Fine-Tuning and Evaluation of DINO Model for Object Detection

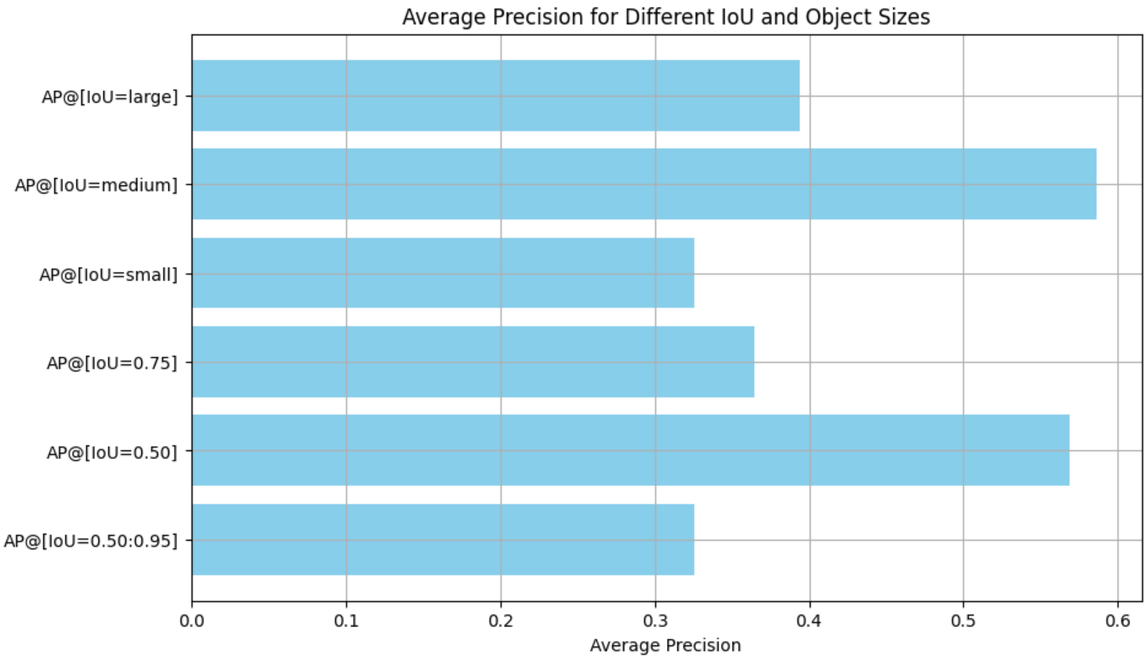
Introduction

In this project my task is to train the DINO object detection model on a pedestrian dataset consisting of 200 images collected within the IIT Delhi campus. The dataset, annotated in COCO format, includes both images and corresponding annotations in JSON format (link provided below). But I have made changes in dataset and customized it by adding 4 classes consisting of Person , Car , Animal , Bike. The goal is to enhance the model's ability to detect these objects accurately under varied conditions, leveraging Average Precision (AP) and Average Recall (AR) metrics to evaluate performance across object sizes.

Experimental Setup

1) Dataset: I have made changes in dataset by adding the 4 classes in it and to also performed Data Augmentation to increase the size of the dataset for better training of the Model.
2) Model Architecture : The model is DINO with ResNet-50 backbone and fine-tuned for 50 epochs to improve the object detection accuracy.
3) Hyperparameters : learning rate = 0.00001 , Batch size = 12 and Number Epochs :50

Results and Visualization



The Average Precision (AP) values on the validation set were as follows:

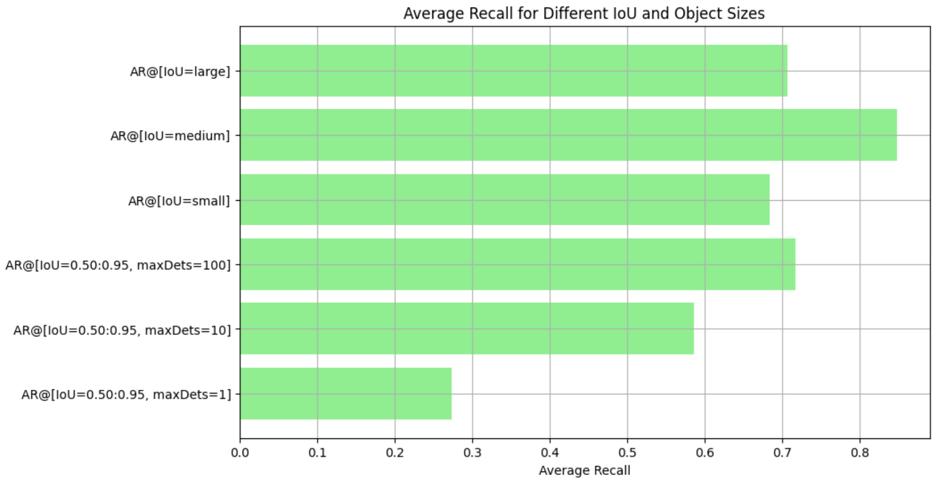
Metric	AP Value
AP (IoU=0.50:0.95, area=all)	0.326
AP (IoU=0.50, area=all)	0.569
AP (IoU=0.75, area=all)	0.365
AP (IoU=0.50:0.95, area=small)	0.326
AP (IoU=0.50:0.95, area=medium)	0.586
AP (IoU=0.50:0.95, area=large)	0.394

These AP values show moderate detection accuracy, with the model performing best on medium-sized objects, achieving an AP of 0.586, while small objects scored lower at 0.326.

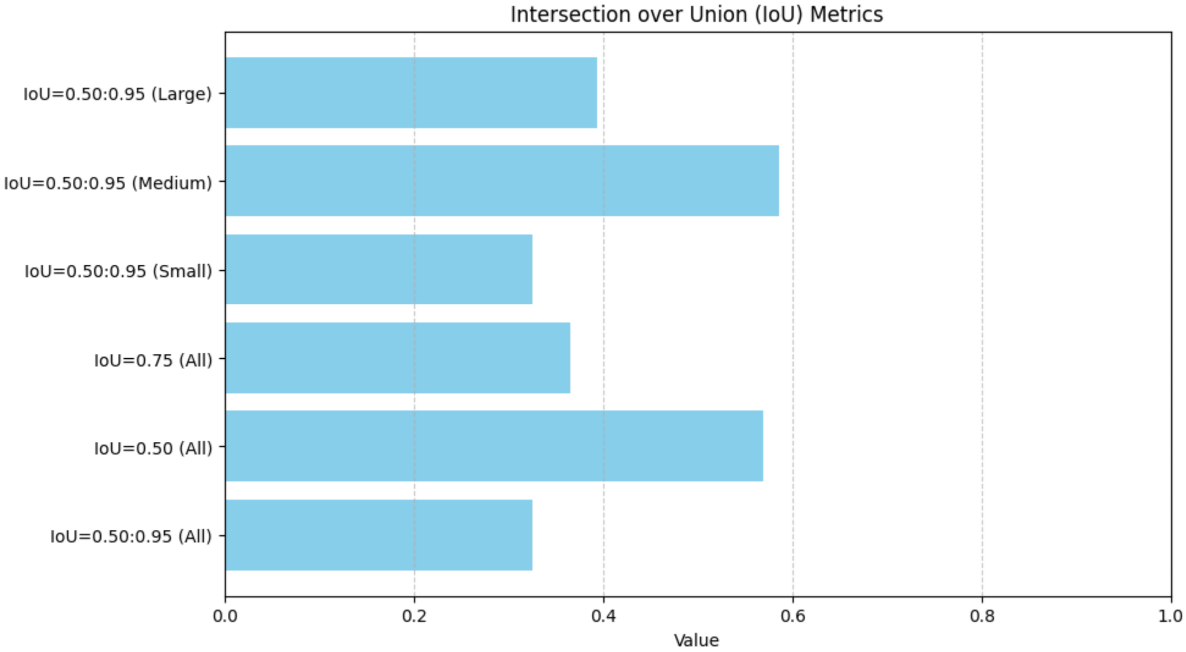
The Average Recall (AR) values for the bounding box predictions on the validation set are as follows:

Metric	AR value
AR (IoU=0.50:0.95, area=all, maxDets=1)	0.273
AR (IoU=0.50:0.95, area=all, maxDets=10)	0.586
AR (IoU=0.50:0.95, area=all,maxDets=100)	0.717
AR (IoU=0.50:0.95, area=small,maxDets=100)	0.684
AR (IoU=0.50:0.95, area=medium, maxDets=100)	0.848
AR (IoU=0.50:0.95, area=large, maxDets=100)	0.707

An overall AR of 0.717 with max detections of 100 suggests good detection performance across various object sizes. The model performs best with medium-sized objects (AR of 0.848), slightly lower for large objects (AR of 0.707), and lower still for small objects (AR of 0.684), indicating that while the model reliably detects most objects, it struggles slightly more with smaller items. These insights highlight potential areas for further refinement, particularly for small object detection.

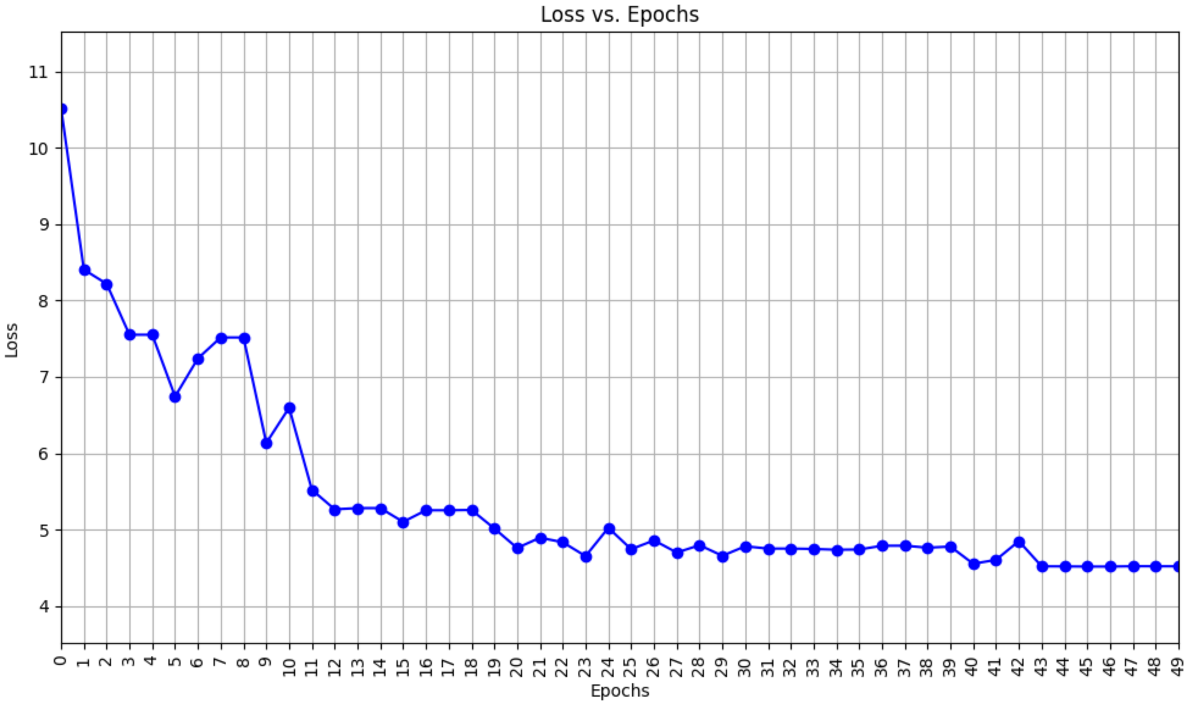


The overall IoU score for all objects, using the standard IoU threshold of 0.50:0.95, is 0.326, reflecting moderate accuracy in object localization. The model achieves a significantly higher IoU of 0.569 at the IoU=0.50 threshold, showing improved performance when a lower overlap is required for a true positive.

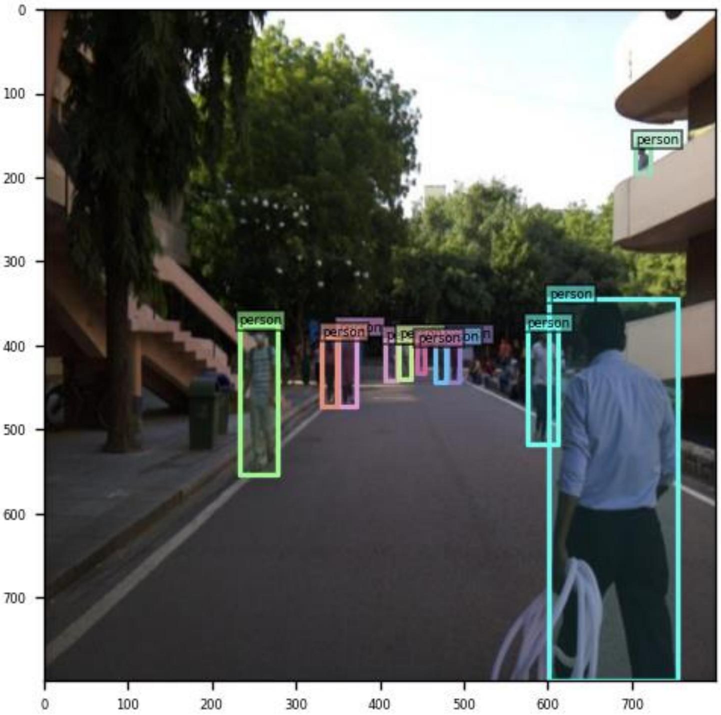


Detection accuracy is strongest with medium-sized objects (IoU=0.50:0.95, Medium = 0.586), while it is lower for large (0.394) and small objects (0.326). From this we can deduce that the model is well trained to detect medium sized objects and need for fine tuning to work well with small and large size objects.

From Loss vs Epoch graph we can see the Loss has decreased significantly but after a certain number of Epochs i.e 26 , the loss is almost constant. To tackle this problem we have to increase the number of images in dataset and each class should be normalize to avoid making model baised.



In this result we can see that model has perfectly detected the persons but not all of them detected this is due to lack of normalized data. If we increase the images in the dataset which has normalized images of each class this obstacle can be tackled easily,



Analysis of Results

From the results we can conclude that the model effectively detects medium and large objects but struggles with small objects, likely due to limited fine details or occlusions in the dataset. The lower AP for small objects (0.326) highlights the model's sensitivity to object size. This limitation may arise from insufficient diversity in small object instances during training, or the ResNet-50 backbone’s limited capacity to handle intricate features for smaller objects.

Possible solutions for improving small object detection could include:

1. Increasing dataset samples with small object variations.
2. Applying advanced augmentation techniques to simulate more complex scenarios.
3. Experimenting with a backbone better suited for small object detection.