JRL780 Computer Vision
Assignment-III

Parthasaradhi Reddy N
2024JRB2028

May 8, 2025

1 Introduction

Models are uploaded inGoogle Drive
I attempted to use the provided checkpoint, but encountered an error during loading. Therefore, I used the Hugging Face Transformers module to access pre-trained models.

For Part 1 Subtask 1, the DETR model was loaded, and inference was performed on the validation dataset. In Subtask 2, I froze the model parameters and fine-tuned specific components like the encoder, decoder, and full model. Similarly, for Grounding DINO, I ran zero-shot inference using different prompts.For subtask2 I have created and trained a small model to get learnable prompt embeddings.

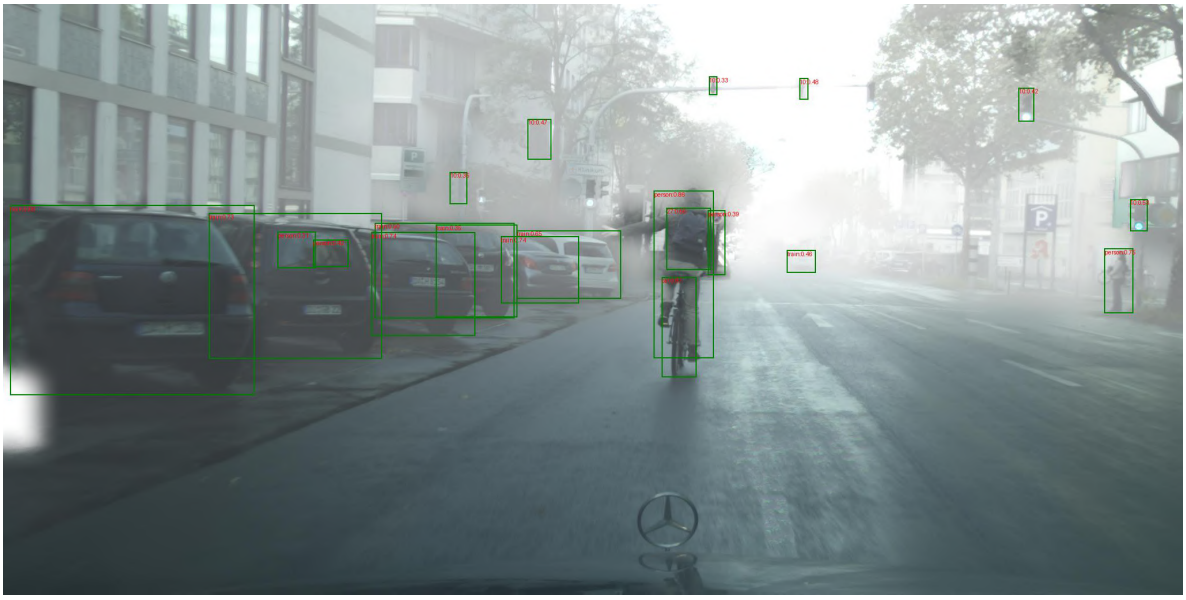
2 Part 1

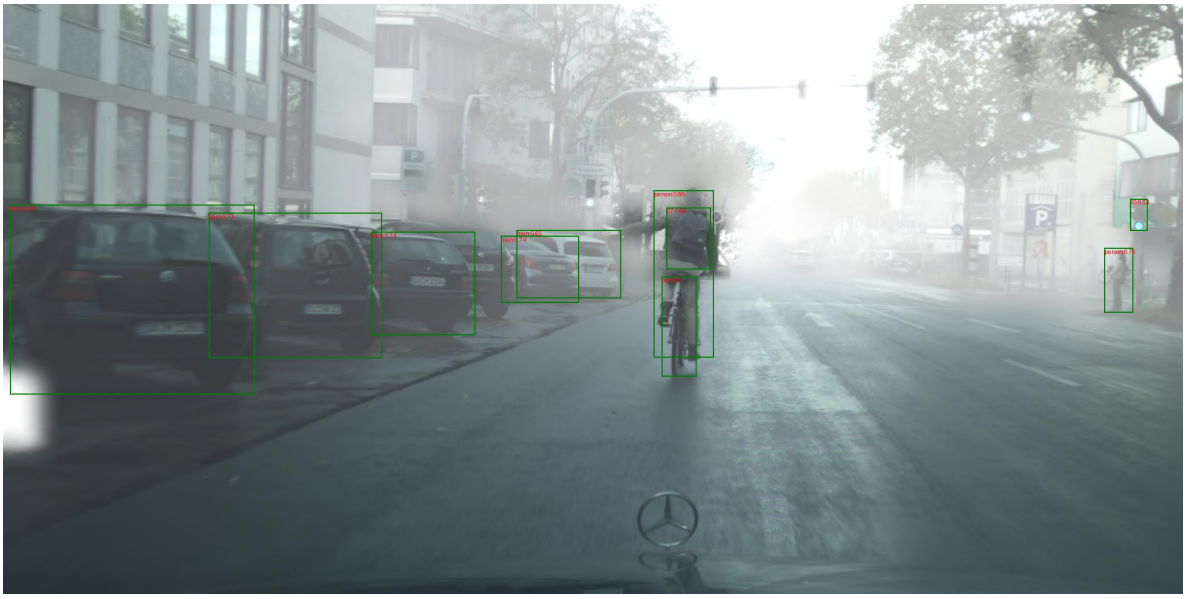
2.1 Subtask 1

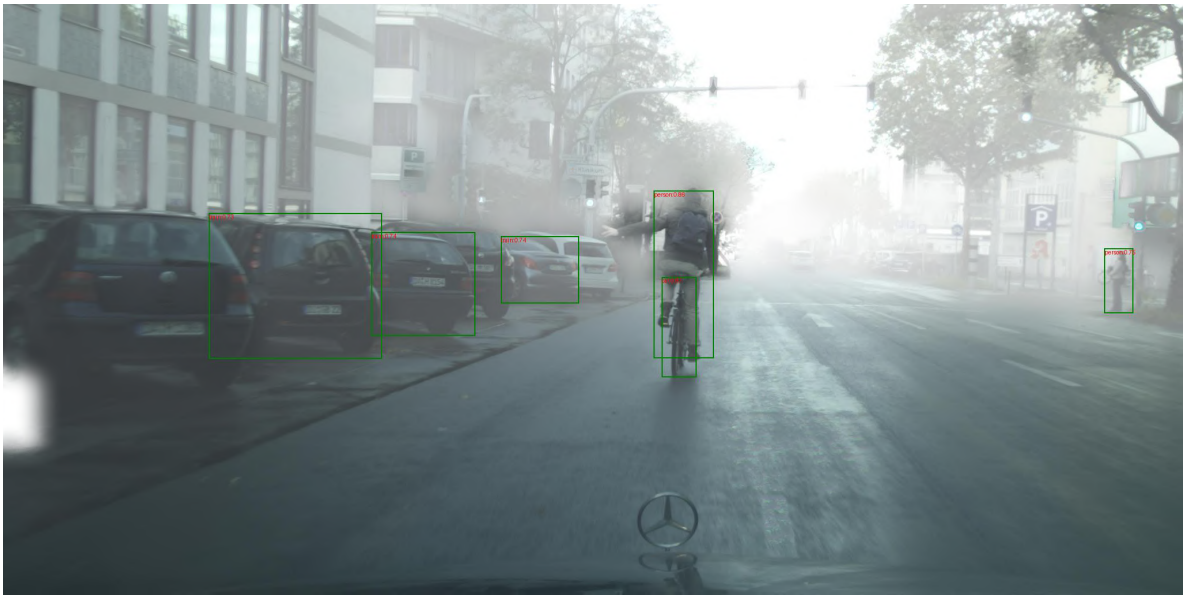
The DETR model was loaded using Hugging Face Transformers and inference was performed on the validation dataset. Below are the qualitative and quantitative results obtained.

Threshold	mAP Score	Precision	Recall
0.3	0.14549	0.23177	0.28749
0.5	0.12607	0.19388	0.22606
0.7	0.08533	0.12369	0.13544

Table 1: Quantitative results from DETR pretrained model.







Observations

The pretrained model was able to detect object classes, but many predictions were inconsistent or inaccurate. This is also reflected in the low mAP score of 13.

Subtask 2: Finetuning Approaches

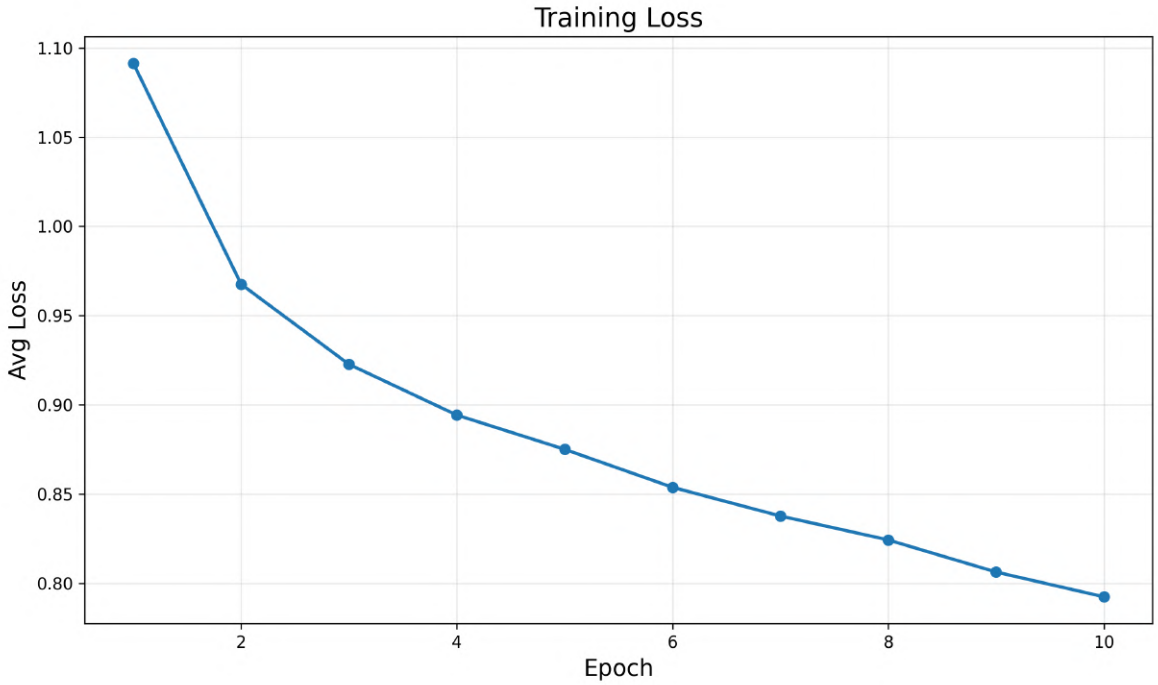
After evaluating the pretrained model, I applied three fine-tuning strategies:

1. Encoder only
2. Decoder only
3. Full model fine-tuning

1) Encoder Fine-tuning

Threshold	mAP Score	Precision	Recall
0.3	0.22022	0.33801	0.38961
0.5	0.15232	0.21400	0.22738
0.7	0.07390	0.09250	0.09185

Table 2: Quantitative results from DETR encoder finetuning.



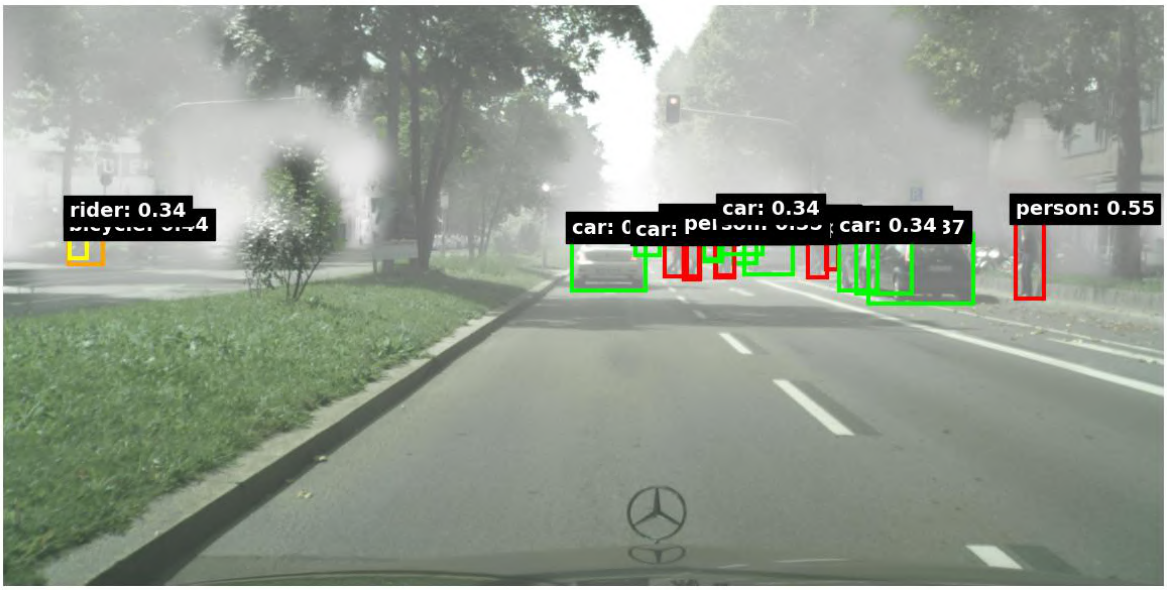


Image ID: 747 | Threshold: 0.50

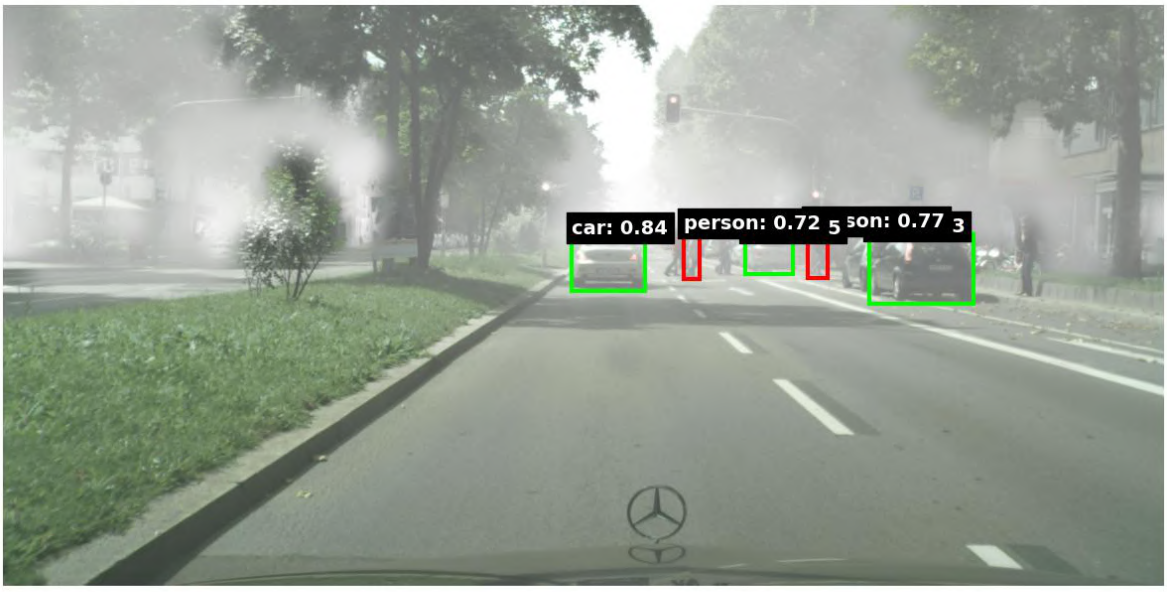


Image ID: 747 | Threshold: 0.70

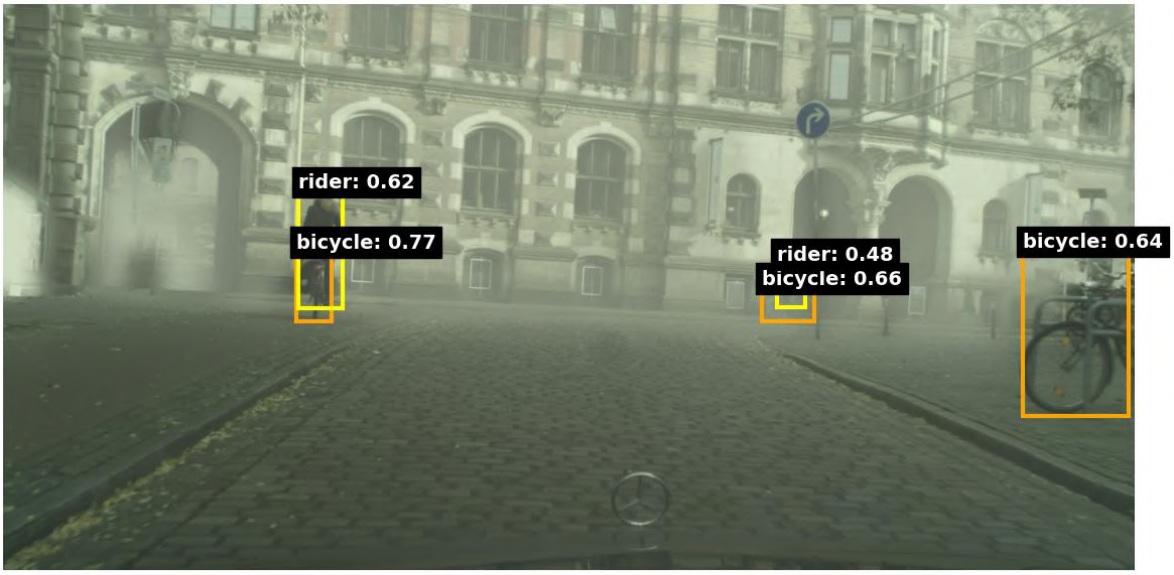


Image ID: 1576 | Threshold: 0.30

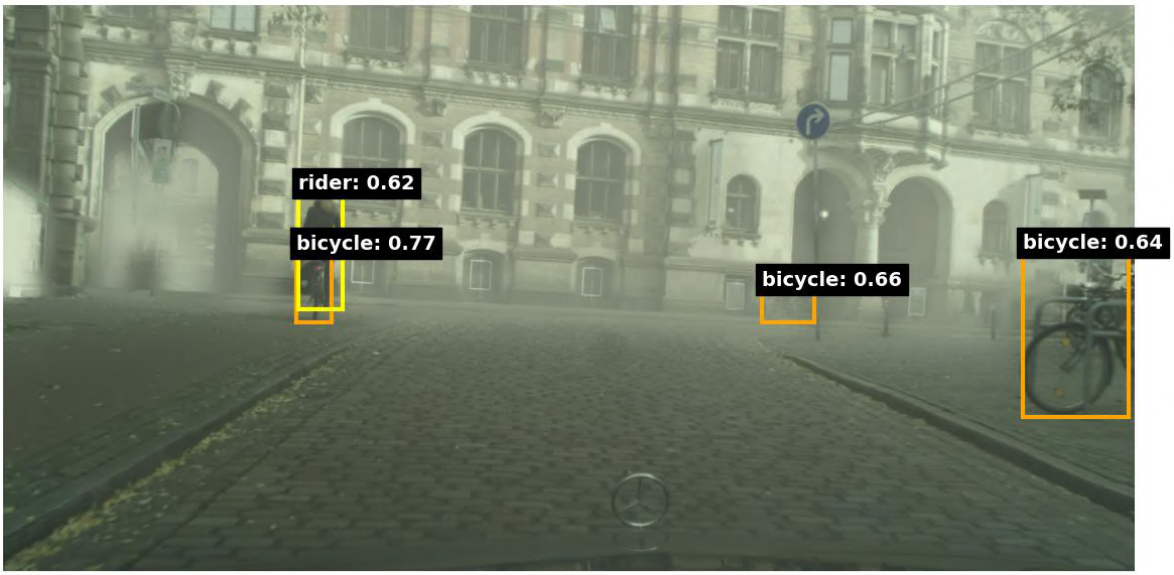
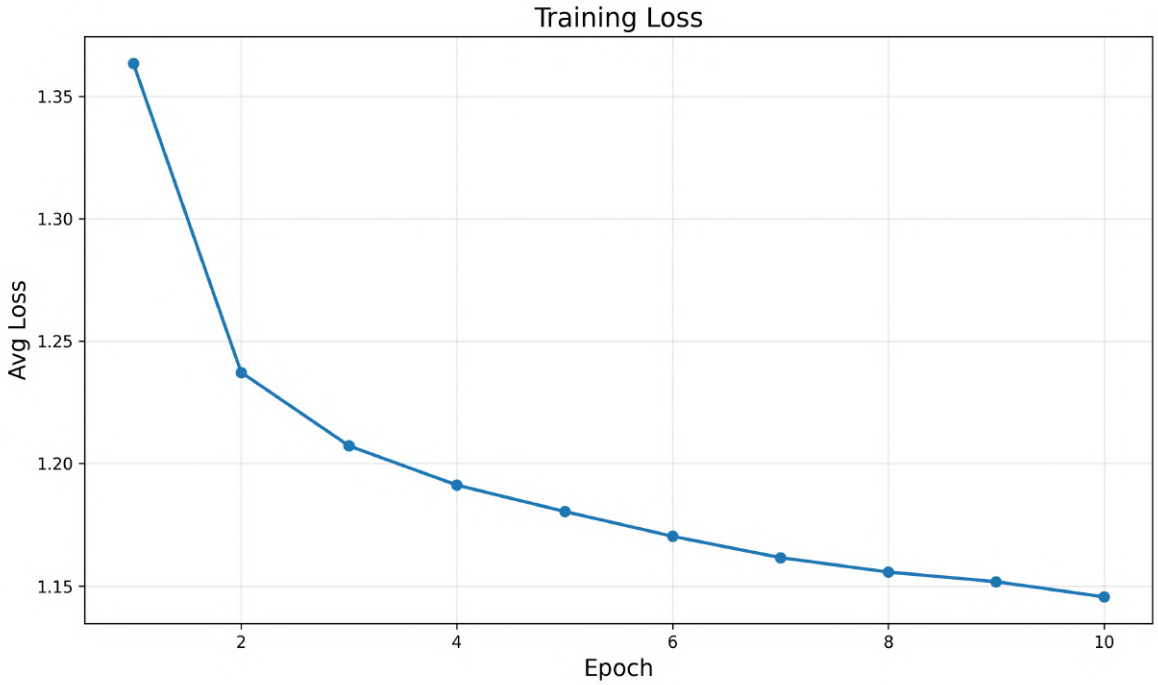


Image ID: 1576 | Threshold: 0.50

2) Decoder Fine-tuning

Threshold	mAP Score	Precision	Recall
0.3	0.14391	0.23661	0.28096
0.5	0.08561	0.12313	0.12386
0.7	0.04694	0.06027	0.05826

Table 3: Quantitative results from DETR decoder finetuning.



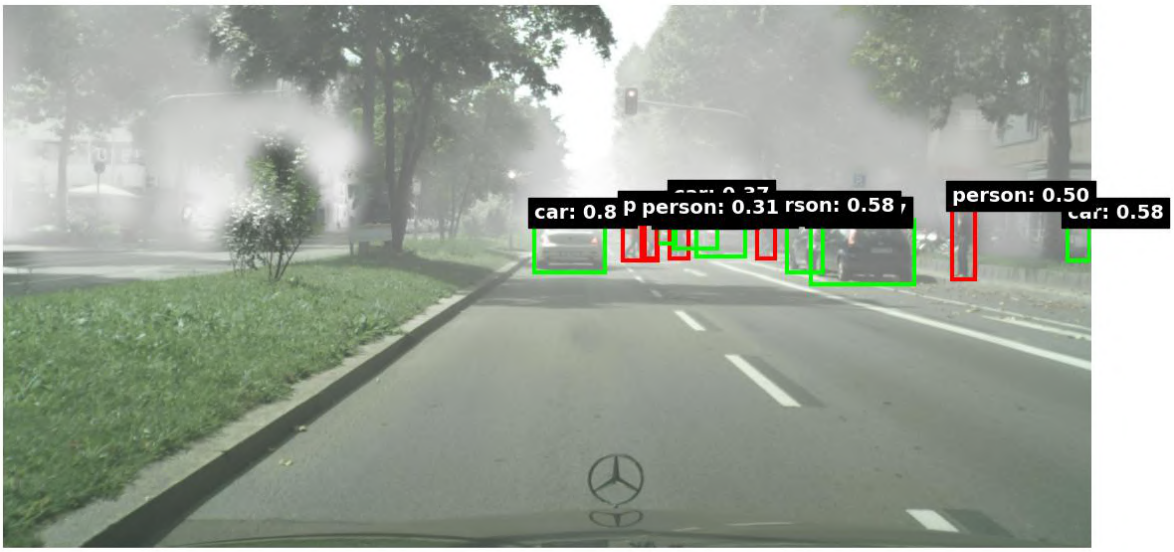


Image ID: 747 | Threshold: 0.30

Image ID: 747 | Threshold: 0.50

Image ID: 747 | Threshold: 0.70

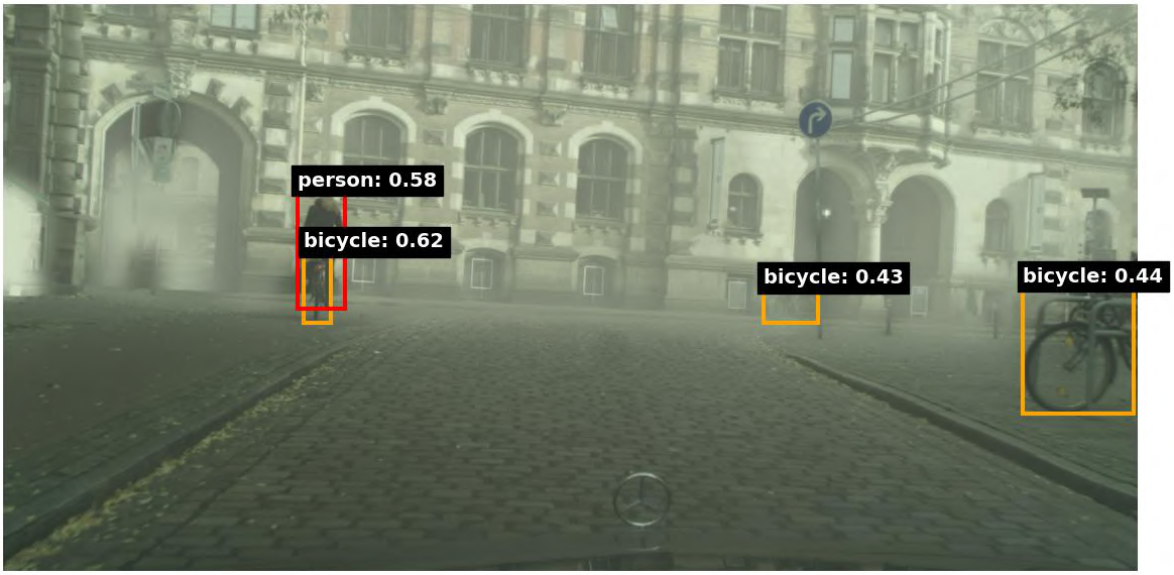


Image ID: 1576 | Threshold: 0.30

Image ID: 1576 | Threshold: 0.50

Image ID: 1576 | Threshold: 0.70

3) Full Model Fine-tuning

Threshold	mAP Score	Precision	Recall
0.3	0.22248	0.33710	0.40110
0.5	0.14547	0.20204	0.21984
0.7	0.05570	0.06927	0.06886

Table 4: Quantitative results from DETR full finetuning.

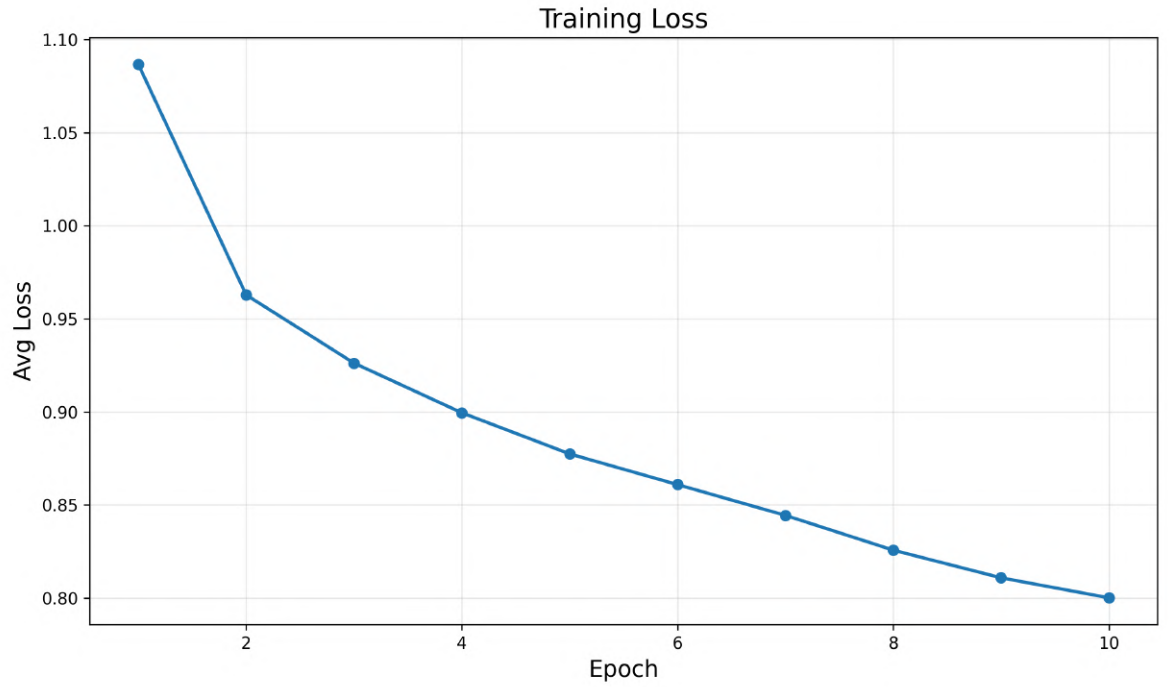


Image ID: 747 | Threshold: 0.30

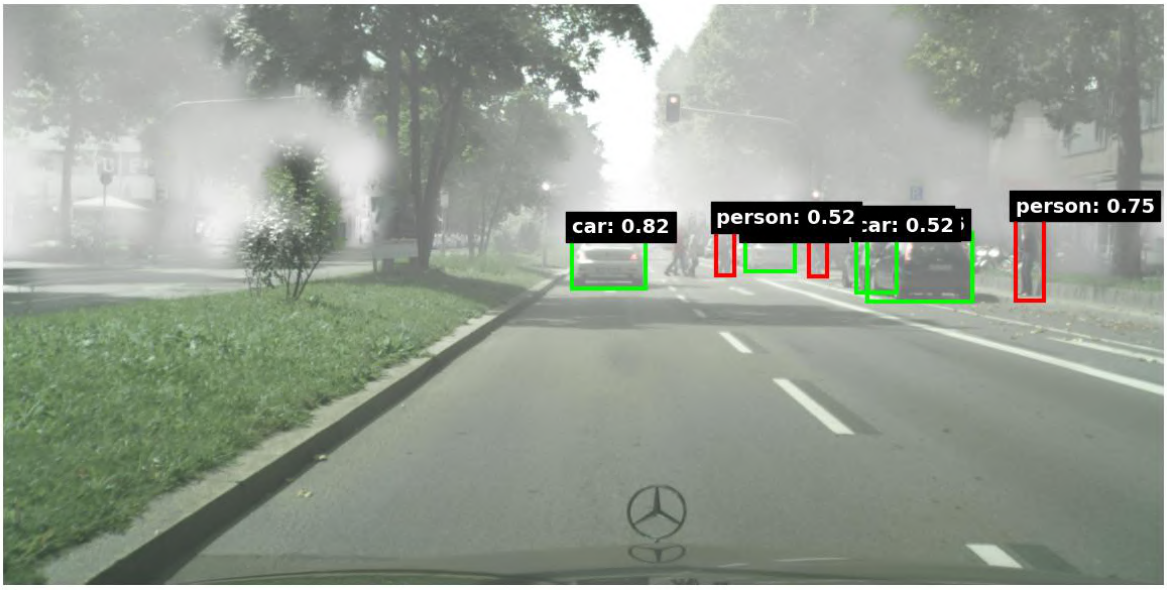


Image ID: 747 | Threshold: 0.50

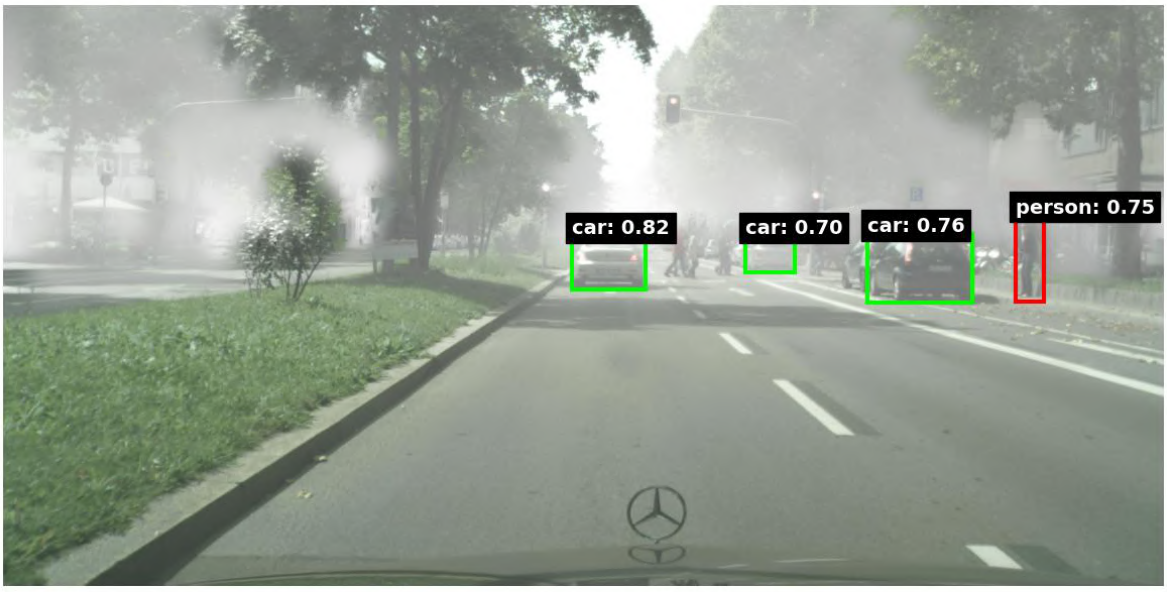


Image ID: 747 | Threshold: 0.70

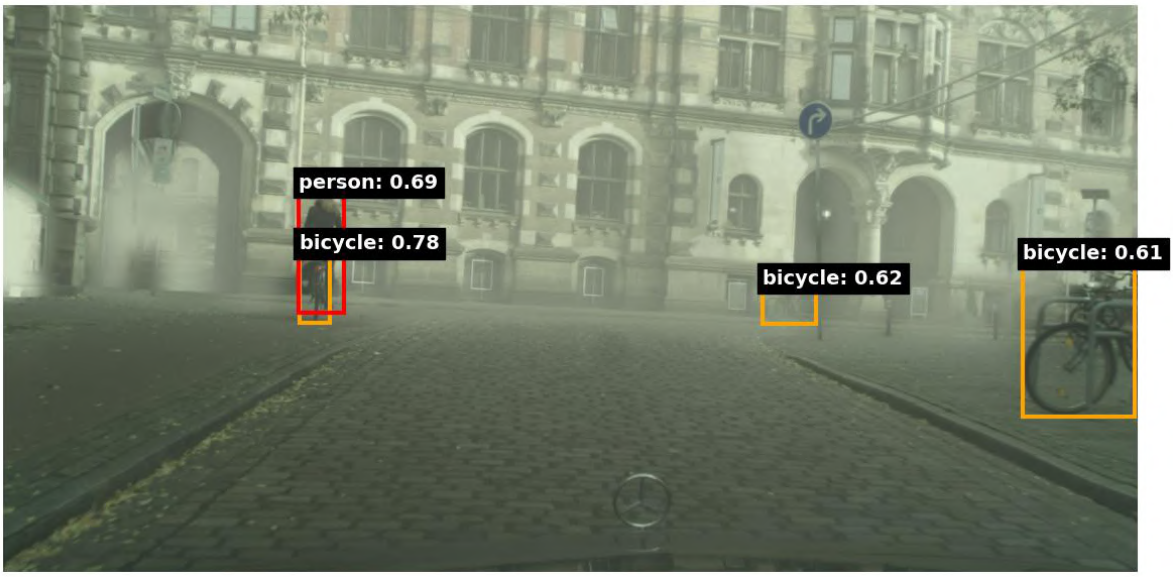


Image ID: 1576 | Threshold: 0.30

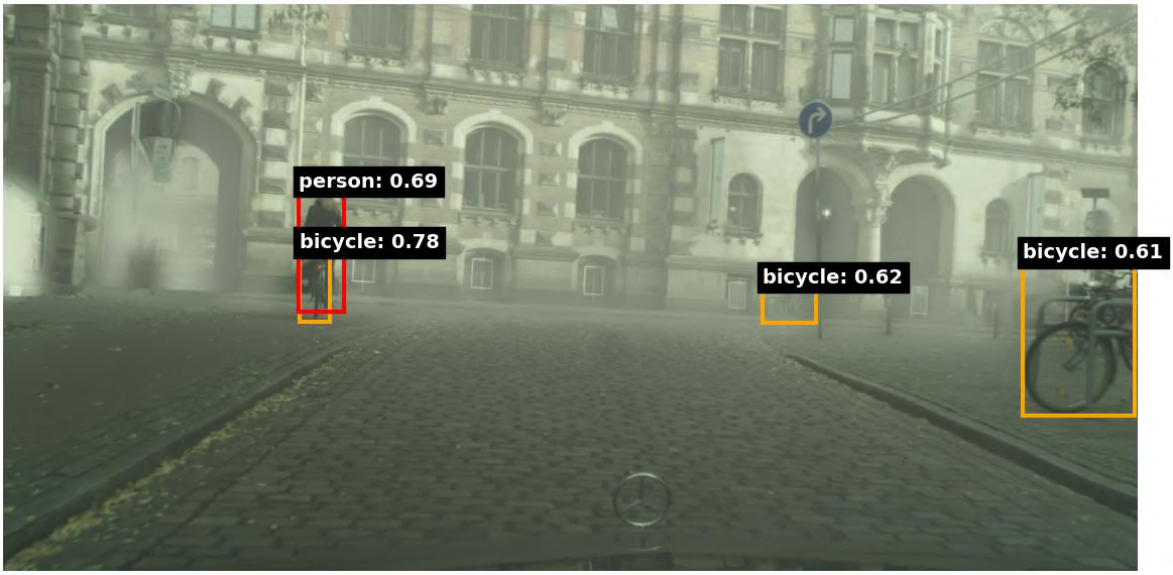


Image ID: 1576 | Threshold: 0.50

3 Conclusion

Fine-tuning significantly improved the model's performance across all evaluation metrics, especially when the full model was fine-tuned.

Part2:Grounding Dino

3.1 Zero Shot Inference

Used Grounding dino model from Hugging face and for zero shot inference and these are the results obtained I used the simple prompt "a person" for zero shot infernece and these are the results

Metric	Value
AP@[IoU=0.50:0.95 — area=all — maxDets=100]	0.014
AP@[IoU=0.50 — area=all — maxDets=100]	0.023
AP@[IoU=0.75 — area=all — maxDets=100]	0.016
AP@[IoU=0.50:0.95 — area=small — maxDets=100]	0.002
AP@[IoU=0.50:0.95 — area=medium — maxDets=100]	0.027
AP@[IoU=0.50:0.95 — area=large — maxDets=100]	0.043
AR@[IoU=0.50:0.95 — area=all — maxDets=1]	0.007
AR@[IoU=0.50:0.95 — area=all — maxDets=10]	0.028
AR@[IoU=0.50:0.95 — area=all — maxDets=100]	0.029
AR@[IoU=0.50:0.95 — area=small — maxDets=100]	0.002
AR@[IoU=0.50:0.95 — area=medium — maxDets=100]	0.049
AR@[IoU=0.50:0.95 — area=large — maxDets=100]	0.096

Table 5: COCO Evaluation Metrics (Formatted)

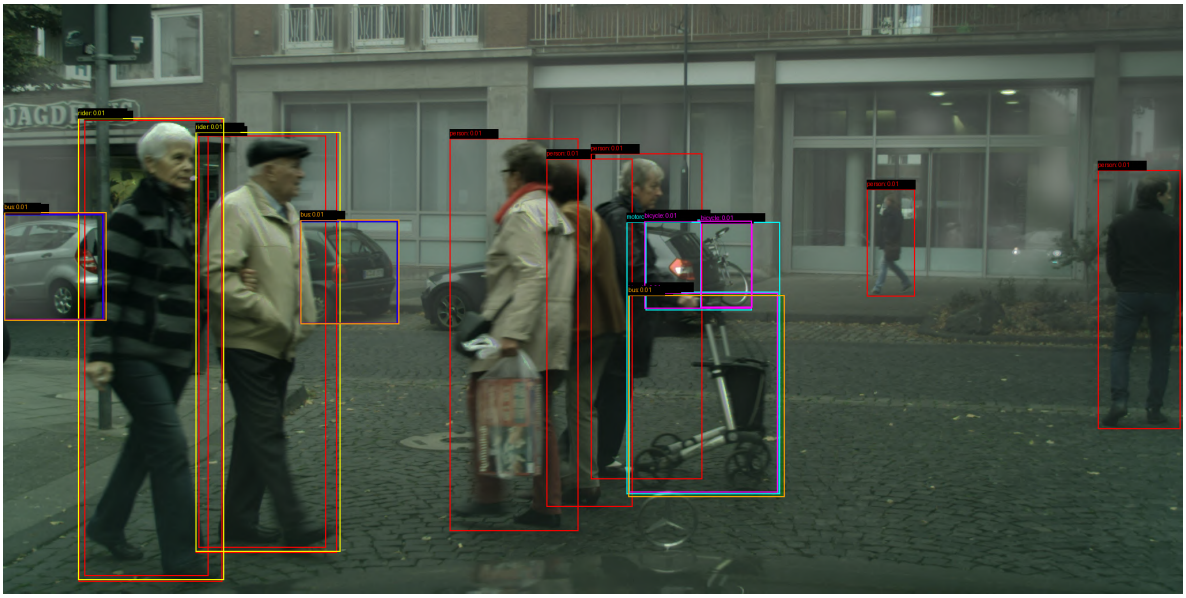
Multiple prompts

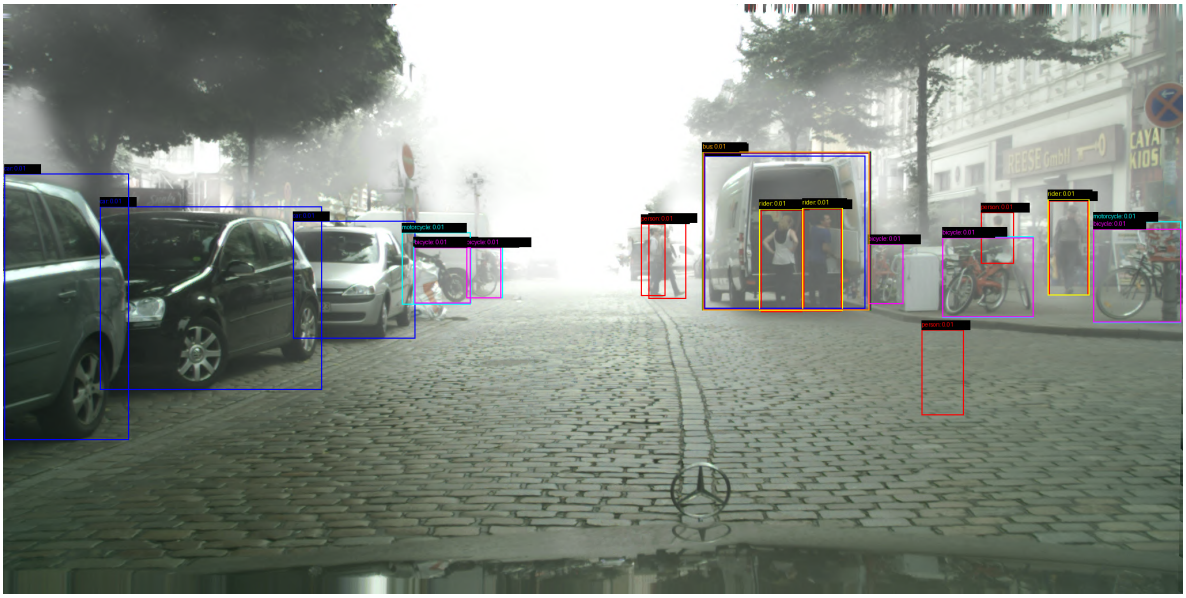
Next I have used multiple prompts i.e all the categories in the dataset and obtained the results
prompts used
"person": "a person",
"car": "a car",
"train": "a train",
"rider": "a rider",
"truck": "a truck",
"motorcycle": "a motorcycle",
"bicycle": "a bicycle",
"bus": "a bus"

Don't know the reason for 0 values in category wise metrics

Metric	Overall Value	Category Averaged
mAP	0.0863	0.0000
mAP@0.50	0.1354	0.0000
mAP@0.75	0.0917	0.0000

Table 6: Detection Performance Metrics





3.2 Prompt tuning

Created a small model and created a hook to inject learnable prompt embeddings into the embedding layer and after training these are the results obtained.

The learned prompts mostly focusing on the foggy areas with small objects.I have trained for only 10 epochs.So increasing the model size and training iterations might helps the model to learn the prompts clearly and can work well with the foggy datasets by generalizing the prompts characteristics of dataset.

Part3

YOLO model for object detection

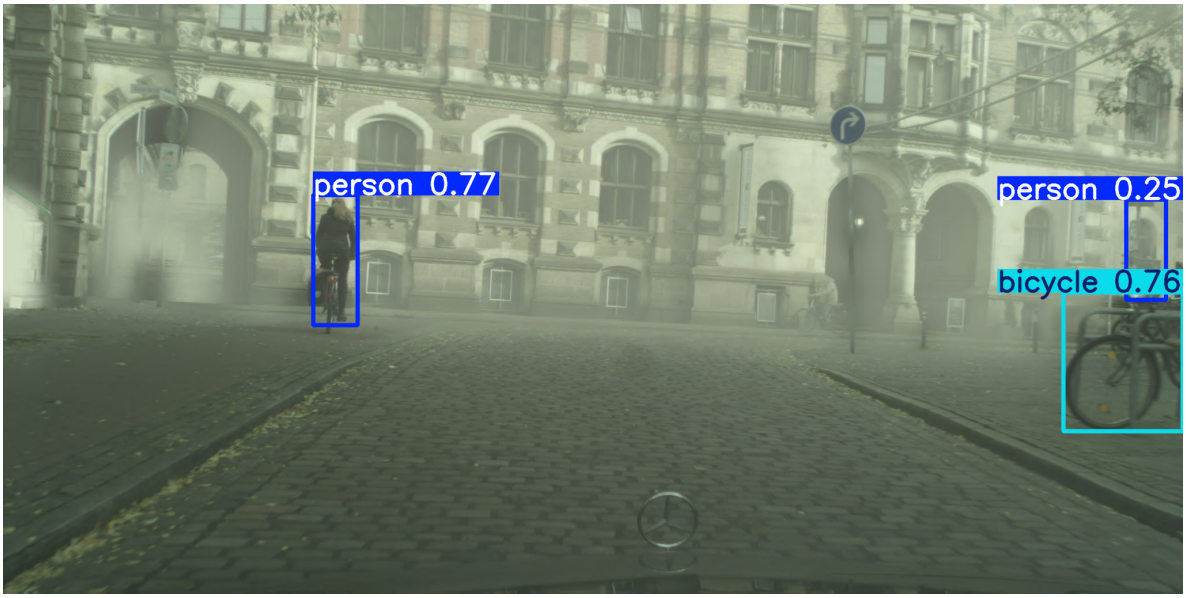
I have used Hugging face yolo v8 model for infernece and these are the results obtained

Category	AP	AP50	AP75
person	0.1269	0.2107	0.1269
car	0.2466	0.3319	0.2756
train	0.0243	0.0330	0.0330
rider	0.0000	0.0000	0.0000
truck	0.0583	0.0741	0.0641
motorcycle	0.0652	0.1333	0.0511
bicycle	0.0402	0.0752	0.0395
bus	0.1141	0.1501	0.1406

Table 7: Per-category average precision metrics (AP, AP50, AP75).

Metric	Value
mAP@[IoU=0.5:0.95]	0.0844
mAP@[IoU=0.5]	0.1260
mAP@[IoU=0.75]	0.0913
AP_small	0.0011
AP_medium	0.0433
AP_large	0.2871
AR@1	0.0735
AR@10	0.1129
AR@100	0.1136
AR_small	0.0002
AR_medium	0.0489
AR_large	0.3805

Table 8: Overall Evaluation Metrics




car 0.52
car car 0.62
car 0car 0.72ır 0.58
person 0.41
person 0.72


car 0.89
car 0.85

Finetuning YoLo

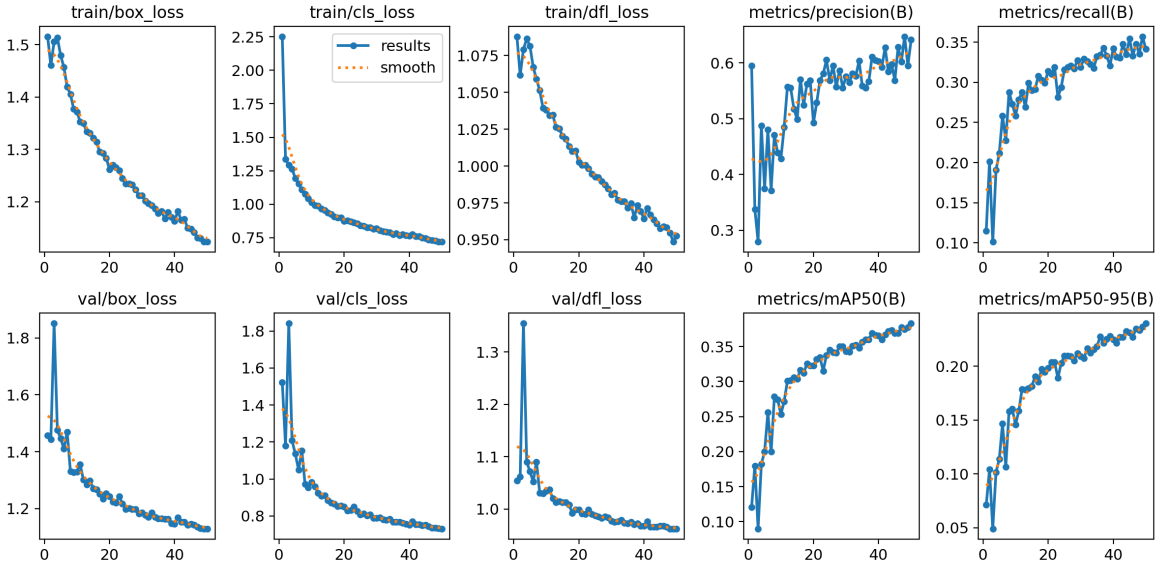
Using yolo v8 from huggig face on the training dataset I have trained the model for 50 epochs initially and these are the results obtained.

Category	AP	AP50	AP75
person	0.1974	0.3343	0.1989
car	0.4001	0.5634	0.4280
train	0.1168	0.1867	0.1353
rider	0.1710	0.3080	0.1668
truck	0.1587	0.1938	0.1685
motorcycle	0.1287	0.2568	0.1428
bicycle	0.0949	0.1827	0.0814
bus	0.2389	0.2979	0.2905

Table 9: Per-category average precision metrics

Metric	Value
mAP@[IoU=0.5:0.95]	0.1883
mAP@[IoU=0.5]	0.2904
mAP@[IoU=0.75]	0.2015
AP_small	0.0115
AP_medium	0.1659
AP_large	0.4728
AR@1	0.1361
AR@10	0.2168
AR@100	0.2225
AR_small	0.0102
AR_medium	0.1915
AR_large	0.5655

Table 10: Overall detection metrics



3.3 Conclusion

Among all the models and the techniques used for objection detection YoLo performance is good with good prediction of bounding boxes and classes with high confidence.The MAP scores and other metrics are also relatively better than other models for object detection. Trainig for more epochs with additional techniques like data augmentation may improve the model's accuaracy and performance.