Cat & Dog Detection

In this experiment, I trained a YOLOv5 model on a dataset of annotated images using the "train.py" script with a batch size of 16, 100 epochs, and an image size of 640. The dataset was split into two folders, one for training and one for validation, and the images were annotated using a third-party app and stored in the "label" folder. The dataset configuration was specified using the "coco128.yaml" file, which was customized for this specific model. Pre-trained weights for the YOLOv5s architecture were used as a starting point for training.

After training, I evaluated the model using the "detect.py" script on the validation set. I used the best-performing weights from the training and a confidence threshold of 0.25 to determine object detections. The source images for the validation set were located in the "/content/training/images/val" folder. The output of the experiment included various metrics such as box loss, object loss, class loss, precision, recall, and mAP scores for both the training and validation sets at each epoch. The learning rate was also displayed for each of the model's three layers.

To improve the model's performance, I suggested several approaches, including increasing the dataset size, fine-tuning hyperparameters, adjusting the confidence threshold, incorporating ensemble methods, using different architectures, and using transfer learning. These methods could help to enhance the model's precision and recall, reduce false positives, and ultimately improve its performance.

Based on my observations from the experiment, I found that the YOLOv5 model achieved decent results on the given dataset. However, I also noticed that there was room for improvement in terms of precision and recall. Therefore, I recommended exploring different approaches to improve the model's performance, such as adjusting the hyperparameters and trying out different architectures. Additionally, increasing the dataset size could also help to enhance the model's performance, especially when combined with data augmentation techniques.