Yolo5:

1.[!git clone https://github.com/ultralytics/yolov5
%cd yolov5
!git reset --hard 886f1c03d839575afecb059accf74296fad395b6]

- Github clone repository

2.

[!pip install -qr requirements.txt # install dependencies (ignore errors)
import torch

from IPython.display import Image, clear_output *# to display images*
from utils.google_utils import gdrive_download *# to download models/datasets*

 # clear_output()
print('Setup complete. Using torch %s %s' % (torch.__version__,
torch.cuda.get_device_properties(0) if torch.cuda.is_available() else 'CPU'))]

- Here are the dependencies that being installed
- Import image and models
-

3.
Here we use “yolov5 pytorch” YAML file export format. Yaml file is commonly used for configuration files. + it is kind of like json in dataset but the format is different.

4.
cat {dataset.location}/data.yaml
In here **concatenate** is used for to display the content of files without having to open the file for editing.

5.

```
# define number of classes based on YAML
import yaml
with open(dataset.location + "/data.yaml", 'r') as stream:
    num_classes = str(yaml.safe_load(stream)['nc'])
```

6.

```
#this is the model configuration we will use for our tutorial
%cat /content/yolov5/models/yolov5s.yaml
```

NC: 80??

Model depth means add more layers to the neural net
And Width means add more filters to the layers so it adds more channels to the layer output

7.

Anchors:
anchor boxes are **a set of predefined bounding boxes of a certain height and width**

8.
Yolov5 configuration:
Yolov5 model architecture consists of three parts model backbone,model neck and model head

Backbone:
Model backbone is mainly used to extract important features from the given input image.
Model head:
The model Head is mainly used to perform the final detection part. It applied anchor boxes on features and generates final output vectors with class probabilities, objectness scores, and bounding boxes.

9.
```
#customize iPython writefile so we can write variables
from IPython.core.magic import register_line_cell_magic

@register_line_cell_magic
def writetemplate(line, cell):
    with open(line, 'w') as f:
        f.write(cell.format(**globals()))
```

It basically enable the variables to use

10.
%writetemplate ???

11. Batch: refers to the batch size (number of training examples utilized in one iteration).

12. Weights: Model weights are **all the parameters (including trainable and non-trainable) of the model which are in turn all the parameters used in the layers of the model**

13. Cache image: Cached data are files, scripts, images, and other multimedia stored on your device after opening an app or visiting a website for the first time
14. %%time-??
14. Box:
loss due to a box prediction not exactly covering an object.

15. Objectness:
loss due to a wrong box-object IoU [1] prediction.
16. Classifications:
loss due to deviations from predicting '1' for the correct classes and '0' for all the other classes for the object in that box.

17. Precision: measures how accurate are the predictions. It is the percentage of your correct predictions

Recall: measures how good it finds all the positives

mAP (mean Average Precision) compares the ground-truth bounding box to the detected box and returns a score. The higher the score, the more accurate the model is in its detections.

- **mAP@ 0.5 : when IoU is set to 0.5, the AP [2] of all pictures of each category is calculated, and then all categories are averaged : mAP**
- **mAP@ 0.5:0.95 : represents the average mAP at different IoU thresholds (from 0.5 to 0.95 in steps of 0.05)**

[1] IoU (Intersection over Union) = measures the overlap between two boundaries. It is used to measure how much the predicted boundary overlaps with the ground truth

Faster Rcnn:

1. Repository link

'https://github.com/roboflow-ai/tensorflow-object-detection-faster-rcnn'

2. !git pull-

he git pull command is used to fetch and download content from a remote repository and immediately update the local repository to match that content.

3. **%ls- list the contents of the current directory**

4.Shutil- Shutil module offers high-level operation on a file like a copy, create, and remote operation on the file

5. Tarfile- used to read and write tar archives

6. Urlib- URL handling module for python

7.