSERVATIONS ON DATASET

- The data is highly imbalance
- Some cases of mislabelling are observed
- The class wise distribution across different conditions : Time of day, weather , scenes, etc is uniformly distributed among train and validation set.

MODEL SELECTION AND TRAINING

Model Selection And Training

- The model selected for object detection here is yolo v8
- The reason behind the selection includes :
  - The model outperforms most of the sota methods for object detection
  - It is user friendly method when used with Ultralytics library, even a simple CLI works easily.
  - Deployment Friendly : ultralytics allows CLI commands to directly convert models in openvino / onnx frameworks.
- The diversity in dataset as mentioned in previous section is precisely the reason why fine-tuning a powerful model like YOLOv8 on it is valuable—it teaches the model to be robust to real-world variations.
- The BDD git page has a comparison of all the SOTA models.

Model Selection : Yolov8 - small

- Hardware constraints
- Dataset characteristics (extreme class imbalance in BDD100K)
- Need for real-time performance (autonomous driving application)
- Training efficiency requirements (reasonable training times)

YOLOv8s provides the optimal balance of accuracy, speed, and resource usage that makes it uniquely suited for your BDD100K training project. The architectural improvements in v8 specifically address the challenges of autonomous driving datasets better than previous versions or competing architectures.

YOLOv8 includes advanced augmentation techniques specifically beneficial for driving datasets:

- Advanced mosaic and mixup augmentations that help with rare classes
- Better handling of varying weather and lighting conditions in BDD100K
- Improved loss functions that work better with imbalanced datasets

YOLOv8 introduces an anchor-free split head that significantly improves performance on diverse object scales present in BDD100K:

- Better handles the extreme class imbalance (713k cars vs. 136 trains)
- Improved detection of small objects (distant traffic signs, pedestrians)

To take care of the data imbalance , few training strategies have been implemented, given in next slide.

Implementation Details

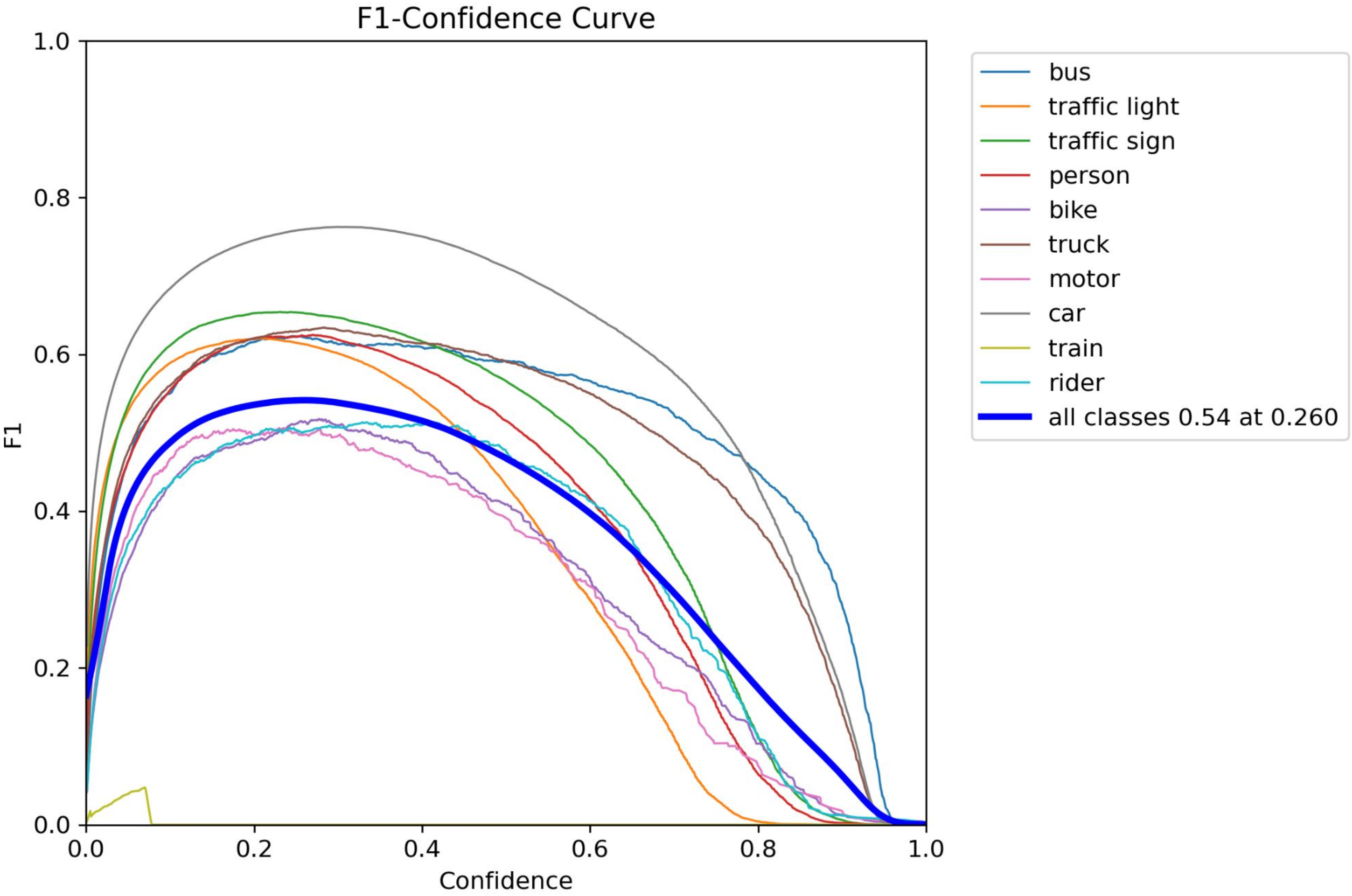
Gradient Accumulation

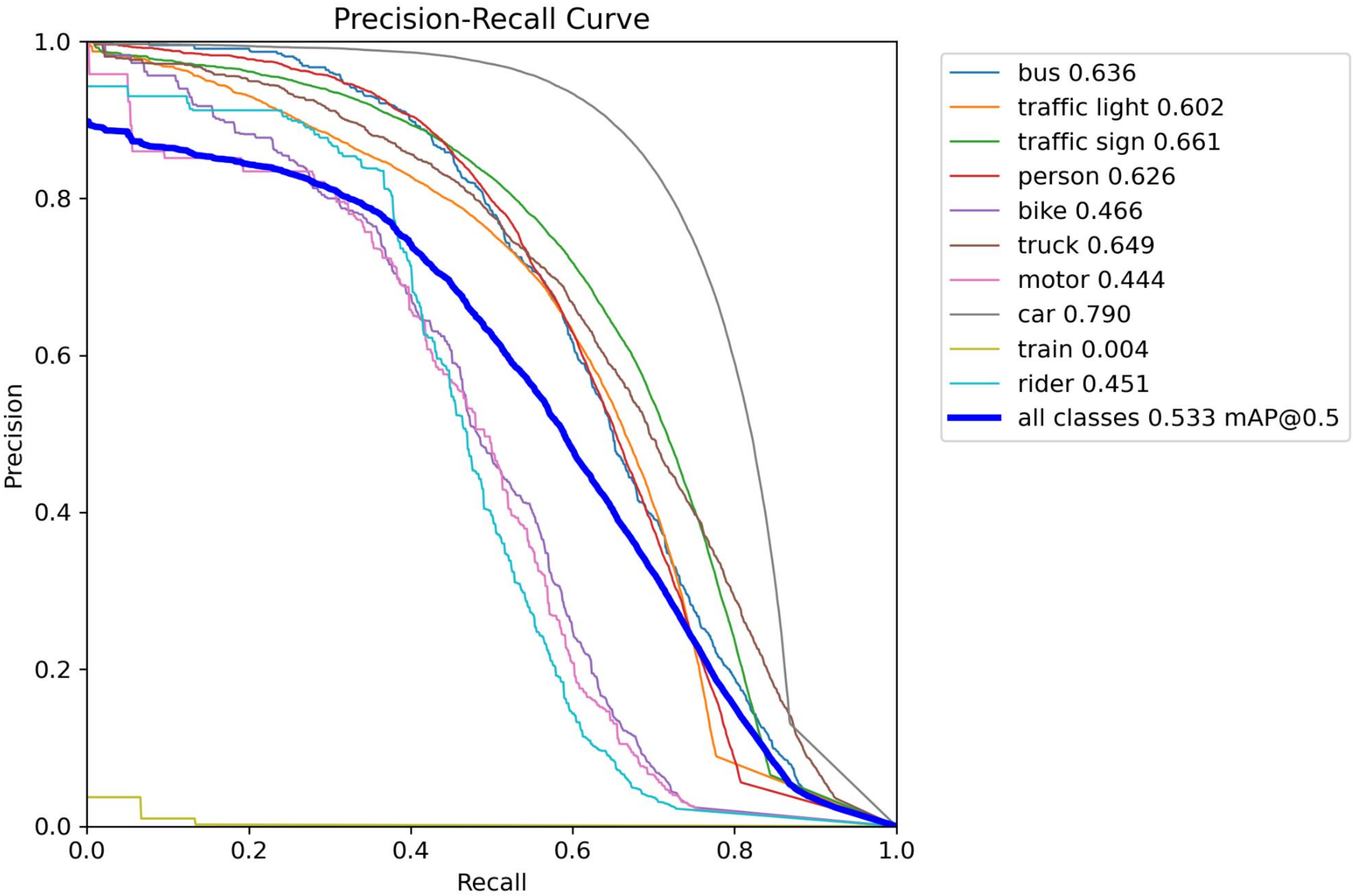
- Uses `accumulation_steps = 4` with `batch_size = 4` to simulate an effective batch size of 16
- Accumulates gradients over multiple batches before updating weights
- Scales the loss by `1/accumulation_steps` to maintain correct gradient magnitudes

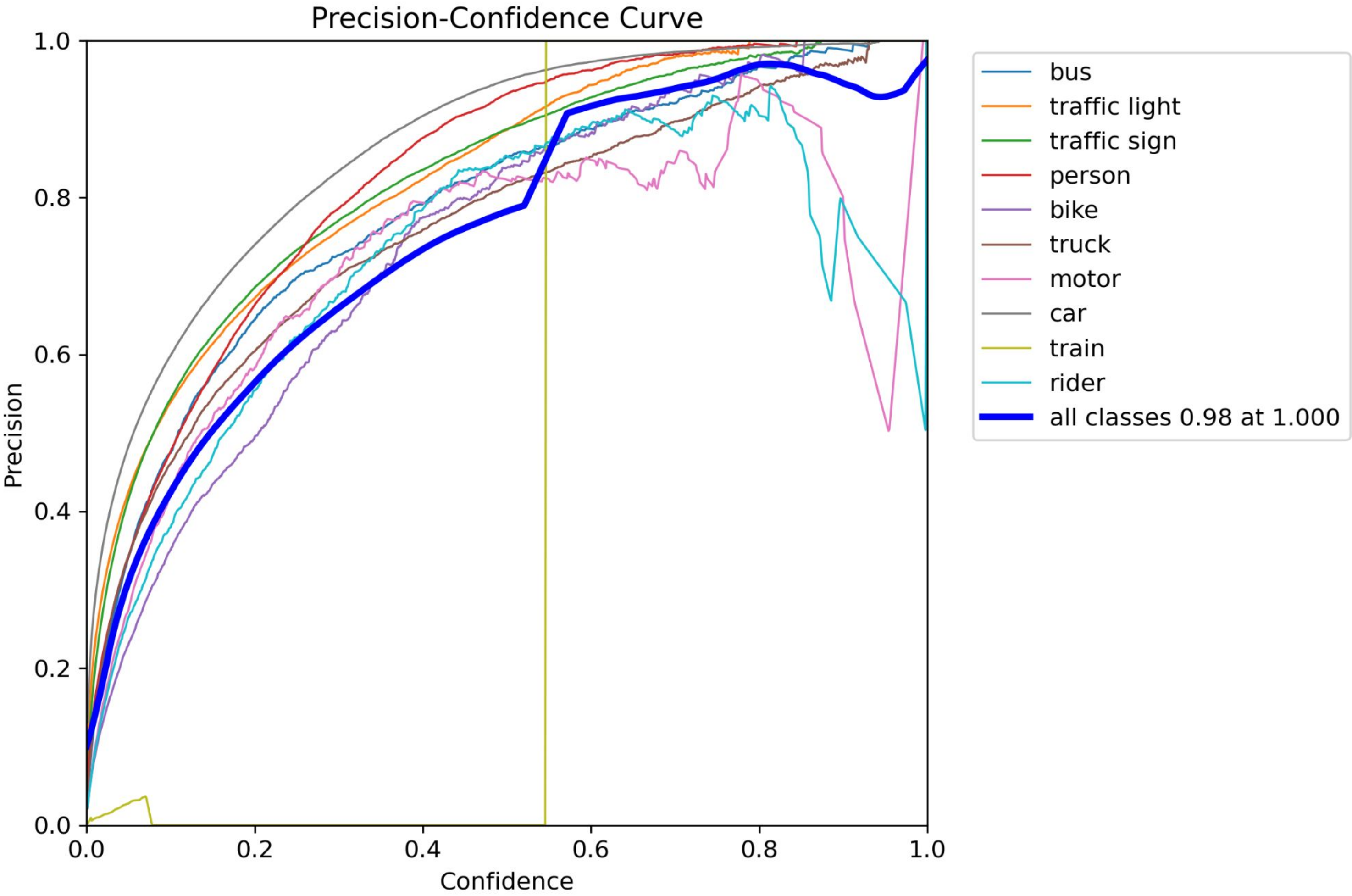
Custom Training Loop

- Uses AdamW optimizer with Cosine Annealing learning rate scheduler
- Model trained on total of 150 epochs

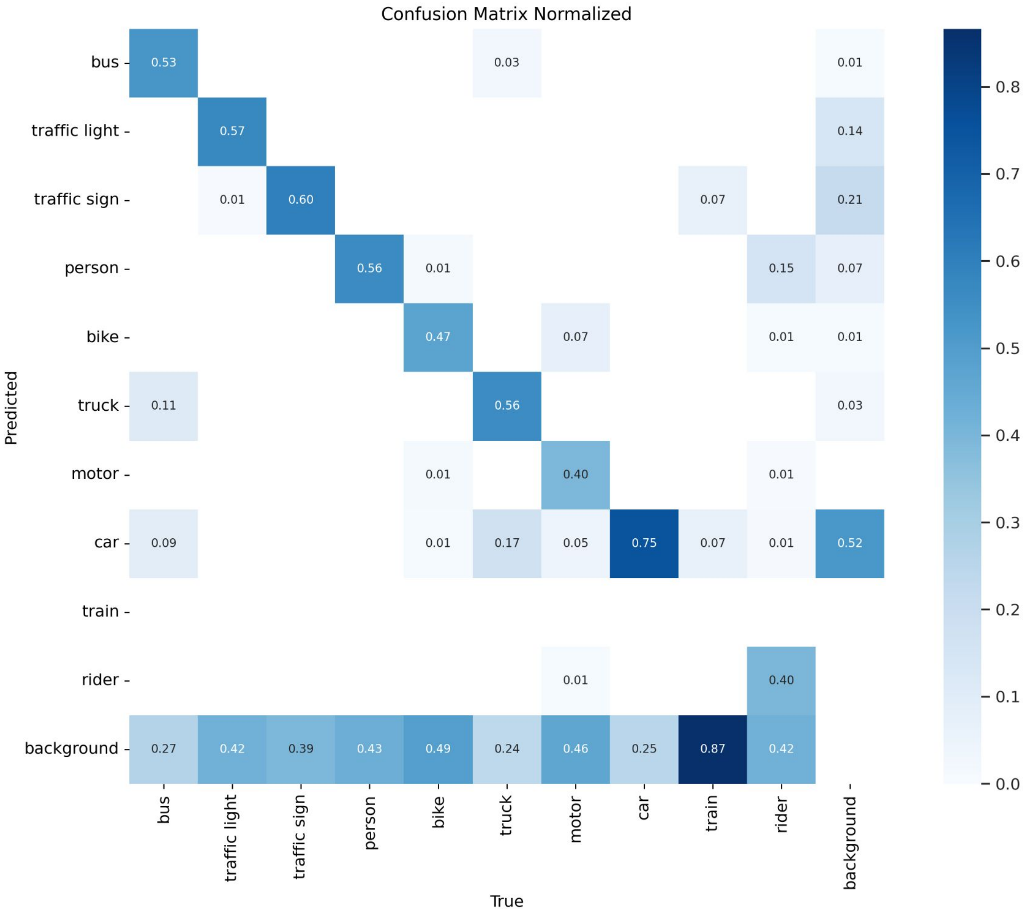
TRAINING CURVES


F1-Confidence Curve
1.0
0.8
0.6
0.4
0.2
0.0
F1
0.0
0.2
0.4
0.6
0.8
1.0
Confidence
bus
traffic light
traffic sign
person
bike
truck
motor
car
train
rider
all classes 0.54 at 0.260


Precision-Recall Curve
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
bus 0.636
traffic light 0.602
traffic sign 0.661
person 0.626
bike 0.466
truck 0.649
motor 0.444
car 0.790
train 0.004
rider 0.451
all classes 0.533 mAP@0.5

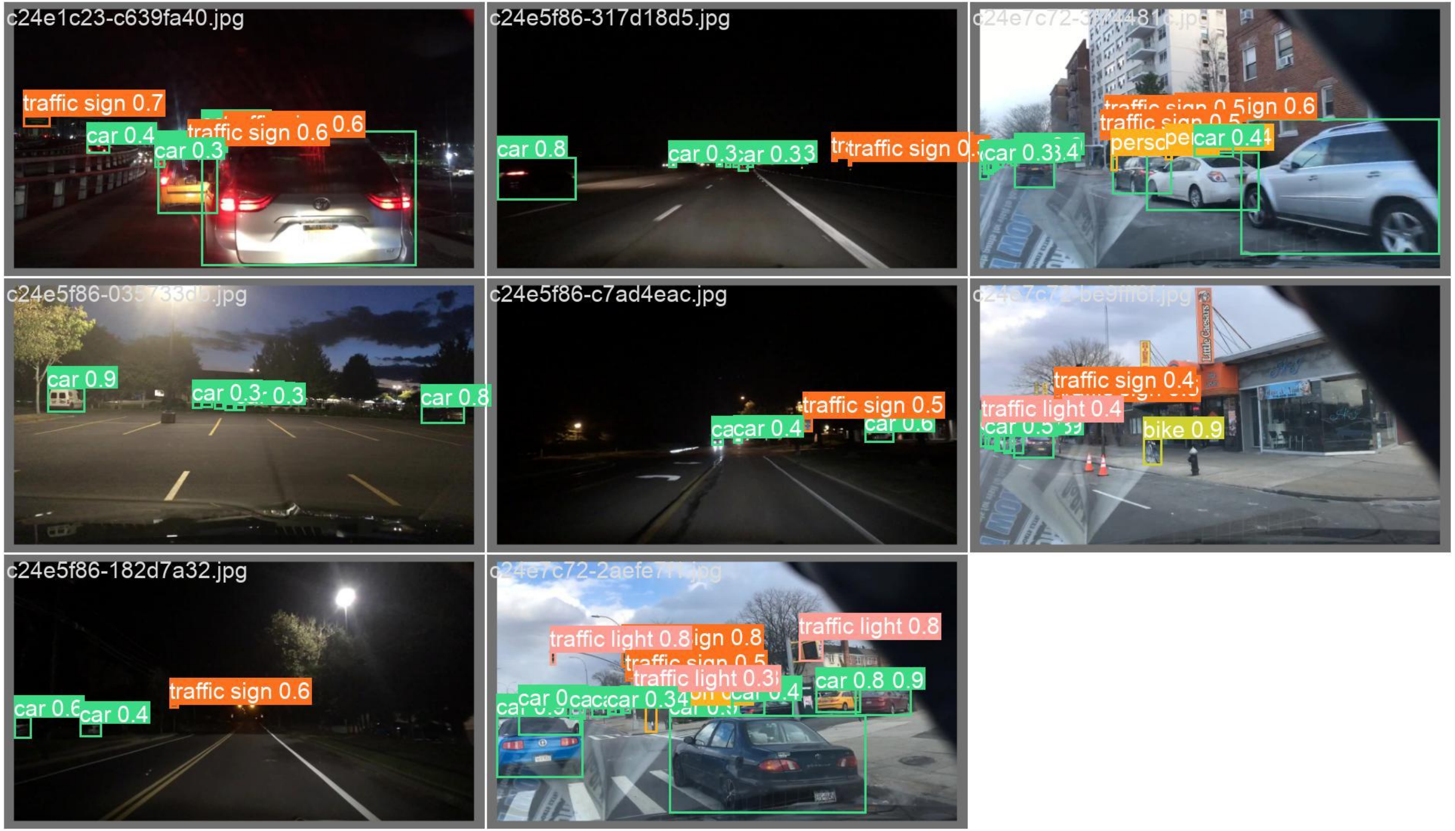

Precision-Confidence Curve
Precision
Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
bus
traffic light
traffic sign
person
bike
truck
motor
car
train
rider
all classes 0.98 at 1.000

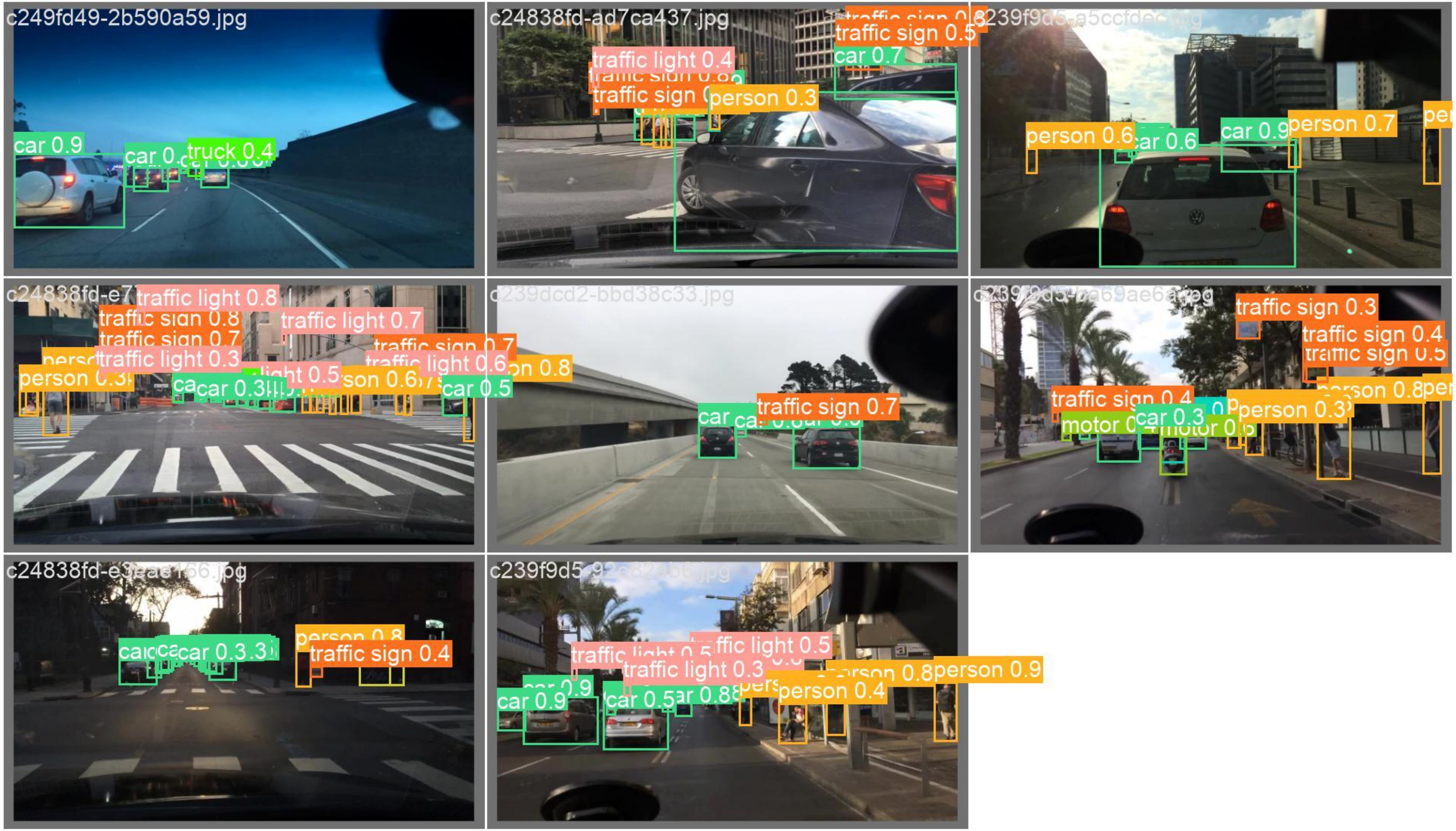
Confusion Matrix



	bus	traffic light	traffic sign	person	bike	truck	motor	car	train	rider	background
bus	0.53					0.03					0.01
traffic light		0.57									0.14
traffic sign		0.01	0.60						0.07		0.21
person				0.56	0.01					0.15	0.07
bike					0.47		0.07			0.01	0.01
truck	0.11					0.56					0.03
motor					0.01		0.40			0.01	
car	0.09				0.01	0.17	0.05	0.75	0.07	0.01	0.52
train											
rider							0.01			0.40	
background	0.27	0.42	0.39	0.43	0.49	0.24	0.46	0.25	0.87	0.42	

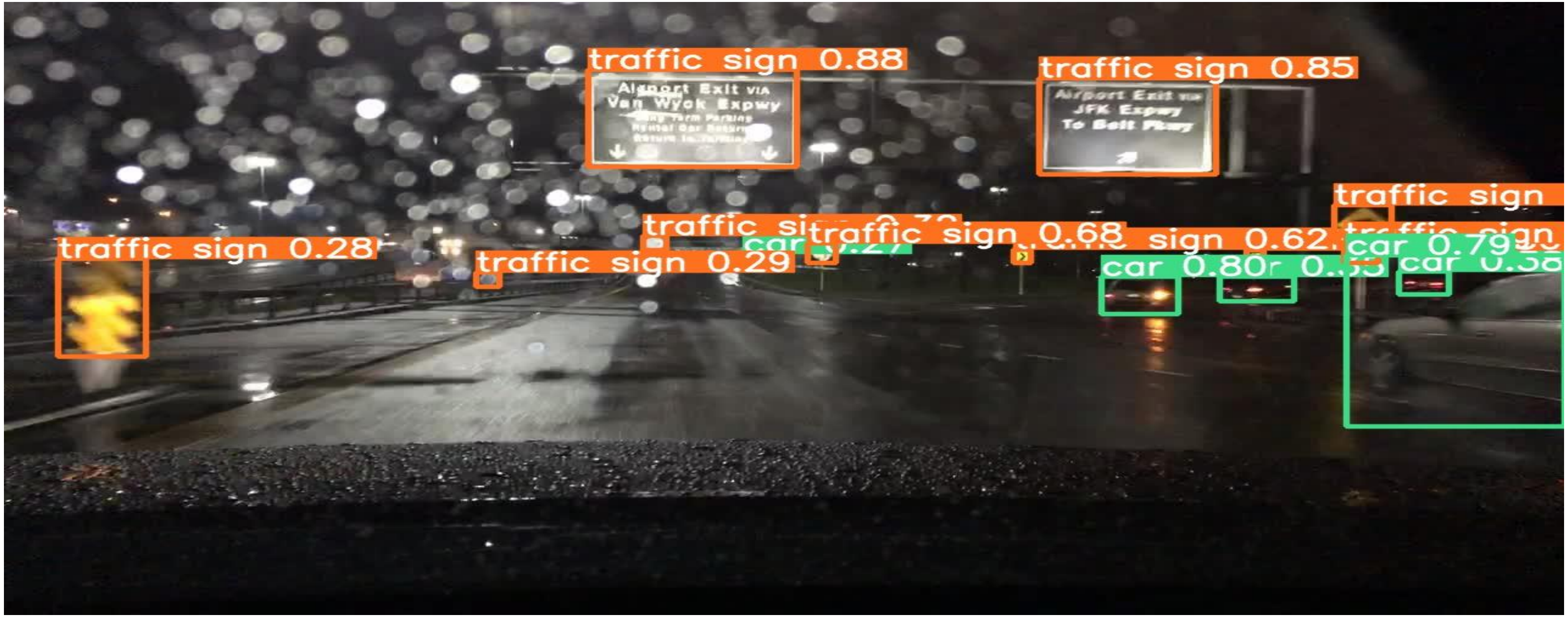
Sample Results



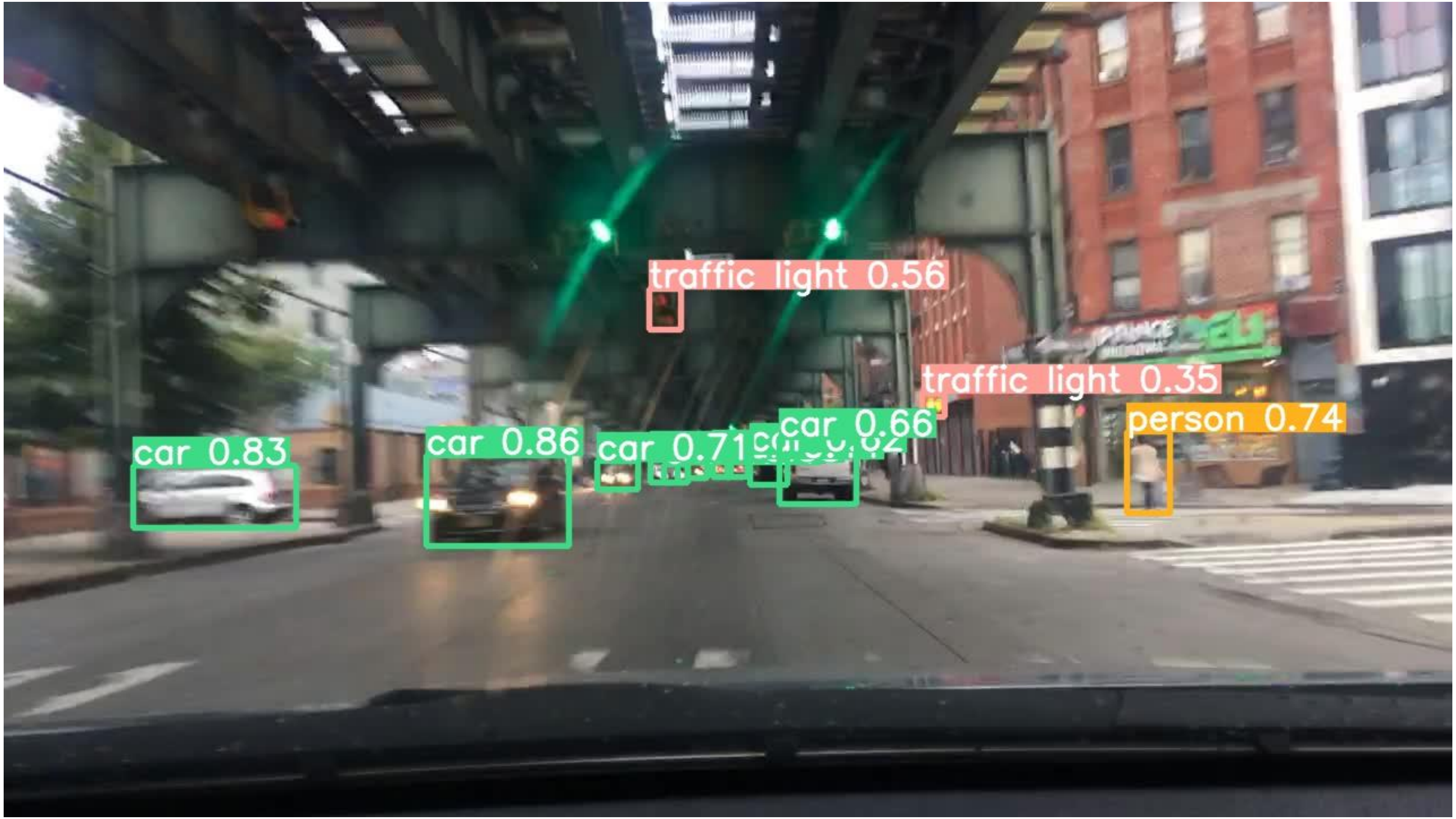

c249fd49-2b590a59.jpg
car 0.9
car 0.
truck 0.4
c24838fd-ad7ca437.jpg
traffic sign 0.5
car 0.7
traffic light 0.4
traffic sign 0.8
traffic sign
person 0.3
c239f9d5-a5ccfdec.jpg
person 0.6
car 0.6
car 0.9
person 0.7
per
c24838fd-e7
traffic light 0.8
traffic sign 0.8
traffic sign 0.7
traffic light 0.7
traffic sign 0.7
traffic light 0.3
traffic light 0.6
person 0.3
car 0.3
light 0.5
son 0.6
car 0.5
on 0.8
c239dcd2-bbd38c33.jpg
car
traffic sign 0.7
c239f9d5-b969ae6a.jpg
traffic sign 0.3
traffic sign 0.4
traffic sign 0.5
traffic sign 0.4
person 0.8
per
motor 0
car 0.3
motor 0.5
person 0.3
c24838fd-e3eae166.jpg
car 0.3
person 0.8
traffic sign 0.4
c239f9d5-92e82eb6.jpg
traffic light 0.5
traffic light 0.5
traffic light 0.3
person 0.8
person 0.9
car 0.9
car 0.9
car 0.5
car 0.8
person 0.4

MODEL EVALUATION

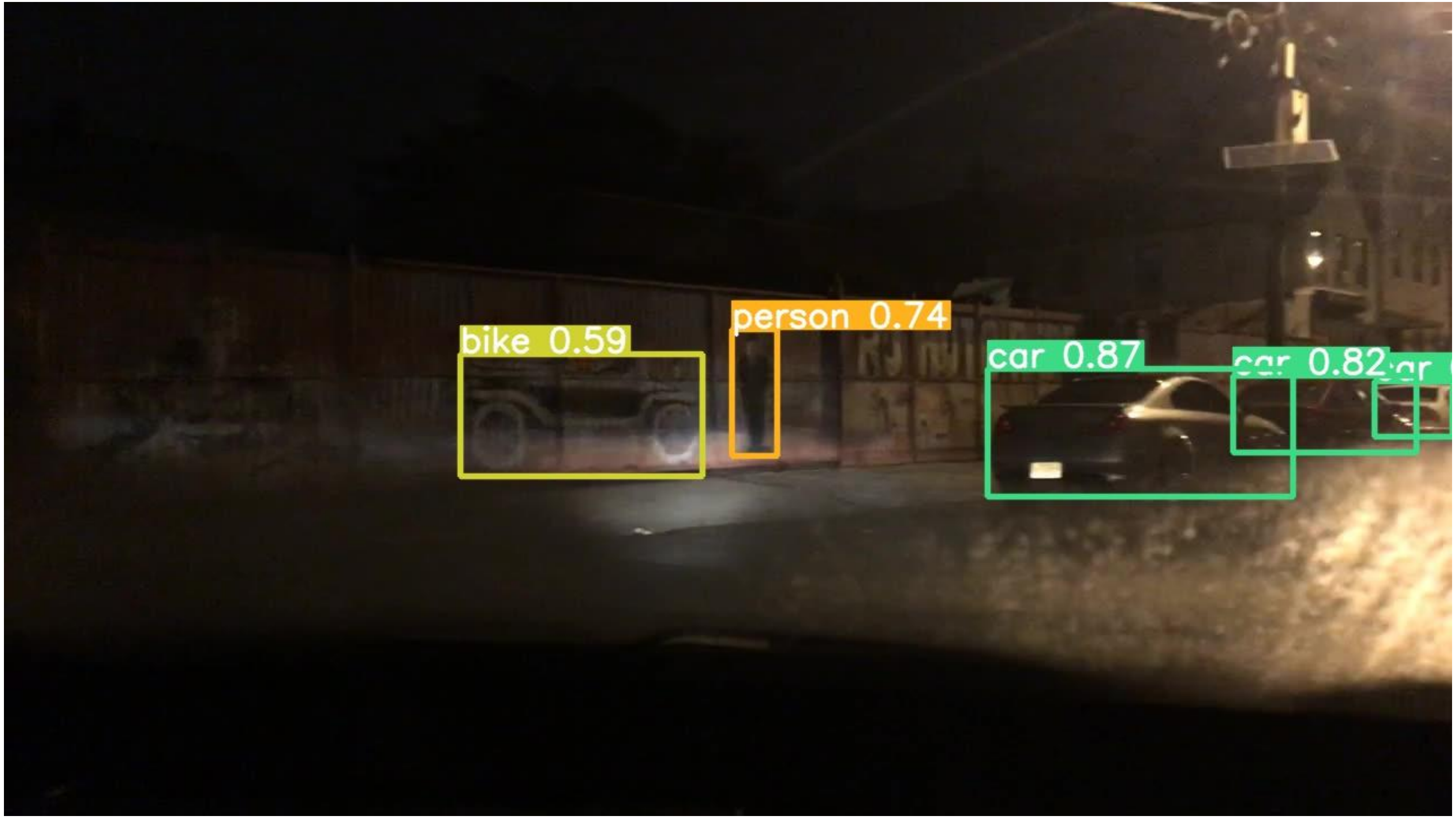
A blurred bike appearing as traffic sign, however the confidence is low.



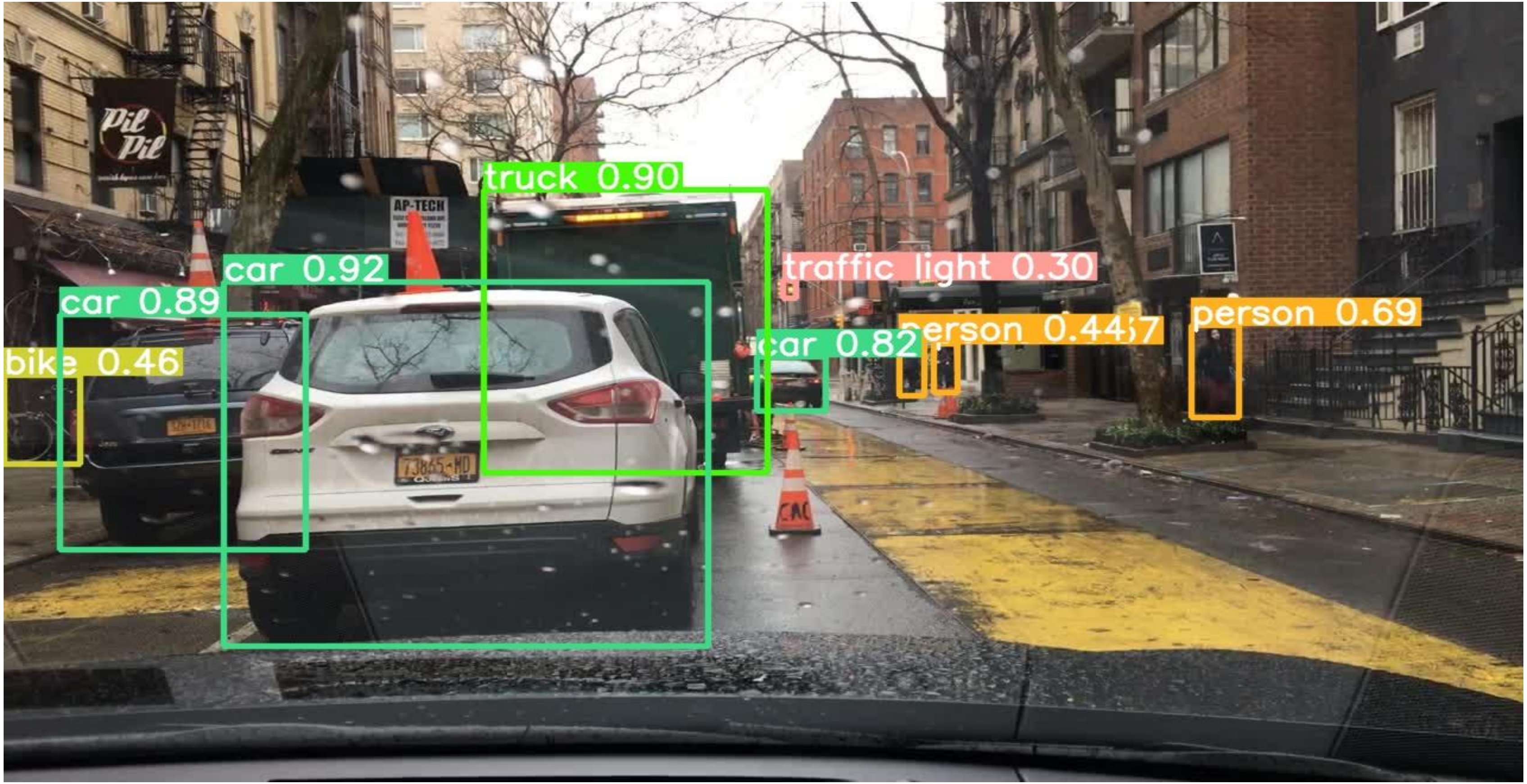
Missing 3 Traffic Lights , detection accuracy of cars is quite high.
However, the image is blurry.



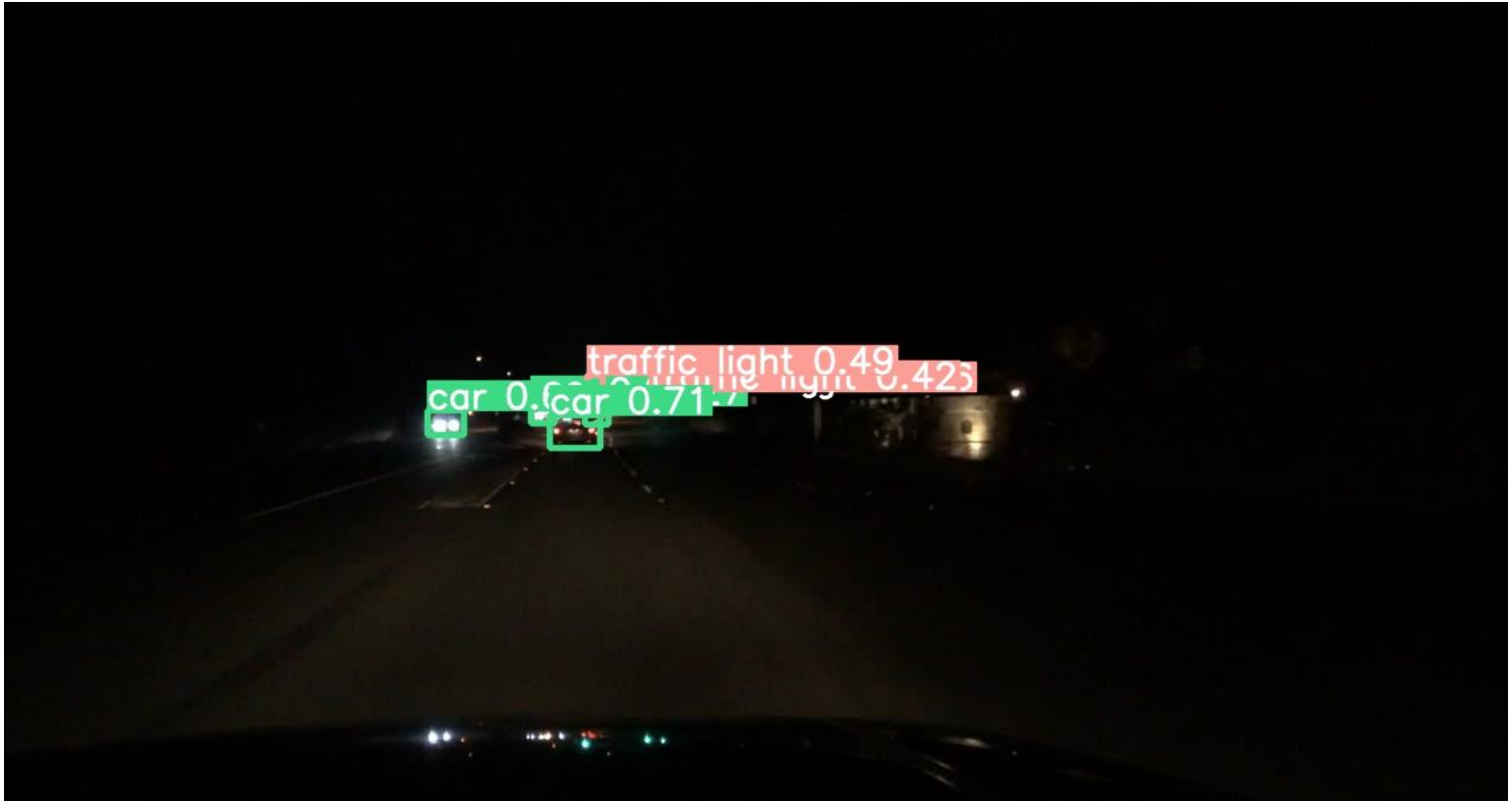
Bike painted on wall is also detected as vehicle.



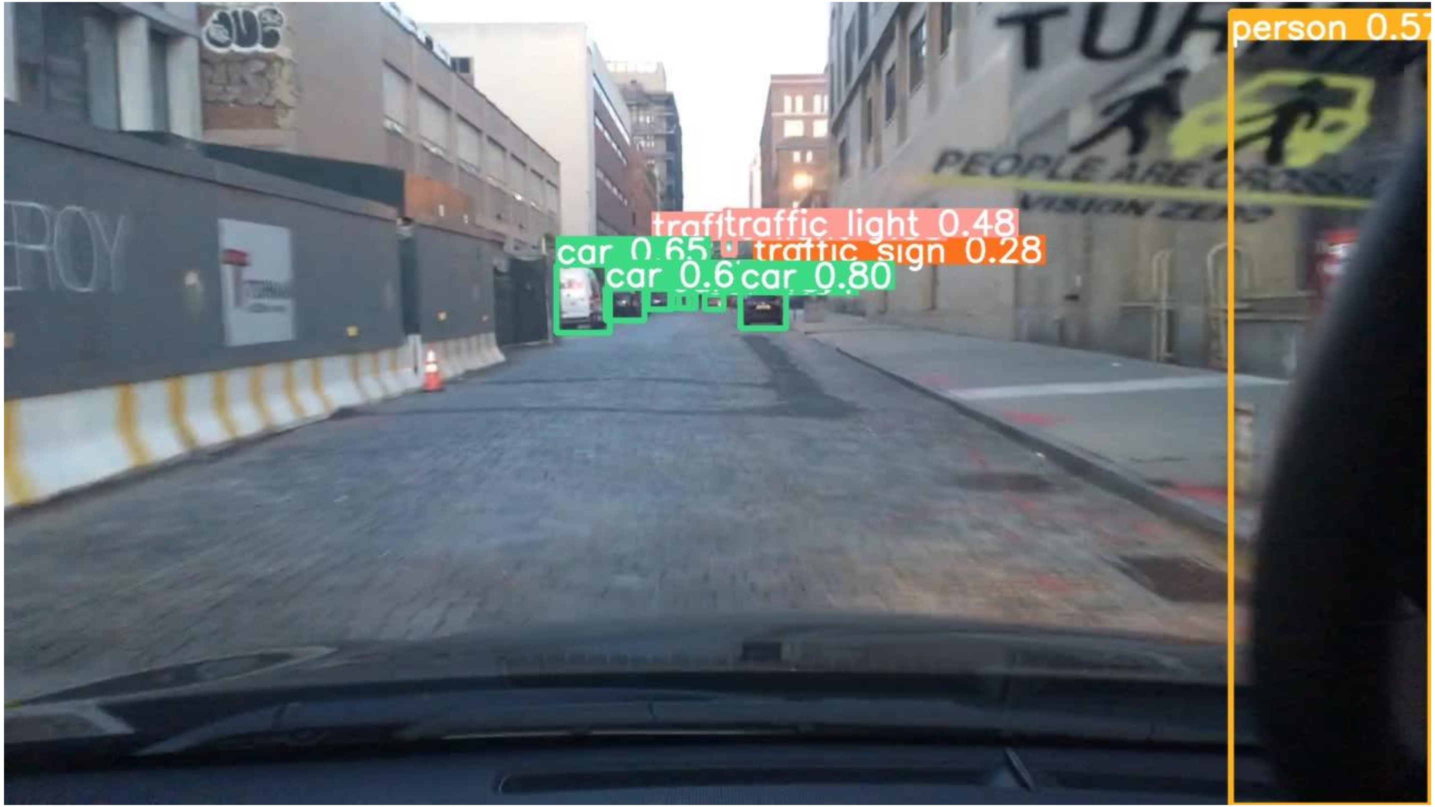
Even partially visible bike is detected even though it has very less samples.



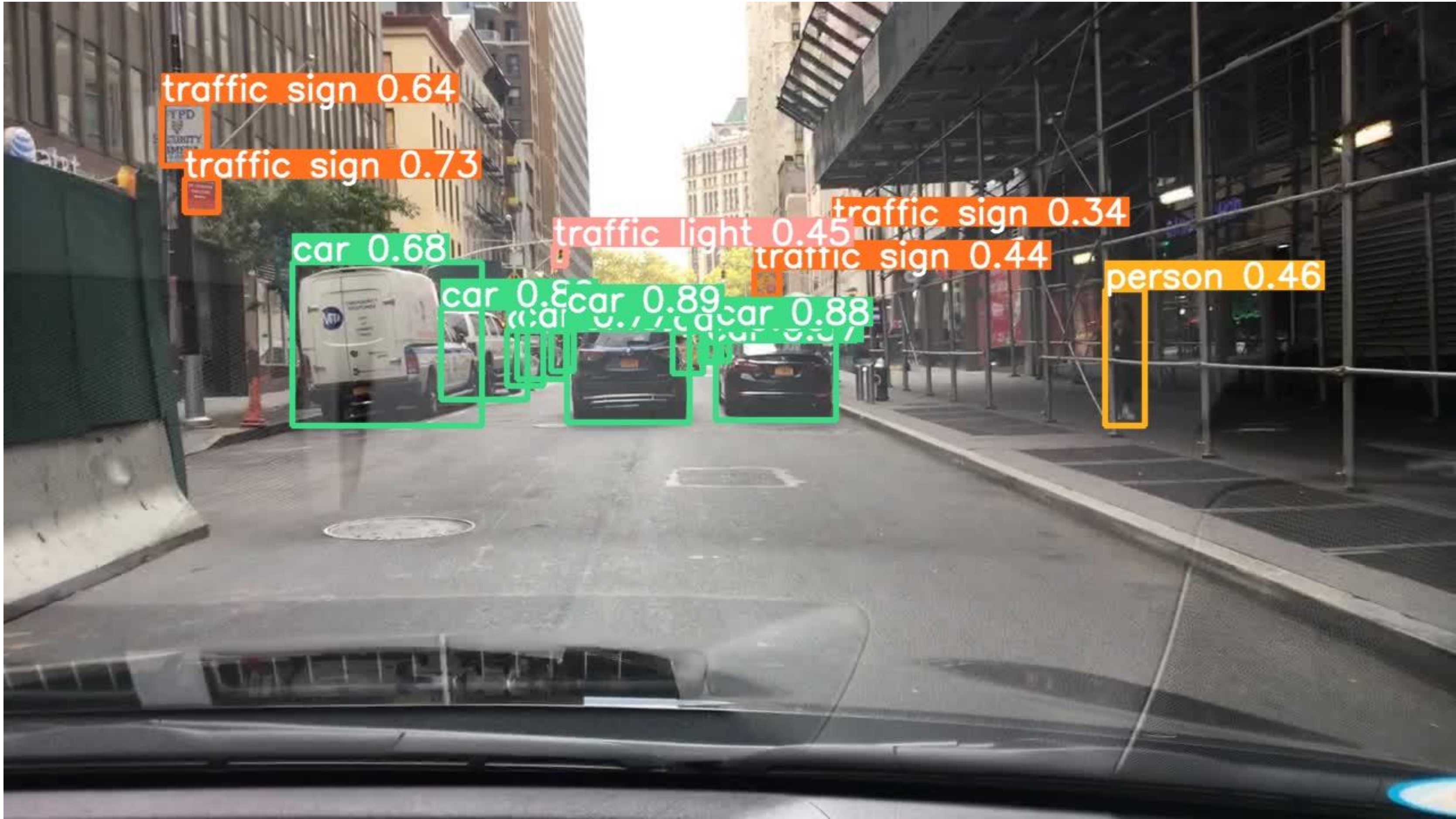
Good detection on big distant objects



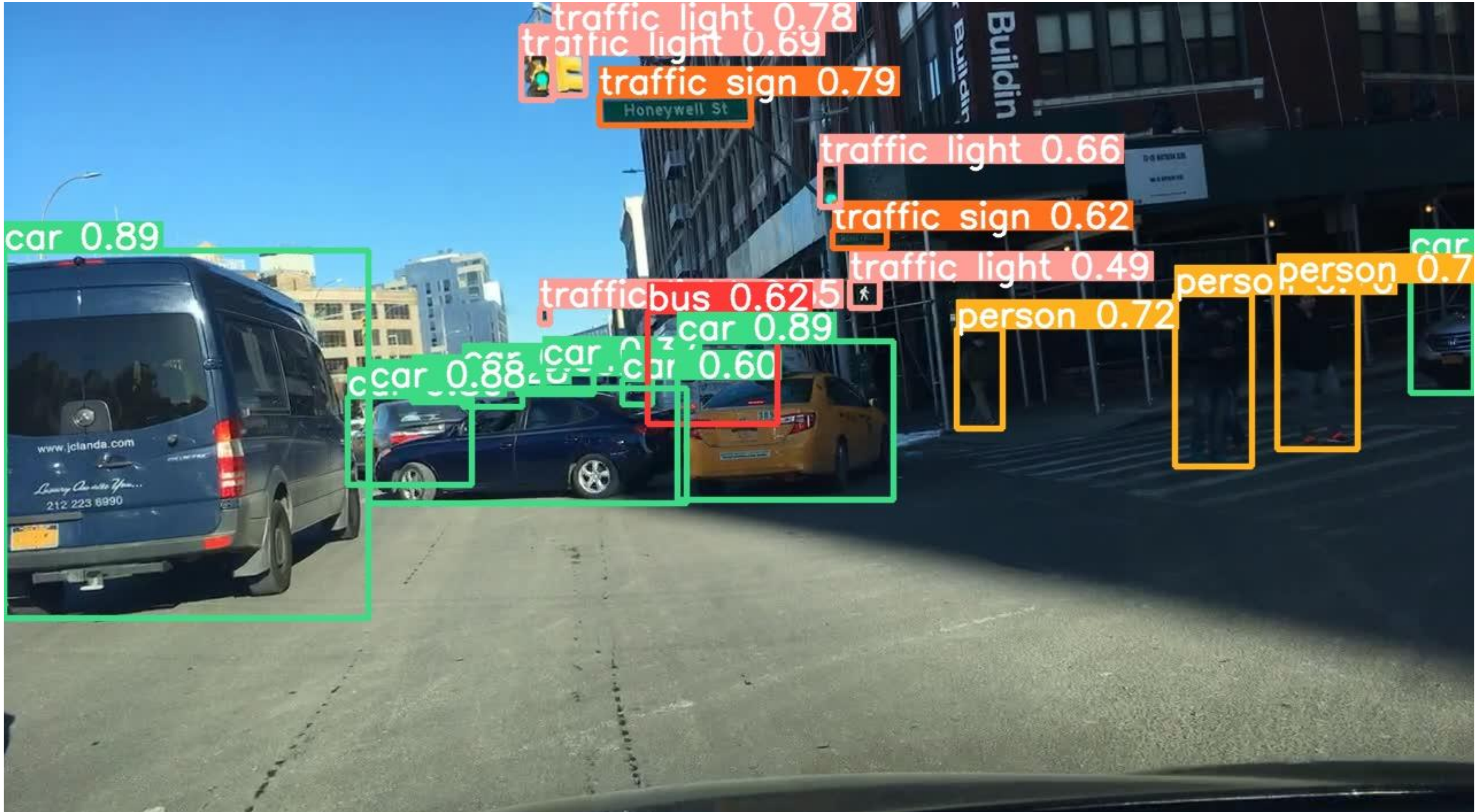
Wrong identification of person



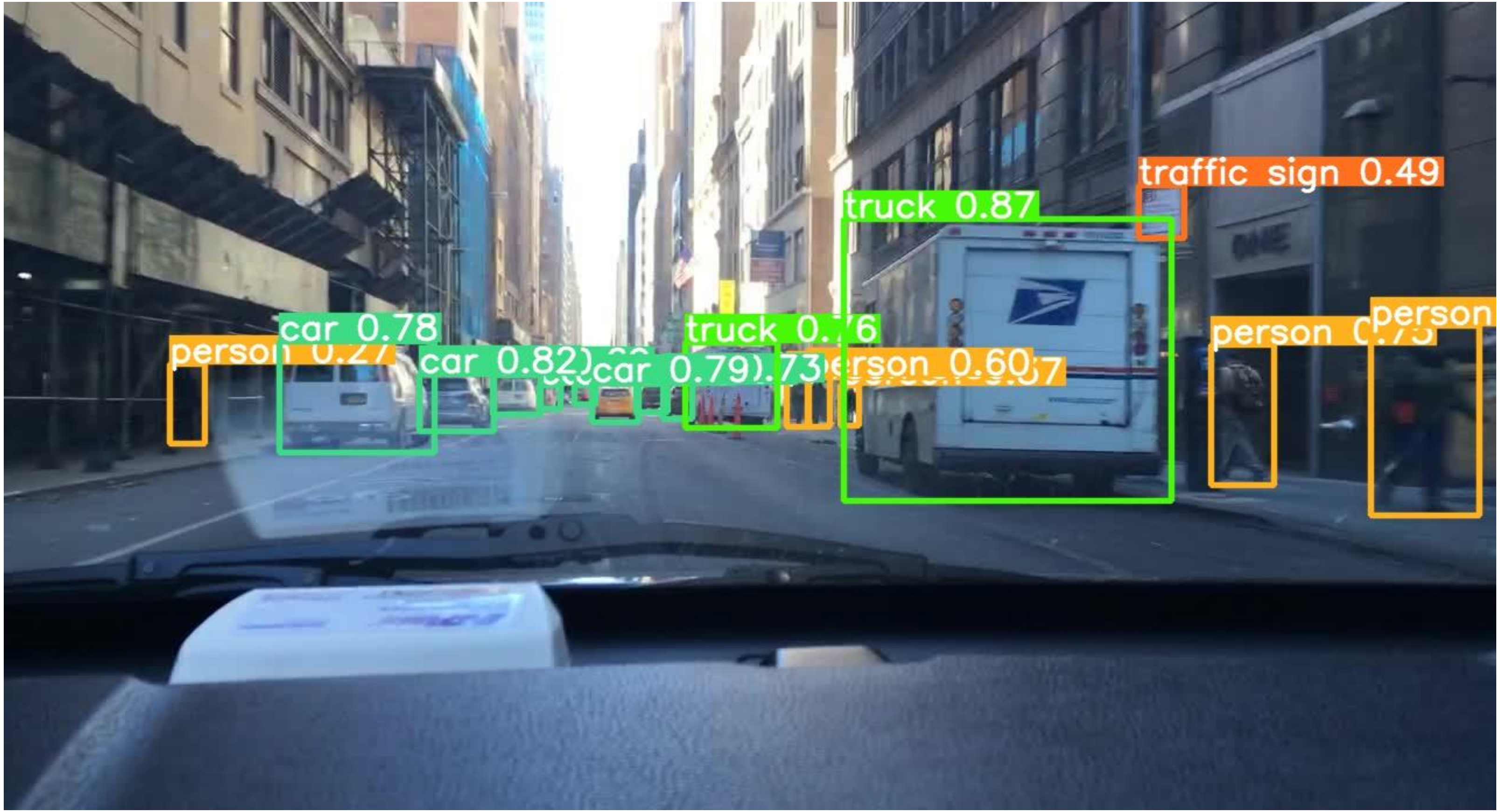
Truck mis-detected as car, as car has high imbalance



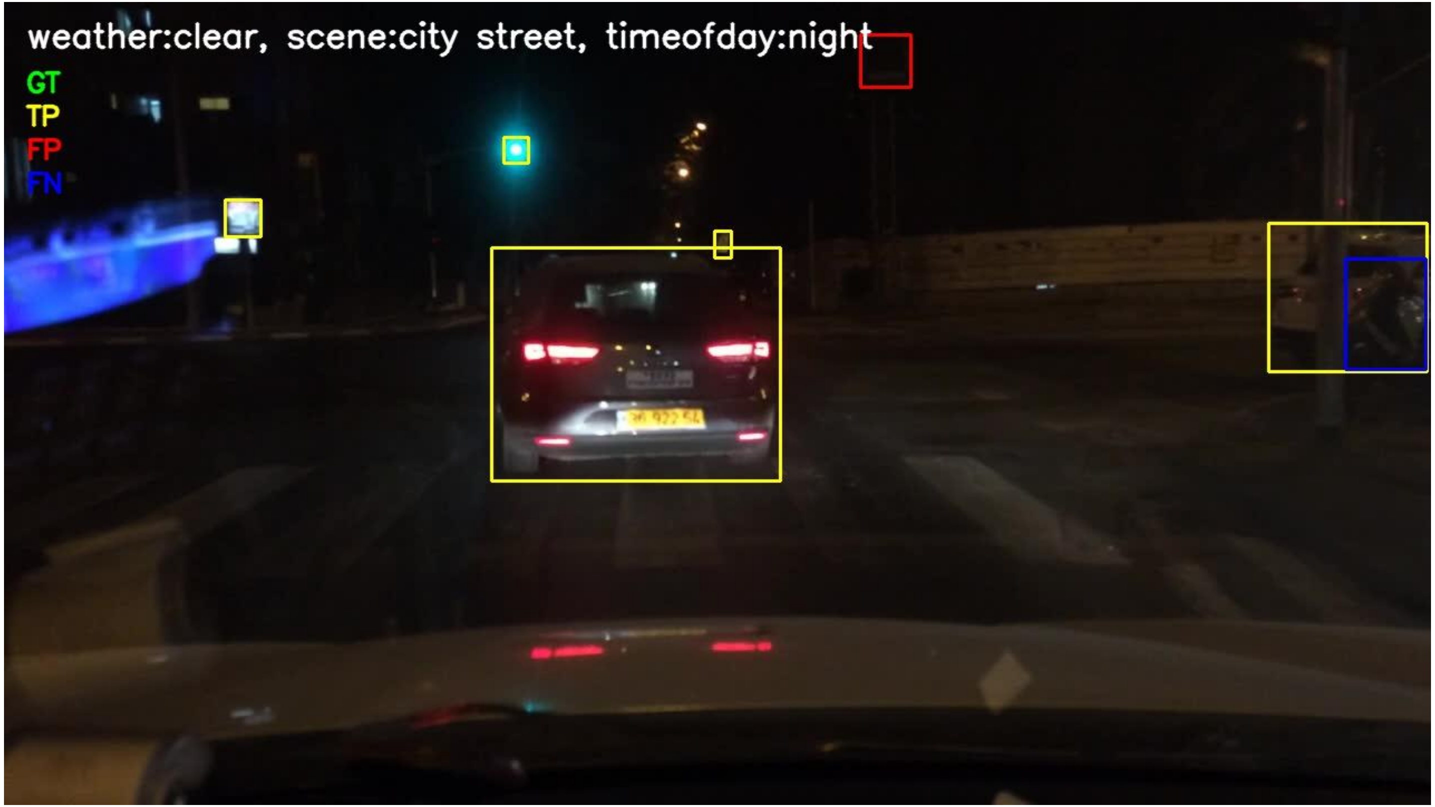
Some traffic lights and signs are mis classified



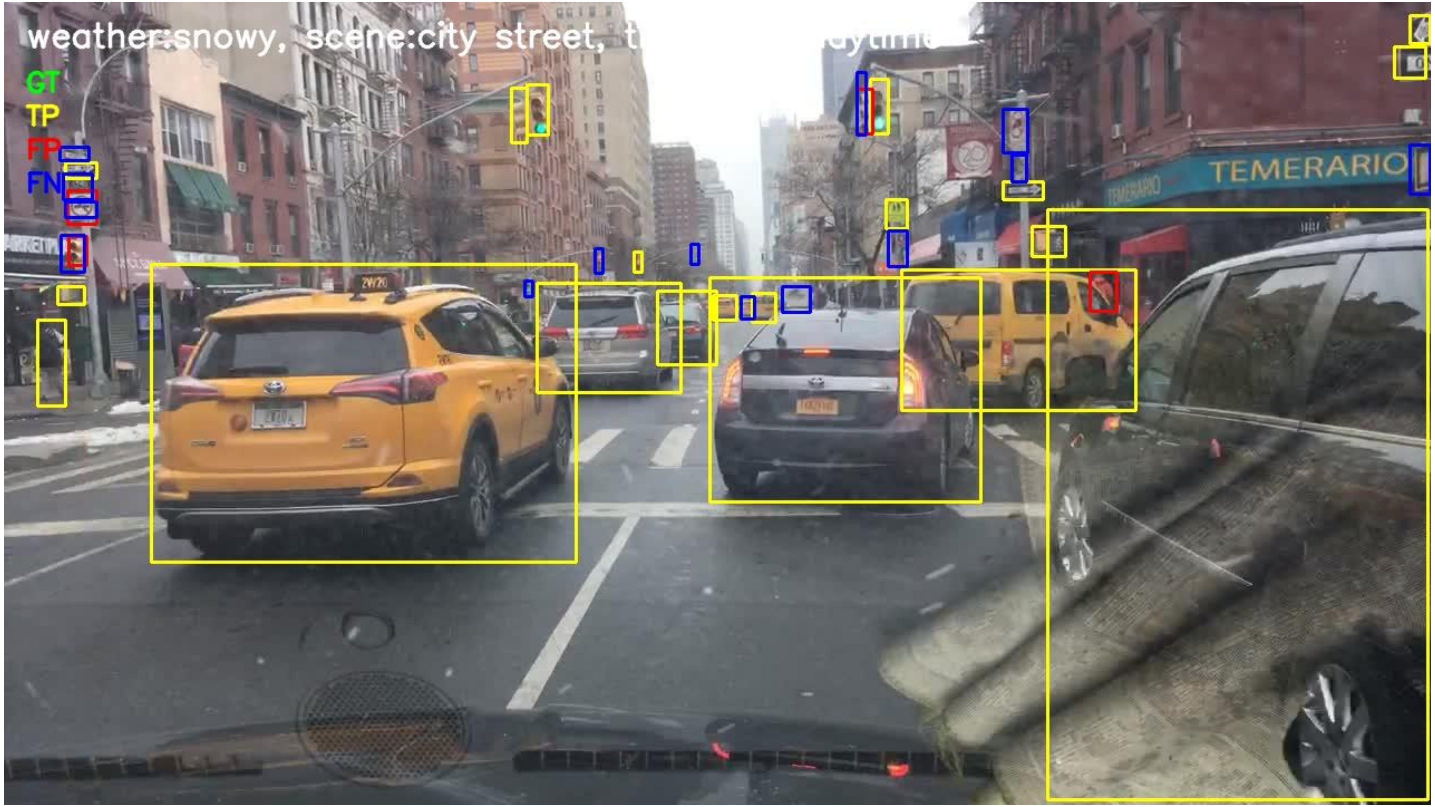
Blurry detections are also correct, some advertisements are mixed with traffic signs.



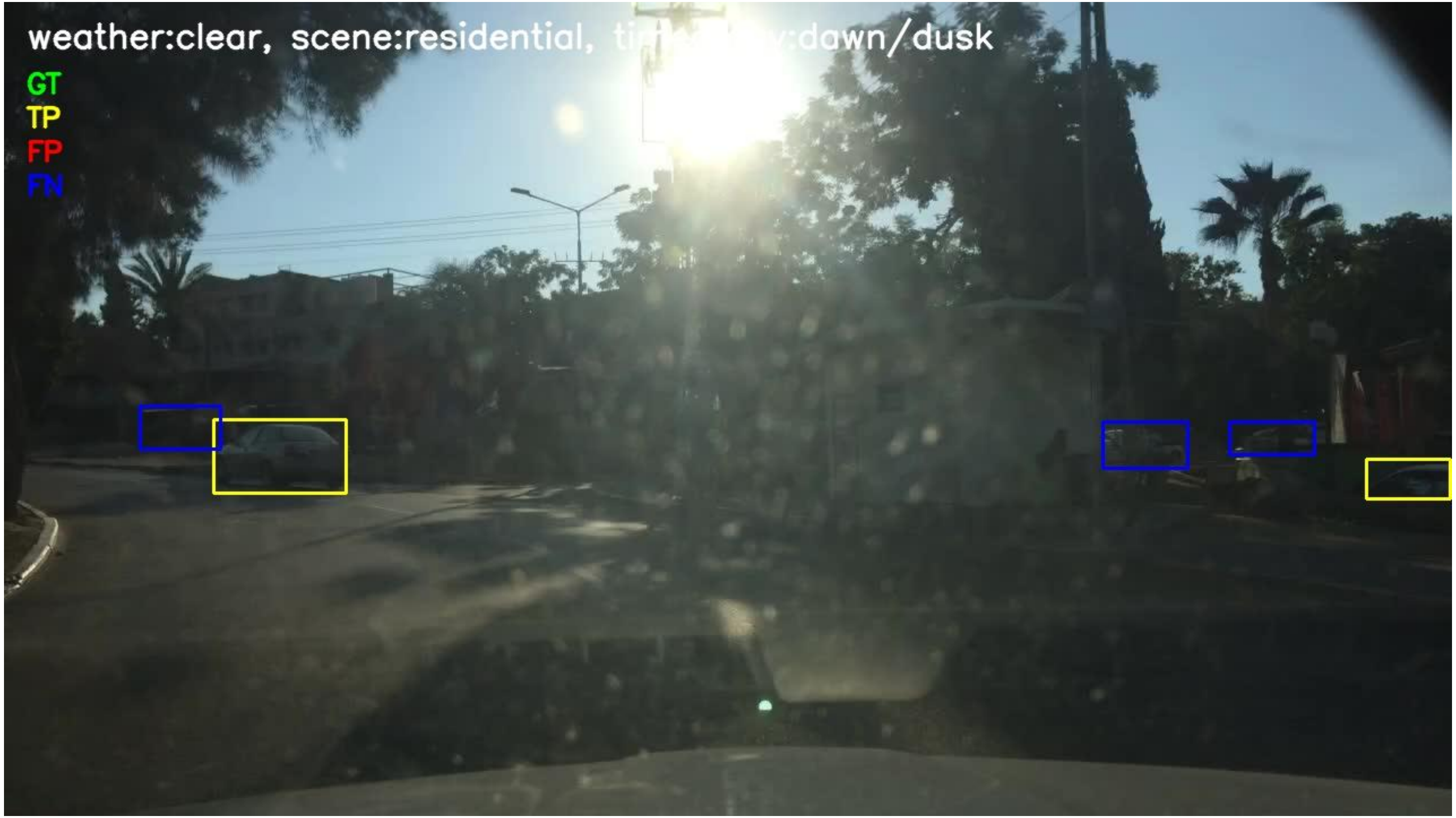
Bike is overlapping with car here, and car being dominant class overpowers the bike detection.



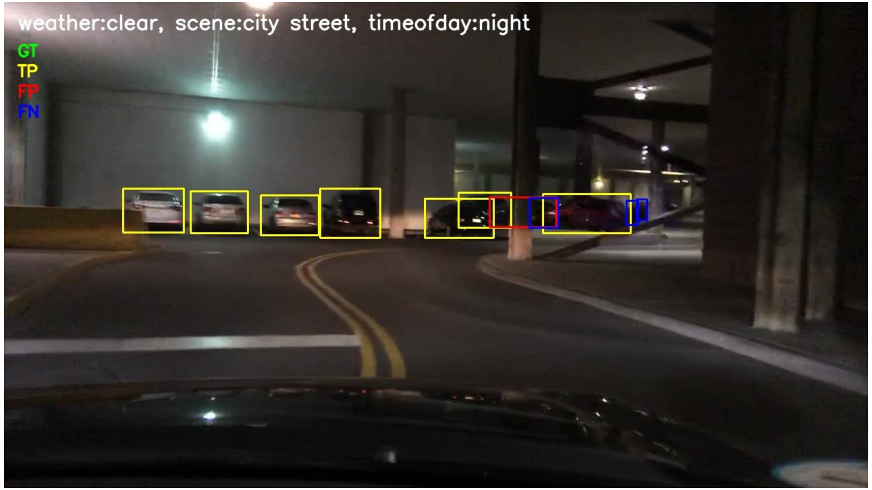
Partially visible person is marked as FP, some very small distant objects are missing.

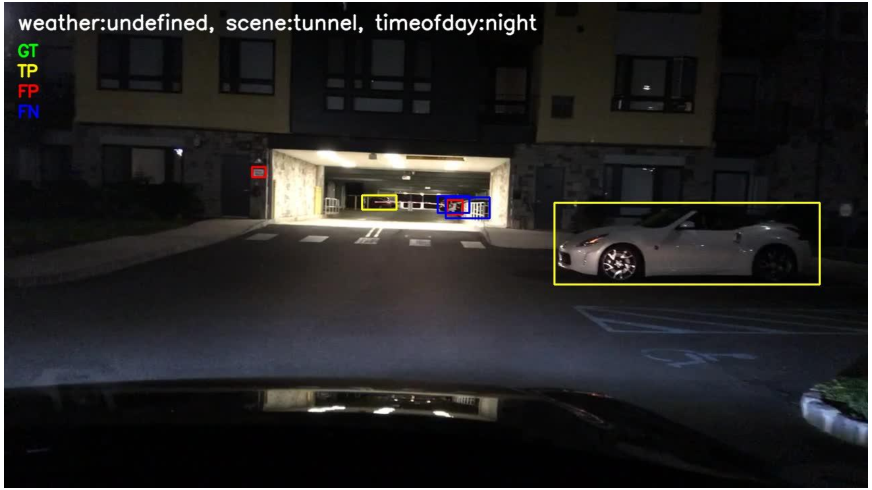


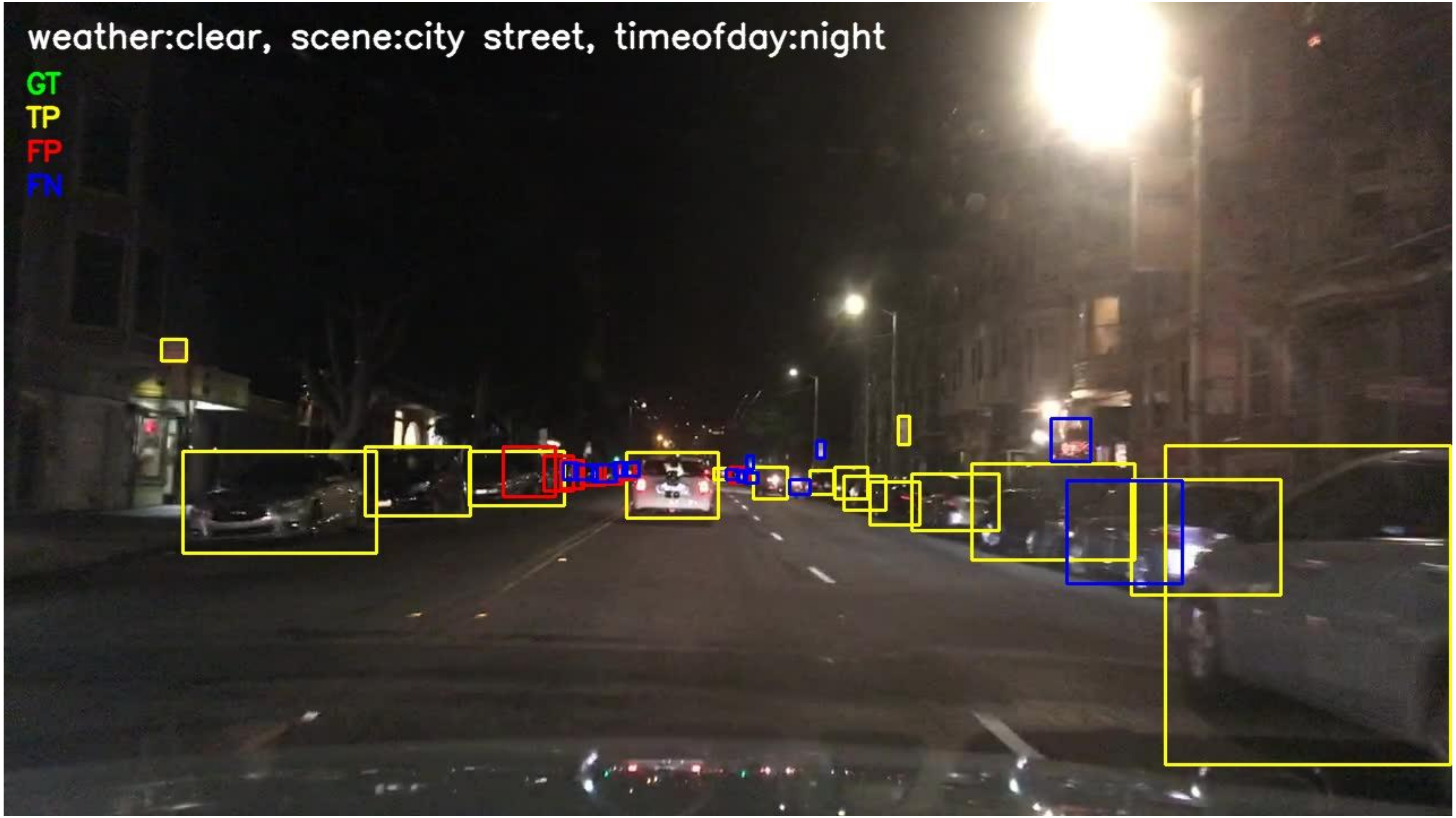
Some cars in the background are missing, but they are also not really very visible in the view. The picture is also not very clear.



Occluded and overlapping objects are missed repeatedly.






weather:clear, scene:city street, timeofday:night
GT
TP
FP
FN

Results

IoU 0.5:

```
 Overall: {'Precision': 0.7969242060261748, 'Recall': 0.6438073369735573, 'F1': 0.7122288634963853, 'TP': 119443, 'FP': 30437, 'FN': 66083}

  Bus: {'Precision': 0.7290268450259841, 'Recall': 0.5441452720449935, 'F1': 0.6231619339047861, 'TP': 869, 'FP': 323, 'FN': 728}

  TrafficLight: {'Precision': 0.7468769518661837, 'Recall': 0.5337176864224021, 'F1': 0.6225562962637625, 'TP': 14349, 'FP': 4863, 'FN': 12536}

  TrafficSign: {'Precision': 0.7447358833972774, 'Recall': 0.5856250716000451, 'F1': 0.6556651795390995, 'TP': 20443, 'FP': 7007, 'FN': 14465}

  Person: {'Precision': 0.7792207791357393, 'Recall': 0.5383803347505368, 'F1': 0.636788814307993, 'TP': 7140, 'FP': 2023, 'FN': 6122}

  Bike:  {'Precision': 0.6734059087505352, 'Recall': 0.4299900690864051, 'F1': 0.524848008546253, 'TP': 433, 'FP': 210, 'FN': 574}

  Truck: {'Precision': 0.7110596407900812, 'Recall': 0.5785630151758391, 'F1': 0.6380044408183995, 'TP': 2456, 'FP': 998, 'FN': 1789}

  Motor: {'Precision': 0.6953124972839355, 'Recall': 0.3938053088632626, 'F1': 0.5028243956562058, 'TP': 178, 'FP': 78, 'FN': 274}

  Car: {'Precision': 0.8318355119731319, 'Recall': 0.7151581370776817, 'F1': 0.7690963061349597, 'TP': 73308, 'FP': 14820, 'FN': 29198}

  Train: {'Precision': 0.0, 'Recall': 0.0, 'F1': 0.0, 'TP': 0, 'FP': 10, 'FN': 15}

  Rider: {'Precision': 0.7177419335544571, 'Recall': 0.4114021565309674, 'F1': 0.5230161861212552, 'TP': 267, 'FP': 105, 'FN': 382}
```

IoU 0.75:

```
 Overall: {'Precision': 0.49959967974046166, 'Recall': 0.4036091976305013, 'F1': 0.4465031460095184, 'TP': 74880, 'FP': 75000, 'FN': 110646}

  Bus {'Precision': 0.6577181202535921, 'Recall': 0.4909204755848964, 'F1': 0.5622081870858632, 'TP': 784, 'FP': 408, 'FN': 813}

  TrafficLight: {'Precision': 0.2666042056908909, 'Recall': 0.1905151571437413, 'F1': 0.22222655681937742, 'TP': 5122, 'FP': 14090, 'FN': 21763}

  TrafficSign: {'Precision': 0.46510018213241894, 'Recall': 0.3657327833056682, 'F1': 0.40947383280742544, 'TP': 12767, 'FP': 14683, 'FN': 22141}

  Person: {'Precision': 0.43566517511342734, 'Recall': 0.3010104056476391, 'F1': 0.356030731836109, 'TP': 3992, 'FP': 5171, 'FN': 9270}

  Bike: {'Precision': 0.37013996832015555, 'Recall': 0.23634558069876307, 'F1': 0.28848437246947106, 'TP': 238, 'FP': 405, 'FN': 769}

  Truck: {'Precision': 0.6181239141232994, 'Recall': 0.5029446406353487, 'F1': 0.5546169879241558, 'TP': 2135, 'FP': 1319, 'FN': 2110}

  Motor: {'Precision': 0.35937499859619143, 'Recall': 0.20353982255854022, 'F1': 0.2598865432355226, 'TP': 92, 'FP': 164, 'FN': 360}

  Car: {'Precision': 0.5627382897539631, 'Recall': 0.4838058259957095, 'F1': 0.5202949380711669, 'TP': 49593, 'FP': 38535, 'FN': 52913}

  Train: {'Precision': 0.0, 'Recall': 0.0, 'F1': 0.0, 'TP': 0, 'FP': 10, 'FN': 15}

  Rider: {'Precision': 0.42204300961816393, 'Recall': 0.24191063136839655, 'F1': 0.30754116205783066, 'TP': 157, 'FP': 215, 'FN': 492}
```

Common Set of Observation

From all the examples illustrated it can be concluded that :

1. Occluded and overlapping objects are missed
2. Very distant small objects are missed (specially in night scenes)
3. Very dark objects are missed
4. Some traffic signs and signals are classified as each other.
5. There is no ‘train’ class in val set.
6. Night time accuracy is lower than daytime.
7. Overall accuracy is 79.6% where on an average each class has around 70% accuracy except car (having 83%). This can also be mapped to the ratio of training images present per classes.