

Ultralytics YOLOv5

Install Pytorch

The screenshot shows the PyTorch website's download page. At the top, there are navigation links: Learn, Ecosystem, Edge, Docs, Blog & News, About, Become a Member, and a search icon. Below the navigation, a note states: "NOTE: Latest PyTorch requires Python 3.9 or later." To the right, there are links to Amazon Web Services, Google Cloud Platform, and Microsoft Azure.

The main content area features a grid of options for selecting PyTorch components:

PyTorch Build	Stable (2.5.1)		Preview (Nightly)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
Compute Platform	CUDA 11.8	CUDA 12.1	CUDA 12.4	ROCM 6.2

Below the grid, a "Run this Command:" section contains the terminal command: `pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu124`.

At the bottom left, a button says "Previous versions of PyTorch >".

The screenshot shows a terminal window with the following session:

```
PS C:\Users\MTechno\OneDrive\Documents\My Courses\A-Projects\Computer Vision -----\\yolo> python -m venv virt_en
PS C:\Users\MTechno\OneDrive\Documents\My Courses\A-Projects\Computer Vision -----\\yolo> virt_en\\scripts\\activate
(virt_en) PS C:\Users\MTechno\OneDrive\Documents\My Courses\A-Projects\Computer Vision -----\\yolo> pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu124
Looking in indexes: https://download.pytorch.org/whl/cu124
Collecting torch
  Downloading https://download.pytorch.org/whl/cu124/torch-2.5.1%2Bcu124-cp311-cp311-win_amd64.whl (2510.8 MB)
    0.0/2.5 GB 2.8 MB/s eta 0:15:05
```

Download the Modal related files-- Yolov5-- Ultralytics

1 - Clone the model from GitHub

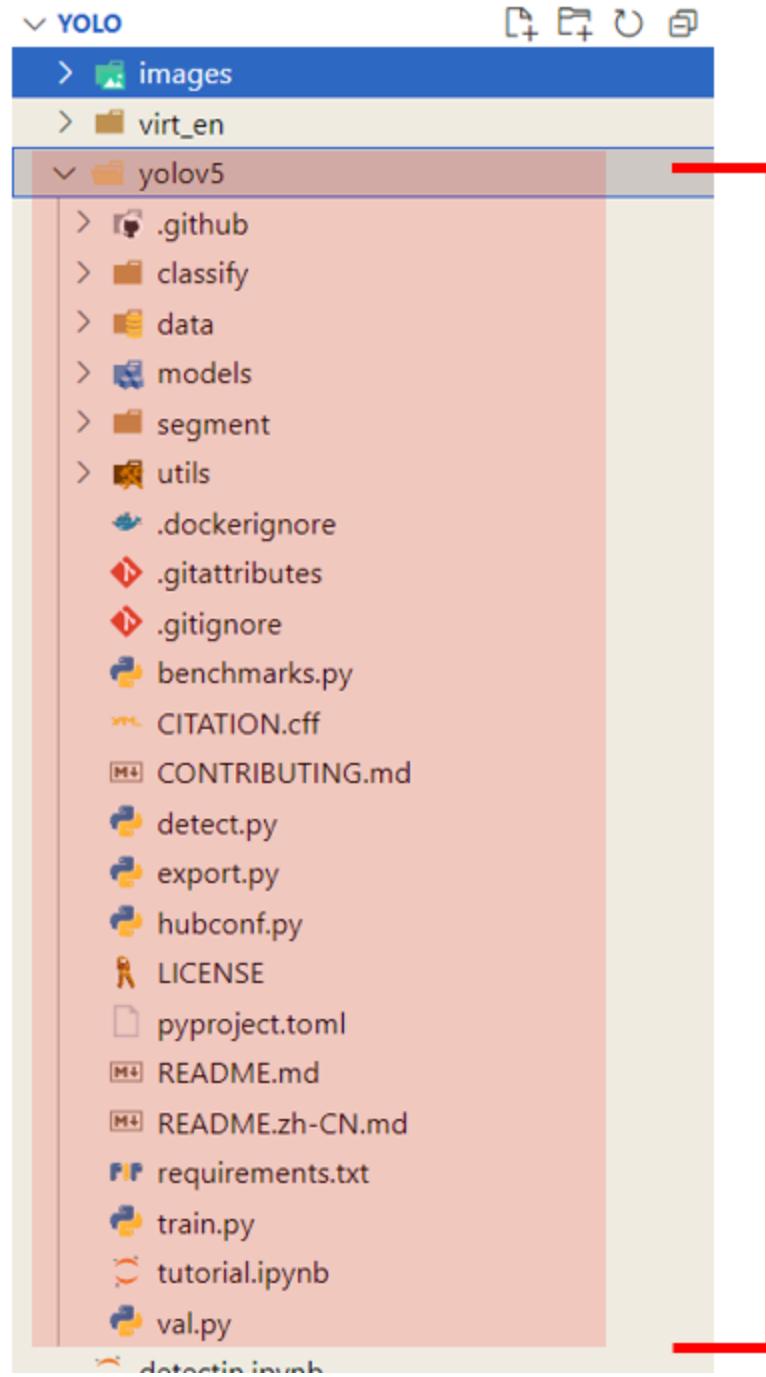
The screenshot shows the GitHub repository page for `ultralytics/yolov5`. The repository has 168 issues, 21 pull requests, and 16.5k forks. The code tab is selected, showing a list of commits from `glen-jocher`. The sidebar includes a summary for `Monica`, details about the repo's purpose (YOLOv5 in PyTorch), and a list of supported models and tags.

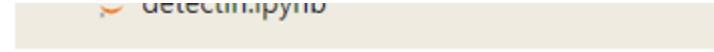
Commit	Message	Time Ago
glenn-jocher Delete .github/workflows/codeql-analysis.yml (#13431)	882c35f · 2 weeks ago	2.872 Commits
.github	Delete .github/workflows/codeql-analysis.yml (#13431)	2 weeks ago
classify	Ultralytics Refactor #13368	2 months ago
data	Ultralytics Refactor #13368	2 months ago
models	Add https://www.reddit.com/r/Ultralytics/ badge (#13284)	4 months ago
segment	Ultralytics Refactor #13368	2 months ago
utils	Fix possible image drawing issues (#13426)	2 weeks ago
.dockerrignore	Add .git to .dockerrignore (#8815)	2 years ago
.gitattributes	git attrib	4 years ago
.gitignore	Update CoreML exports to support newer *.mlpackage outp...	5 months ago
CITATION.cff	Update LICENSE to AGPL-3.0 (#11359)	last year
CONTRIBUTING.md	Ultralytics Code Refactor #13...	3 months ago

```
In [8]: # clone the model
!git clone https://github.com/ultralytics/yolov5
```

```
Cloning into 'yolov5'...
```







Install the requirements

```
In [9]: !cd yolov5 & pip install -r requirements.txt
```

Requirement already satisfied: gitpython>=3.1.30 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 5)) (3.1.43)
Requirement already satisfied: matplotlib>=3.3 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 6)) (3.9.3)
Requirement already satisfied: numpy>=1.23.5 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 7)) (1.26.3)
Requirement already satisfied: opencv-python>=4.1.1 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 8)) (4.10.0.84)
Requirement already satisfied: pillow>=10.3.0 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 9)) (11.0.0)
Requirement already satisfied: psutil in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 10)) (6.1.0)
Requirement already satisfied: PyYAML>=5.3.1 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 11)) (6.0.2)
Requirement already satisfied: requests>=2.32.2 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 12)) (2.32.3)
Requirement already satisfied: scipy>=1.4.1 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 13)) (1.14.1)
Requirement already satisfied: thop>=0.1.1 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 14)) (0.1.1.post2209072238)
Requirement already satisfied: torch>=1.8.0 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 15)) (2.5.1+cu124)
Requirement already satisfied: torchvision>=0.9.0 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 16)) (0.20.1+cu124)
Requirement already satisfied: tqdm>=4.66.3 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 17)) (4.67.1)
Requirement already satisfied: ultralytics>=8.2.34 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 18)) (8.3.44)
Requirement already satisfied: pandas>=1.1.4 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 27)) (2.2.3)
Requirement already satisfied: seaborn>=0.11.0 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 28)) (0.13.2)
Requirement already satisfied: setuptools>=70.0.0 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from -r requirements.txt (line 42)) (75.6.0)
Requirement already satisfied: gitdb<5,>=4.0.1 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from gitpython>=3.1.30->-r requirements.txt (line 5)) (4.0.11)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from matplotlib>=3.3->-r requirements.txt (line 6)) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from matplotlib>=3.3->-r requirements.txt (line 6)) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from matplotlib>=3.3->-r requirements.txt (line 6)) (4.55.2)

Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from matplotlib>=3.3->-r requirements.txt (line 6)) (1.4.7)
Requirement already satisfied: packaging>=20.0 in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from matplotlib>=3.3->-r requirements.txt (line 6)) (24.2)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from matplotlib>=3.3->-r requirements.txt (line 6)) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from matplotlib>=3.3->-r requirements.txt (line 6)) (2.9.0.post0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from requests>=2.32.2->-r requirements.txt (line 12)) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from requests>=2.32.2->-r requirements.txt (line 12)) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from requests>=2.32.2->-r requirements.txt (line 12)) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from requests>=2.32.2->-r requirements.txt (line 12)) (2024.8.30)
Requirement already satisfied: filelock in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (3.13.1)
Requirement already satisfied: typing-extensions>=4.8.0 in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (4.9.0)
Requirement already satisfied: networkx in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (3.2.1)
Requirement already satisfied: jinja2 in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (3.1.3)
Requirement already satisfied: fsspec in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (2024.2.0)
Requirement already satisfied: sympy==1.13.1 in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from sympy==1.13.1->torch>=1.8.0->-r requirements.txt (line 15)) (1.3.0)
Requirement already satisfied: colorama in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from tqdm>=4.66.3->-r requirements.txt (line 17)) (0.4.6)
Requirement already satisfied: py-cpuinfo in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from ultralytics>=8.2.34->-r requirements.txt (line 18)) (9.0.0)
Requirement already satisfied: ultralytics-thop>=2.0.0 in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from ultralytics>=8.2.34->-r requirements.txt (line 18)) (2.0.12)
Requirement already satisfied: pytz>=2020.1 in c:\users\mtechno\onedrive\documents\my courses\my projects\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from pandas>=1.1.4->-r requirements.txt (line 27)) (2024.2)

```
Requirement already satisfied: tzdata>=2022.7 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from pandas>=1.1.4->-r requirements.txt (line 27)) (2024.2)
Requirement already satisfied: smmap<6,>=3.0.1 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from gitdb<5,>=4.0.1->gitpython>=3.1.30->-r requirements.txt (line 5)) (5.0.1)
Requirement already satisfied: six>=1.5 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from python-dateutil>=2.7->matplotlib>=3.3->-r requirements.txt (line 6)) (1.17.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\mtechno\onedrive\documents\my courses\computer vision -----\\yolo\\virt_en\\lib\\site-packages (from jinja2->torch>=1.8.0->-r requirements.txt (line 15)) (2.1.5)
```

Import Needed Libraries

```
In [4]: import torch # type: ignore
from matplotlib import pyplot as plt # type: ignore
import numpy as np # type: ignore
import cv2 # type: ignore
```

Libraries

- **Torch** -- Will be used to load the YOLO model and make detections.
- **Matplotlib** -- Will be used for visualizing images.
- **OpenCV** -- Helps access the webcam and render feeds.
- **Numpy** -- Will be used for array transformation.

Load the Pretrained Model from Torch Hub

if i already downloaded the model - why do i need to load it ?

1 - Downloading a Model

This step fetches the model file from a source like a server - repository - hub and saves it to our local storage. The downloaded file is passive - it's just stored data at this stage.

2 - Loading a Model

This step **reads** the downloaded file into **memory** and **prepares** it for **inference** or **training**. The model is **structured** into **objects** and **data structures** that the framework - like TensorFlow - PyTorch - ONNX can understand and execute. Loading involves **deserialization** - where the saved model **weights** and **architecture** are reconstructed into an **active - ready-to-use object**.

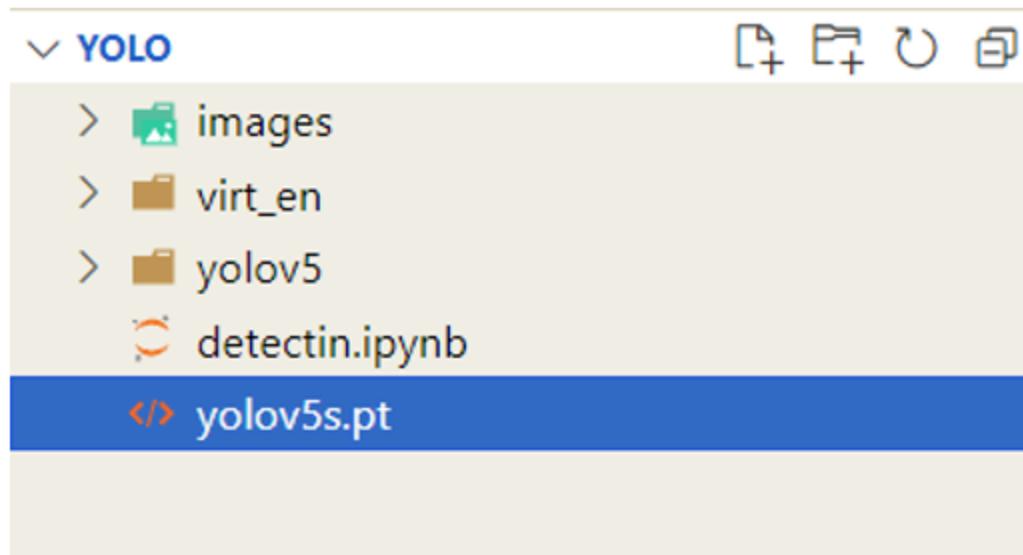
In [5]:

```
# Load the yolov5 downloaded model
# s-- for small
# from ultralytics/yolov5 we load yolov5s
model = torch.hub.load('ultralytics/yolov5', 'yolov5s')
```

```
Using cache found in C:\Users\MTTechno/.cache\torch\hub\ultralytics_yolov5_master
YOLOv5 2024-12-16 Python-3.11.1 torch-2.5.1+cu124 CUDA:0 (NVIDIA GeForce GTX 1050 with Max-Q Design, 3072MiB)
```

```
Fusing layers...
YOLOv5s summary: 213 layers, 7225885 parameters, 0 gradients, 16.4 GFLOPs
Adding AutoShape...
```

Remark -- The cloned folder of YOLOv5 we downloaded contains only configuration files **.yaml** and model scripts **.py**. It does not contain the actual model weights **.pt** which are typically available through PyTorch Hub.



yolov5s.pt is the **pre-trained model file** for YOLOv5.

It contains the **weights of the neural network** that has been **trained** on a **large dataset** - like COCO or custom data - for object detection tasks.

Format -- .pt stands for **PyTorch model file** - which is the **format** used to **store** the **trained weights** of a **PyTorch model**.

Pre-trained Weights -- The file contains the **learned parameters** - weights and biases - of the YOLOv5 model after being trained on a dataset.

Use -- It is used to make predictions on new images or videos. Once loaded, the model can detect objects in images based on the pre-trained knowledge.

The **yolov5s.pt** model file **already** has **pre-trained weights** even if we haven't trained the model on our own dataset yet.

These **weights** are the result of **training** the model on a **large** - publicly available **dataset** such as COCO or VOC - which is why it can already perform object detection on common objects without needing any additional training.



Learn

Ecosystem

Edge

Docs

Blog & News

About

Become a Member

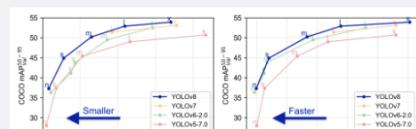


< YOLOV5

By Ultralytics

Ultralytics YOLOv5 🚀 for object detection, instance segmentation and image classification.

[View on Github](#) > [Open on Google Colab](#) > [Open Model Demo](#) >



BEFORE YOU START

Start from a **Python>=3.8** environment with **PyTorch>=1.7** installed. To install PyTorch see <https://pytorch.org/get-started/locally/>. To install YOLOv5 dependencies:

```
pip install -U ultralytics
```

In [15]: `# the different Layers available inside of this Model
model`

```
Out[15]: AutoShape(  
    (model): DetectMultiBackend(  
        (model): DetectionModel(  
            (model): Sequential(  
                (0): Conv(  
                    (conv): Conv2d(3, 32, kernel_size=(6, 6), stride=(2, 2), padding=(2, 2))  
                    (act): SiLU(inplace=True)  
                )  
                (1): Conv(  
                    (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))  
                    (act): SiLU(inplace=True)  
                )  
                (2): C3(  
                    (cv1): Conv(  
                        (conv): Conv2d(64, 32, kernel_size=(1, 1), stride=(1, 1))  
                        (act): SiLU(inplace=True)  
                    )  
                    (cv2): Conv(  
                        (conv): Conv2d(64, 32, kernel_size=(1, 1), stride=(1, 1))  
                        (act): SiLU(inplace=True)  
                    )  
                    (cv3): Conv(  
                        (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))  
                        (act): SiLU(inplace=True)  
                    )  
                (m): Sequential(  
                    (0): Bottleneck(  
                        (cv1): Conv(  
                            (conv): Conv2d(32, 32, kernel_size=(1, 1), stride=(1, 1))  
                            (act): SiLU(inplace=True)  
                        )  
                        (cv2): Conv(  
                            (conv): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
                            (act): SiLU(inplace=True)  
                        )  
                    )  
                )  
                (3): Conv(  
                    (conv): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))  
                    (act): SiLU(inplace=True)  
                )
```

```
(4): C3(
    (cv1): Conv(
        (conv): Conv2d(128, 64, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (cv2): Conv(
        (conv): Conv2d(128, 64, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (cv3): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (m): Sequential(
        (0): Bottleneck(
            (cv1): Conv(
                (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
            )
            (cv2): Conv(
                (conv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
                (act): SiLU(inplace=True)
            )
        )
        (1): Bottleneck(
            (cv1): Conv(
                (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
            )
            (cv2): Conv(
                (conv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
                (act): SiLU(inplace=True)
            )
        )
    )
)
(5): Conv(
    (conv): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (act): SiLU(inplace=True)
)
(6): C3(
    (cv1): Conv(
```

```
(conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
(act): SiLU(inplace=True)
)
(cv2): Conv(
(conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
(act): SiLU(inplace=True)
)
(cv3): Conv(
(conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
(act): SiLU(inplace=True)
)
(m): Sequential(
(0): Bottleneck(
(cv1): Conv(
(conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
(act): SiLU(inplace=True)
)
(cv2): Conv(
(conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(act): SiLU(inplace=True)
)
)
(1): Bottleneck(
(cv1): Conv(
(conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
(act): SiLU(inplace=True)
)
(cv2): Conv(
(conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(act): SiLU(inplace=True)
)
)
(2): Bottleneck(
(cv1): Conv(
(conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
(act): SiLU(inplace=True)
)
(cv2): Conv(
(conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(act): SiLU(inplace=True)
)
)
```

```
        )
    )
(7): Conv(
    (conv): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (act): SiLU(inplace=True)
)
(8): C3(
    (cv1): Conv(
        (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (cv2): Conv(
        (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (cv3): Conv(
        (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
)
(m): Sequential(
    (0): Bottleneck(
        (cv1): Conv(
            (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
        )
        (cv2): Conv(
            (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
            (act): SiLU(inplace=True)
        )
    )
)
(9): SPPF(
    (cv1): Conv(
        (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (cv2): Conv(
        (conv): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
)
(m): MaxPool2d(kernel_size=5, stride=1, padding=2, dilation=1, ceil_mode=False)
```

```
)  
(10): Conv(  
    (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))  
    (act): SiLU(inplace=True)  
)  
(11): Upsample(scale_factor=2.0, mode='nearest')  
(12): Concat()  
(13): C3(  
    (cv1): Conv(  
        (conv): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1))  
        (act): SiLU(inplace=True)  
)  
    (cv2): Conv(  
        (conv): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1))  
        (act): SiLU(inplace=True)  
)  
    (cv3): Conv(  
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))  
        (act): SiLU(inplace=True)  
)  
(m): Sequential(  
    (0): Bottleneck(  
        (cv1): Conv(  
            (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))  
            (act): SiLU(inplace=True)  
)  
        (cv2): Conv(  
            (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
            (act): SiLU(inplace=True)  
)  
    )  
)  
(14): Conv(  
    (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))  
    (act): SiLU(inplace=True)  
)  
(15): Upsample(scale_factor=2.0, mode='nearest')  
(16): Concat()  
(17): C3(  
    (cv1): Conv(  
        (conv): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1))
```

```
(act): SiLU(inplace=True)
)
(cv2): Conv(
    (conv): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1))
    (act): SiLU(inplace=True)
)
(cv3): Conv(
    (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
    (act): SiLU(inplace=True)
)
(m): Sequential(
    (0): Bottleneck(
        (cv1): Conv(
            (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
        )
        (cv2): Conv(
            (conv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
            (act): SiLU(inplace=True)
        )
    )
)
(18): Conv(
    (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (act): SiLU(inplace=True)
)
(19): Concat()
(20): C3(
    (cv1): Conv(
        (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (cv2): Conv(
        (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (cv3): Conv(
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
)
(m): Sequential(
```



```
)  
(24): Detect(  
    (m): ModuleList(  
        (0): Conv2d(128, 255, kernel_size=(1, 1), stride=(1, 1))  
        (1): Conv2d(256, 255, kernel_size=(1, 1), stride=(1, 1))  
        (2): Conv2d(512, 255, kernel_size=(1, 1), stride=(1, 1))  
    )  
)  
)  
)  
)  
)
```

Make Detections and Classification with the pretrained Yolov5s

- One Image
- A List of images

Print out the results

```
In [6]: image_path_1 = 'images_general_detection_testing/chien.jpg'  
image_path_2 = 'images_general_detection_testing/dogs.jpeg'  
image_path_3 = 'images_general_detection_testing/monkey.jpg'  
image_path_4 = 'images_general_detection_testing/po.png'
```

```
In [7]: results_4 = model(image_path_4)  
results_4.print() # ---> image 1/1: 478x1024 6 persons, 4 motorcycles
```

```
C:\Users\MTTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
with amp.autocast(autocast):  
image 1/1: 512x512 1 apple  
Speed: 141.8ms pre-process, 75.8ms inference, 350.1ms NMS per image at shape (1, 3, 640, 640)
```

```
In [16]: # give a single image to the model for detection and classification
results_1 = model(image_path_1)
results_1.print() # ---> image 1/1: 768x1280 6 cats, 10 dogs
```

C:\Users\MTTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
with amp.autocast(autocast):
image 1/1: 554x554 1 cat, 1 dog
Speed: 42.3ms pre-process, 150.1ms inference, 19.7ms NMS per image at shape (1, 3, 640, 640)

```
In [10]: results_2 = model(image_path_3)
results_2.print() # ---> image 1/1: 824x1100 1 bird
```

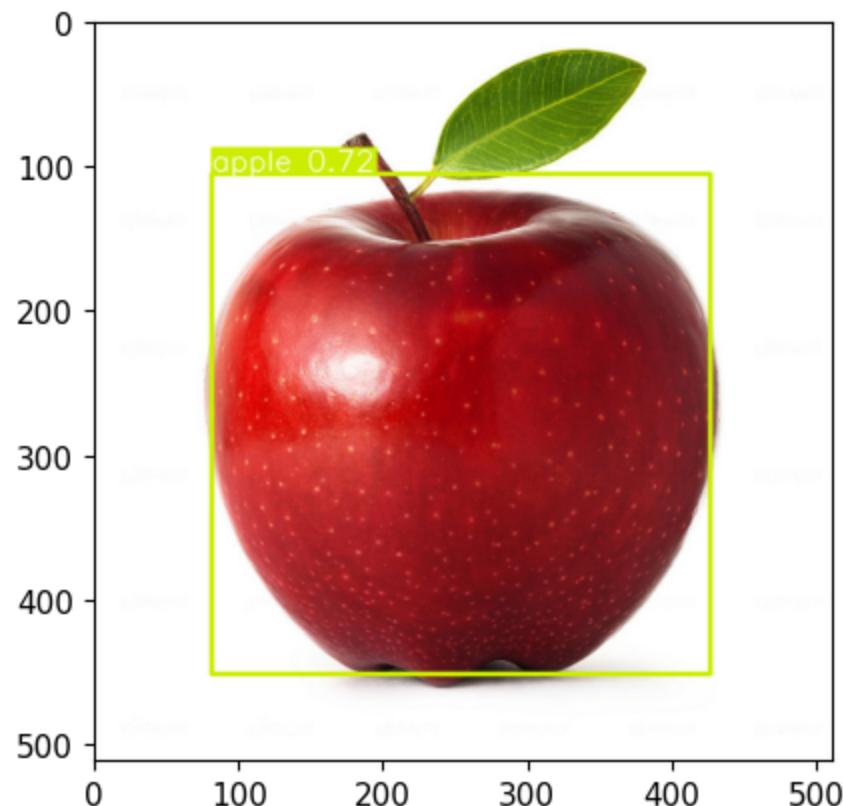
C:\Users\MTTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
with amp.autocast(autocast):
image 1/1: 824x1100 1 bird
Speed: 43.8ms pre-process, 78.3ms inference, 9.3ms NMS per image at shape (1, 3, 480, 640)

```
In [8]: # give multiple images to the model for detection and classification
results_3 = model([image_path_1, image_path_2])
results_3.print()
# --->
# image 1/2: 768x1280 7 cats, 9 dogs
# image 2/2: 768x1280 3 cats, 1 dogs
```

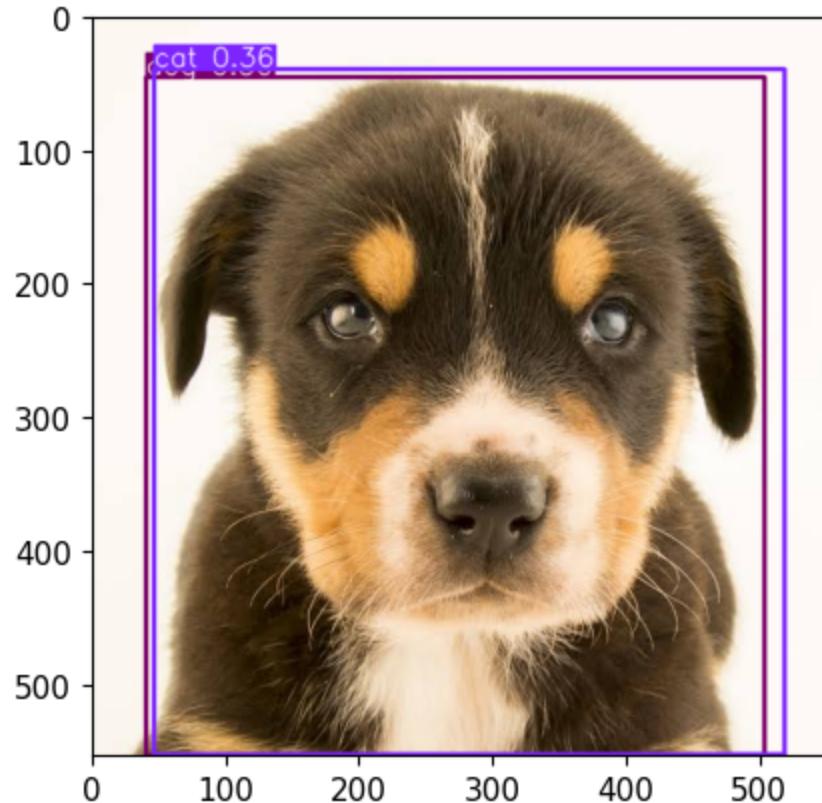
C:\Users\MTTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
with amp.autocast(autocast):
image 1/2: 768x1280 7 cats, 9 dogs
image 2/2: 183x275 3 cats, 1 dog
Speed: 15.1ms pre-process, 19.5ms inference, 34.1ms NMS per image at shape (2, 3, 448, 640)

Show the results

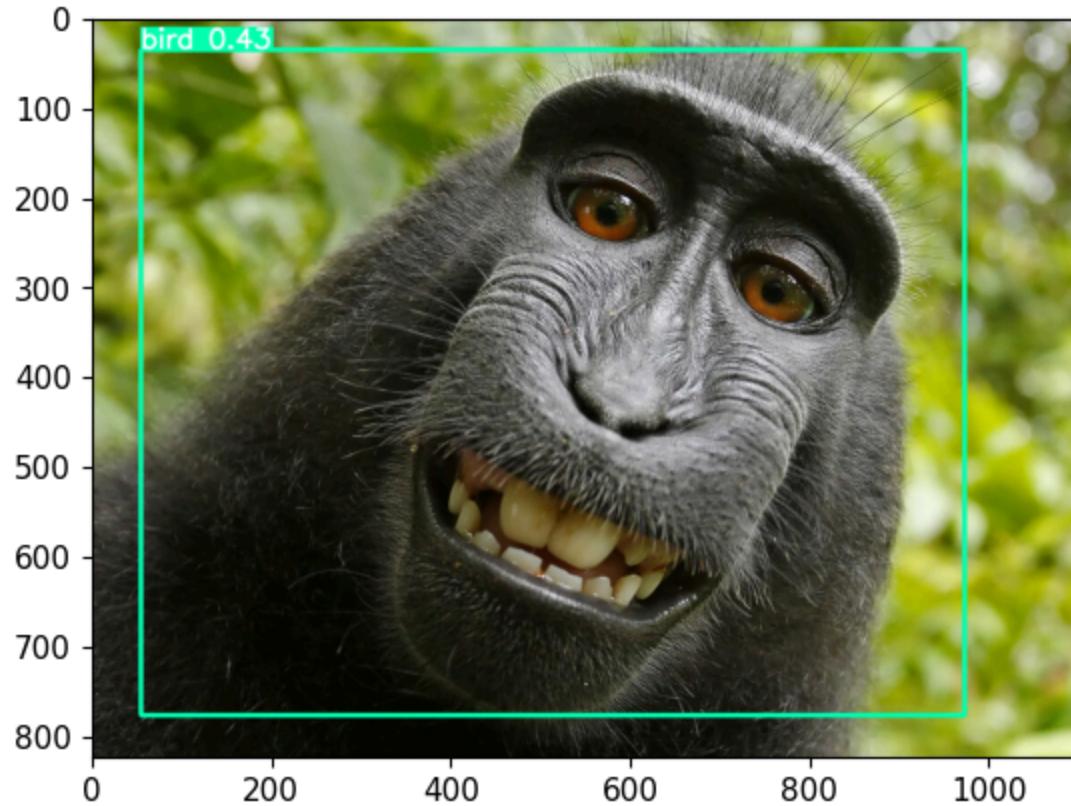
```
In [8]: %matplotlib inline
plt.imshow(np.squeeze(results_4.render()))
plt.show()
```



```
In [17]: %matplotlib inline  
plt.imshow(np.squeeze(results_1.render()))  
plt.show()
```



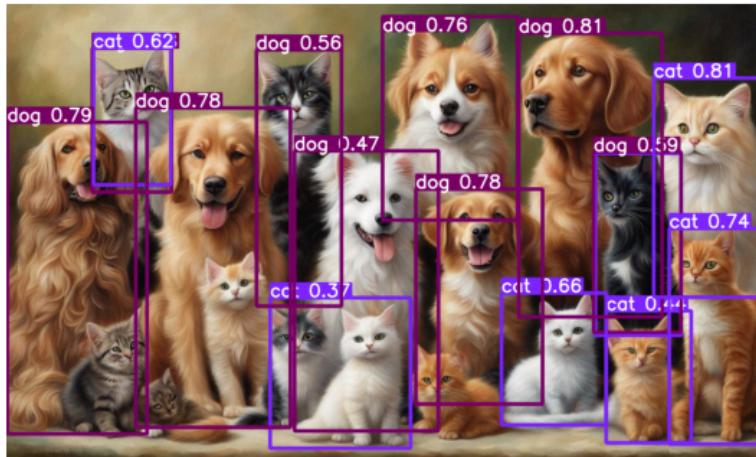
```
In [39]: %matplotlib inline  
plt.imshow(np.squeeze(results_2.render()))  
plt.show()
```



```
In [41]: # results_3.render() returns 2 images
# Create subplots for each image with larger figure size
fig, axes = plt.subplots(1, len(results_3.render()), figsize=(15, 10)) # Adjust figsize to make the plots larger

# Display each image in the corresponding subplot
for i, img in enumerate(results_3.render()):
    axes[i].imshow(np.squeeze(img))
    axes[i].axis('off') # Hide axis labels

plt.show()
```

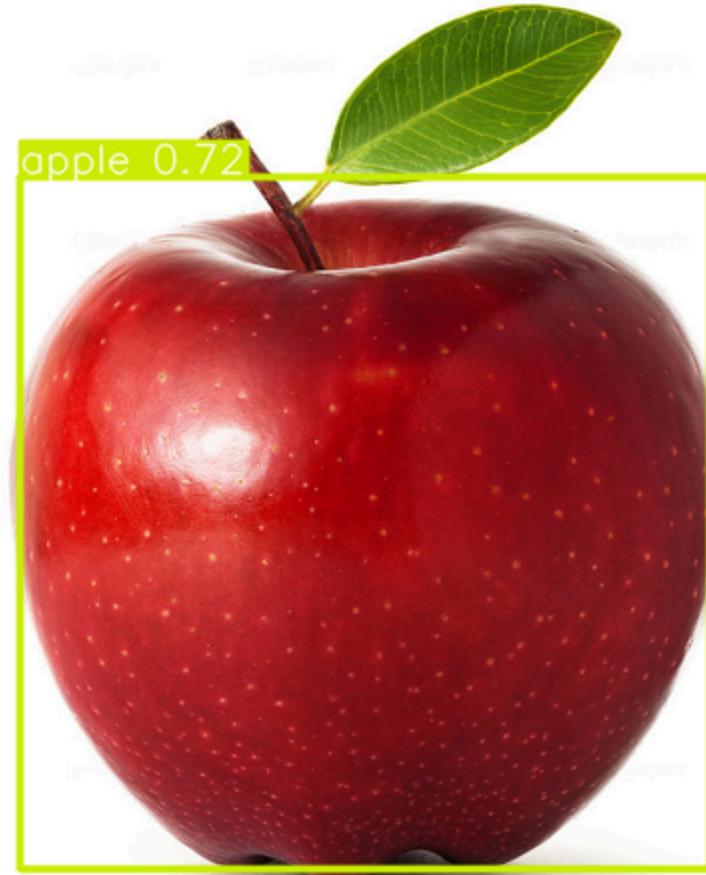


```
In [48]: # This methods returns a List of NumPy arrays - where each array corresponds to an image with the annotations drawn on it
results_2.render()
```

```
Out[48]: [array([[[160, 176, 88],  
[158, 174, 86],  
[154, 171, 79],  
...,  
[213, 215, 212],  
[211, 213, 210],  
[209, 211, 208]],  
  
[[161, 176, 93],  
[159, 174, 91],  
[155, 171, 83],  
...,  
[213, 215, 210],  
[211, 213, 208],  
[209, 211, 206]],  
  
[[160, 176, 101],  
[160, 174, 97],  
[157, 172, 91],  
...,  
[212, 214, 209],  
[211, 213, 208],  
[210, 212, 207]],  
  
...,  
  
[[ 33, 34, 29],  
[ 30, 31, 26],  
[ 28, 29, 24],  
...,  
[222, 230, 157],  
[222, 230, 157],  
[222, 230, 157]],  
  
[[ 29, 30, 25],  
[ 29, 30, 25],  
[ 28, 29, 24],  
...,  
[218, 225, 148],  
[218, 225, 148],  
[218, 225, 148]],
```

```
[[ 27,  28,  23],  
 [ 26,  27,  22],  
 [ 23,  24,  19],  
 ...,  
 [213, 220, 140],  
 [213, 220, 140],  
 [213, 220, 140]]], dtype=uint8)]
```

```
In [11]: results_4.show()  
results_2.show()
```



bird 0.43



Video Detections and Classification with the pretrained Yolov5s

```
In [10]: video_path = 'images_general_detection_testing/test_video.mp4'
```

```
In [11]: # video detection
cap = cv2.VideoCapture(video_path)
while cap.isOpened():
    ret, frame = cap.read()

    # Make detections
    results = model(frame)

    cv2.imshow('YOLO', np.squeeze(results.render()))

    # to exit press q
    if cv2.waitKey(10) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```



```
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):
```

Real Time Detections and Classification with the pretrained Yolov5s

In [13]:

```
# real time detection  
cap = cv2.VideoCapture(0)  
while cap.isOpened():  
    ret, frame = cap.read()  
  
    # Make detections
```

```
results = model(frame)

cv2.imshow('YOLO', np.squeeze(results.render()))

# to exit press q
if cv2.waitKey(10) & 0xFF == ord('q'):
    break
cap.release()
cv2.destroyAllWindows()
```



```
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):  
C:\Users\MTechno/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:892: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.  
    with amp.autocast(autocast):
```

Pretrained Model Limitations

The **pretrained YOLOv5 model** we are using is likely **trained** on the **COCO dataset**.

- The model does **not recognize monkeys**.
- The model may also **confuse cats and dogs** because of their visual similarity or not recognize them at all.

Here is the list of the **80 object classes** in the **COCO dataset**:

1. person
2. bicycle
3. car

4. motorcycle
5. airplane
6. bus
7. train
8. truck
9. boat
10. traffic light
11. fire hydrant
12. stop sign
13. parking meter
14. bench
15. bird
16. cat
17. dog
18. horse
19. sheep
20. cow
21. elephant
22. bear
23. zebra
24. giraffe
25. backpack
26. umbrella
27. handbag
28. tie
29. suitcase
30. frisbee
31. skis
32. snowboard
33. sports ball
34. kite
35. baseball bat

- 36. baseball glove
- 37. skateboard
- 38. surfboard
- 39. tennis racket
- 40. bottle
- 41. wine glass
- 42. cup
- 43. fork
- 44. knife
- 45. spoon
- 46. bowl
- 47. banana
- 48. apple
- 49. sandwich
- 50. orange
- 51. broccoli
- 52. carrot
- 53. hot dog
- 54. pizza
- 55. donut
- 56. cake
- 57. chair
- 58. couch
- 59. potted plant
- 60. bed
- 61. dining table
- 62. toilet
- 63. TV
- 64. laptop
- 65. mouse
- 66. remote
- 67. keyboard

- 68. cell phone
- 69. microwave
- 70. oven
- 71. toaster
- 72. sink
- 73. refrigerator
- 74. book
- 75. clock
- 76. vase
- 77. scissors
- 78. teddy bear
- 79. hair dryer
- 80. toothbrush

Solution: Fine-Tune the Pretrained Model