1. git clone <https://github.com/ultralytics/yolov5.git>
2. cd yolov5

Step 3: Set Up the Python Environment

python -m venv <env name>

to activate environment run <env name>\Scripts\activate   
to deactivate environment run -- deactivate  
  
pip install -U pip setuptools wheel

pip install -r requirements.txt  
pip install opencv-python – optional  
run -- python detect.py --source 0

save result run – python detect.py –save resullts  
  
To use an MP4 file for object detection with YOLOv5 in Visual Studio Code, you can follow these steps:

Step 1: Prepare Your MP4 File

Make sure you have the MP4 video file you want to use for detection ready and note its file path.

Step 2: Open YOLOv5 in VS Code

Open VS Code and navigate to your YOLOv5 project directory (where you cloned YOLOv5).

Open a new terminal in VS Code.  
python detect.py --source path/to/your/video.mp4  
  
**Specify a model weight (optional):** If you want to use a specific YOLOv5 model, you can add the --weights argument. For example:  
  
python detect.py --source video.mp4 --weights yolov5s.pt  
python detect.py --source path/to/your/video.mp4 --save-results  
  
to run better  
  
-- python detect.py --source 0 --weights yolov5l.pt  
Set a Higher Confidence Threshold  
-- python detect.py --source 0 --weights yolov5l.pt --conf-thres 0.5  
Adjust the IoU Threshold  
-- python detect.py --source 0 --weights yolov5l.pt --conf-thres 0.5 --iou-thres 0.5

A higher IoU threshold can help in minimizing false positives  
  
  
