DINO Object Detection Model Training and Evaluation Report

Chidaksh Ravuru

November 28, 2023

1 Introduction

This report summarizes the process and findings of training the DINO object detection model on a given pedestrian dataset. The tasks were performed in a Jupyter Notebook environment on google colab. I used a T4 GPU on colab with GPU Ram of 15GB for the given task.

2 Setup and Data Preprocessing

2.1 Issues faced

- sample unit tests
- importing MultiScalableDeformableAttention

These issues were resolved and the setup was successful.

3 Model Pipeline for DINO

3.1 Downloading pre-trained checkpoint

Downloaded the pre-trained DINO-4scale R50 backbone model trained for 36 epoch setting as the box AP score was highest for the 36 epoch setting compared to 12 and 24 epoch settings. With the link provided in the repository, we downloaded the checkpoint for the DINO-4scale R50 backbone, which was saved after 33 epochs. There were many other checkpoints corresponding to epochs 11 and 22 but we chose was for 33 epoch with the intuition that it would give us a better box AP values compared to other checkpoints.

3.2 Data Preparation and Analysis

The initial step involved preparing the pedestrian dataset. The annotations were converted from Pascal VOC XML format to a format compatible with the

DINO model (json format). This conversion was essential to ensure the model could interpret the data correctly. The category_id for the images in our dataset was initialized to 1, which maps to the class person and since our dataset deals with pedestrians, it looked meaningful to map images to the class person with category_id 1. The dataset was also visually analyzed by plotting bounding boxes on the images.

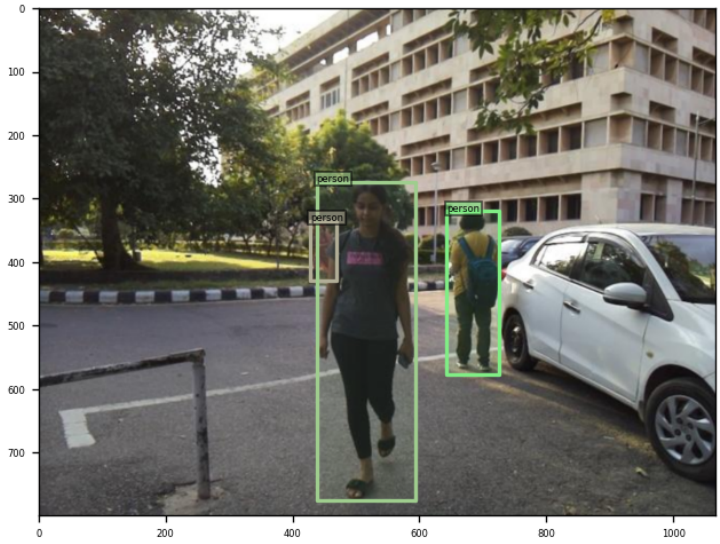

Figure 1: Example of bounding box plotting on pedestrian dataset.

4 Visualizing model predictions

We take a sample image 4 and pass it through the pre-trained model for inference and visualize the bounding boxes drawn by the model. The following image contains the visualization of the bounding boxes for the sample image,

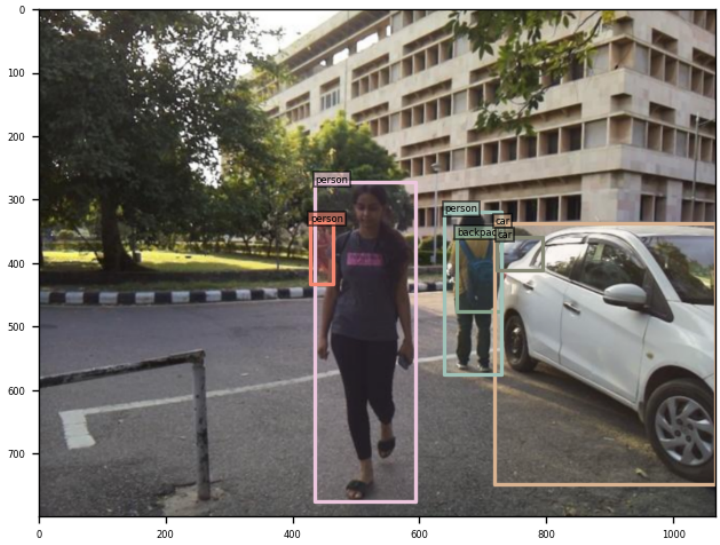

Figure 2: Model predictions for the sample image 4

4.1 Results for inference using validation pedestrian dataset

The following results are obtained after running the pre-trained DINO-4scale R50 backbone model on the validation dataset containing pedestrian images.

```
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.397
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.740
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.418
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.422
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.095
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.431
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.502
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.502
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = -1.000
```

Figure 3: box AP values for the validation dataset

4.2 Visualizing bounding boxes after each decoder layer

The bounding boxes are visualized based on the output of each decoder layer. The pretrained-model has **6 decoder layers**. So, we have 6 images one after each layer.

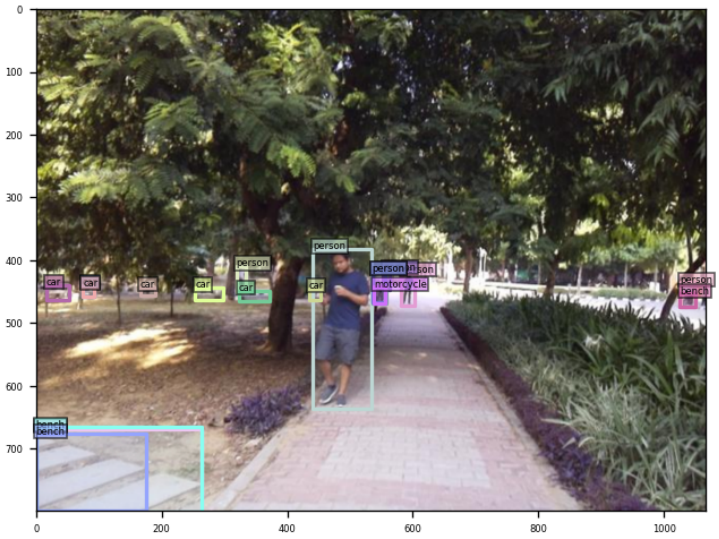

Figure 4: bounding boxes after 1st decoder layer

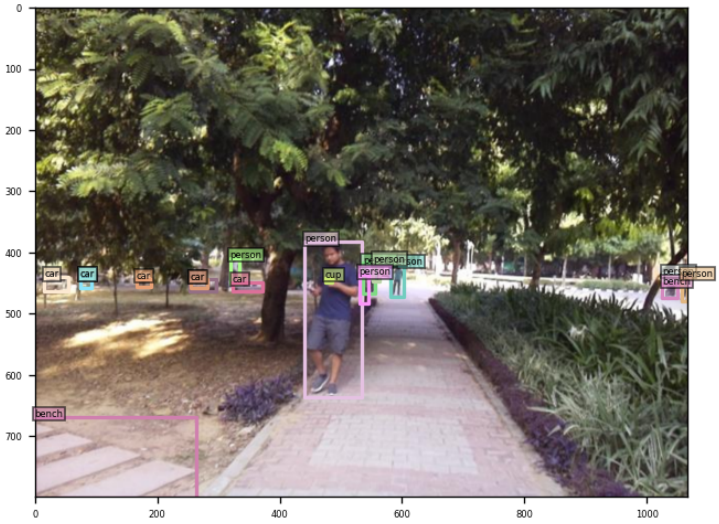

Figure 5: bounding boxes after 3rd decoder layer

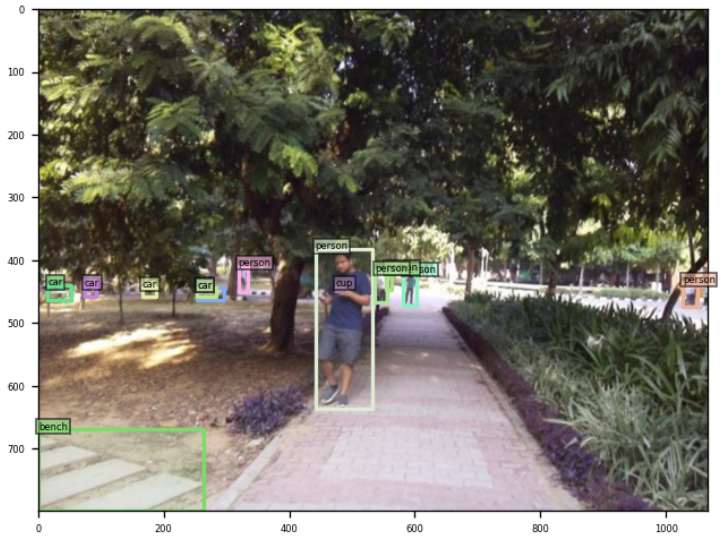

Figure 6: bounding boxes after 4th decoder layer

Figure 7: bounding boxes after 5th decoder layer

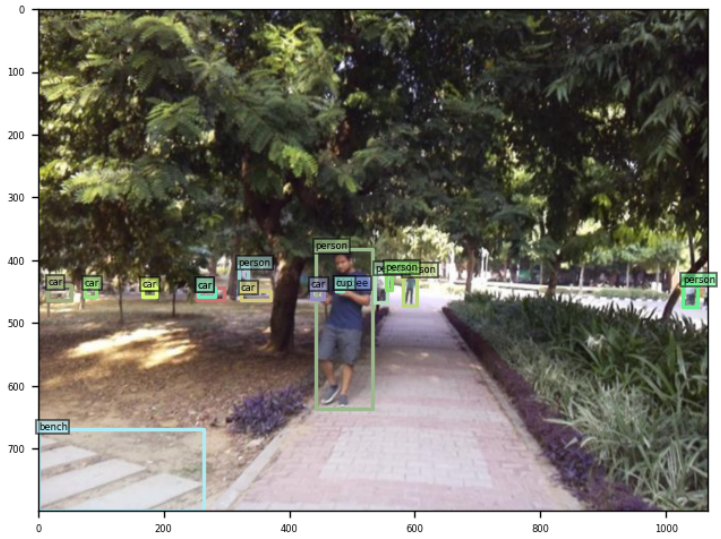

Figure 8: bounding boxes after 6th decoder layer

5 Model Fine-Tuning

The DINO model was fine-tuned on our dataset using the pre-trained checkpoint. Following are the results after fine-tuning after training on pedestrain dataset.

```
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.574
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.920
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.600
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.595
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.107
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.578
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.724
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.724
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = -1.000
Training time 0:07:11
```

Figure 9: box AP values after finetuning on pedestrian train dataset

The finetuned model is now evaluated on the predestrain valid dataset and the results are as follows,

```
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.586
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.909
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.658
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.613
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.108
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.591
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.711
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.711
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = -1.000
```

Figure 10: box AP values for pedestrain eval dataset after finetuning

We can clearly see that the results improved compared to the table 4.1

6 Conclusion

This report presented a comprehensive overview of the process and results of training and evaluating the DINO object detection model on the given pedestrian dataset.