**Yolov5 custom model training project description:**

YOLOv5 is a popular object detection algorithm that uses a deep neural network to detect objects in images. Custom YOLOv5 training involves fine-tuning the existing pre-trained YOLOv5 model to detect objects specific to a particular use case. In our case, we use the yolov5 small version model for training our custom dataset which includes cat and dog images and our main target is to detect the cat and dog image with the bounding box. This involves three main steps:

1. **Data preparation**: Collecting and preparing annotated images for training, validation, and testing.
2. **Model training**: Fine-tuning the pre-trained YOLOv5 model on the annotated images using transfer learning techniques, which involves adjusting the model's weights to fit the new dataset.
3. **Evaluation**: Testing the trained model on a separate set of annotated images to evaluate its performance and adjust the hyperparameters if necessary.

**Data preparation(Roboflow):**

For data preparation, we use the (Roboflow.com) website for the annotation of the dataset and for splitting and transforming the dataset.

For preparing the dataset with Roboflow these steps are followed:

1. Sign up or log in to Roboflow.
2. Create a new dataset by clicking on the "Datasets" tab and selecting "New Dataset".
3. Select the source of your data (e.g., upload images or connect to a cloud storage service) and follow the prompts to import the data into Roboflow.
4. Once our data is imported, you can start preprocessing it by creating a new "Export" from the "Exports" tab.
5. Select "YOLOv5" as the format for export and choose the desired image size and annotation type (e.g., "Pascal VOC" or "YOLO Darknet").
6. Choose any additional preprocessing options may need, such as data augmentation or filtering by object class.
7. Generate the export and download the resulting ZIP file containing the processed data.
8. Use the downloaded data to train the YOLOv5 model using our preferred framework or platform.

**Model training:**

1. Install YOLOv5: First, we need to install YOLOv5 on our machine. we can use pip to install it by running the following command:

!pip install yolov5

2. Clone the YOLOv5 repository: Clone the YOLOv5 repository from GitHub using the following command:

!git clone <https://github.com/ultralytics/yolov5.git>

3. Download the Roboflow dataset: Once we prepared the dataset with Roboflow, download it in the YOLOv5 format (YOLOv5 PyTorch .yaml format).

4. Define the YOLOv5 configuration: In the cloned YOLOv5 repository, we need to create a new file my\_yolo.yaml, and copy the contents of the downloaded Roboflow .yaml file into it. Also, set the paths to the train and val directories as follows:

train: /path/to/train/directory

val: /path/to/val/directory

5. Train the model: Use the following command to train the YOLOv5 model with the Roboflow dataset:

!python yolov5/train.py --img 640 --batch 16 --epochs 50 --data my\_yolo.yaml --weights yolov5s.pt

Here, we are training a YOLOv5s model with an image size of 640, and batch size of 16, for 50 epochs using the my\_yolo.yaml configuration file, and starting with the pre-trained yolov5s.pt weights.

**Evaluation:**

1. Monitor the training: During the training, we can monitor the progress of the training and see the loss and other metrics using Tensorboard. We use the following command to start Tensorboard:

!tensorboard --logdir runs/train

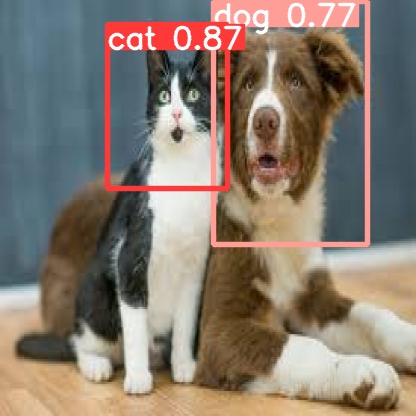
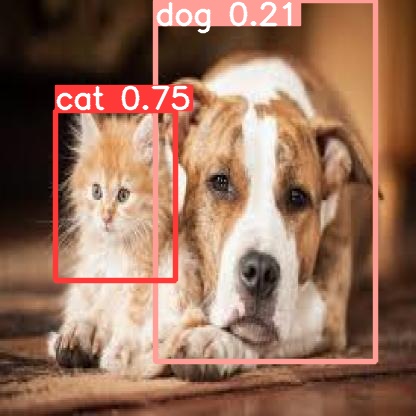
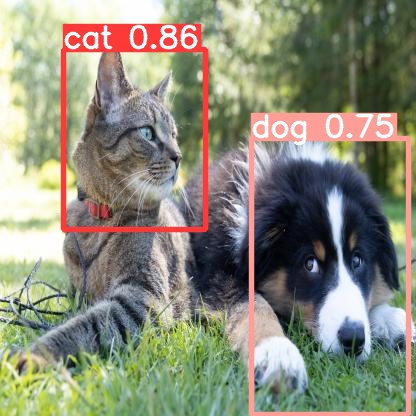
2. Evaluate the trained model: After the training is complete, evaluate the performance of the trained model on the validation set using the following command:

!python yolov5/val.py --img 640 --batch 16 --data my\_yolo.yaml --weights runs/train/exp/weights/best.pt

Here, we are evaluating the best weights obtained during training on the validation set.

That's it! These are the basic steps to train a YOLOv5 model using a dataset prepared with Roboflow. We can further fine-tune the model by changing the hyperparameters and other training settings.

**Observations about the yolov5 custom training project:**

****

From the above images, we saw Yolov5 model accurately detects the test image classes such as cat and dog which is good in sound. Moreover, our images are insufficient, if we provide more images for training purposes, our model can perform very well as we expected.