

✓ OBJECT DETECTION IN IMAGES

Import Utilities

```
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
!pip install -r requirements.txt
```

[Show hidden output](#)

Upload your Image

```
from google.colab import files
uploaded = files.upload()
```



Choose Files

No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to

Saving random-neonle-at-a-street-intersection-hn1eavrm87i8awwd.ing to random-neonle-at-a-street-intersection-hn1eavrm87i8awwd.ing

Perform object detection


```
import os
from pathlib import Path
```

```
#Get uploaded file name
img_path = list(uploaded.keys())[0]
```

```
#Run yolo5 detection using pre-trained model
!python detect.py --weights yolov5s.pt --img 640 --conf 0.25 --source {img_path}
```



detect: weights=['yolov5s.pt'], source=random-people-at-a-street-intersection-bq1eayrm87i8awwd.jpg, data=data/coco128.yaml, imgsz=[640, 640], device=0, verbose=True

YOLOv5  v7.0-416-gfe1d4d99 Python-3.11.12 torch-2.6.0+cu124 CPU

```

Downloading https://github.com/ultralytics/yolov5/releases/download/v7.0/yolov5s.pt to yolov5s.pt...
100% 14.1M/14.1M [00:00<00:00, 176MB/s]

```

Fusing layers...

YOLOv5s summary: 213 layers, 7225885 parameters, 0 gradients, 16.4 GFLOPs

image 1/1 /content/yolov5/yolov5/random-people-at-a-street-intersection-bq1eayrm87i8awwd.jpg: 448x640 4 persons, 3 cars, 2 trucks, :

Speed: 4.3ms pre-process, 517.9ms inference, 4.7ms NMS per image at shape (1, 3, 640, 640)

```
Results saved to runs/detect/exp
```



VIEW THE RESULT

```
from IPython.display import Image, display
```

```
#Output will be saved in 'run/detect/exp' by default
result_img_path = Path("runs/detect/exp") / img_path
display(Image(filename=result_img_path))
```

