

▼ Install Dependencies

(Remember to choose GPU in Runtime if not already selected. Runtime --> Change Runtime Type --> Hardware accelerator --> GPU)

```
# clone YOLOv5 repository
!git clone https://github.com/ultralytics/yolov5 # clone repo
%cd yolov5
!git reset --hard 064365d8683fd002e9ad789c1e91fa3d021b44f0

Cloning into 'yolov5'...
remote: Enumerating objects: 15637, done.
remote: Counting objects: 100% (244/244), done.
remote: Compressing objects: 100% (144/144), done.
remote: Total 15637 (delta 123), reused 173 (delta 100), pack-reused 15393
Receiving objects: 100% (15637/15637), 14.58 MiB | 27.15 MiB/s, done.
Resolving deltas: 100% (10652/10652), done.
/content/yolov5
HEAD is now at 064365d Update parse_opt() in export.py to work as in train.py (#10789)

# install dependencies as necessary
!pip install -qr requirements.txt # install dependencies (ignore errors)
import torch
from IPython.display import Image, clear_output # to display images
from utils.downloads import attempt_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

===== 184.3/184.3 kB 5.4 MB/s eta 0:00:00
===== 62.7/62.7 kB 5.3 MB/s eta 0:00:00
===== 1.6/1.6 MB 34.8 MB/s eta 0:00:00
Setup complete. Using torch 2.0.0+cu118 _CudaDeviceProperties(name='Tesla T4', major=7, minor=5, total_memory=15101MB, m
```

▼ Download Correctly Formatted Custom Dataset

```
!pip install -q roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="BcWQu6mF00fAS8y412da", model_format="yolov5", notebook="roboflow-yolov5")

===== 56.3/56.3 kB 3.4 MB/s eta 0:00:00
===== 58.8/58.8 kB 6.8 MB/s eta 0:00:00
===== 54.5/54.5 kB 5.4 MB/s eta 0:00:00
Preparing metadata (setup.py) ... done
===== 67.8/67.8 kB 8.3 MB/s eta 0:00:00
Building wheel for wget (setup.py) ... done

!pip install -q roboflow
%cd /content/yolov5
from roboflow import Roboflow
rf = Roboflow(api_key="BcWQu6mF00fAS8y412da")
project = rf.workspace("siewchinyip-outlook-my").project("sixray")
dataset = project.version(4).download("yolov5")

/content/yolov5
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in Sixray-4 to yolov5pytorch: 99% [345726976 / 347156419] bytesExtracting Dataset Versior

# this is the YAML file Roboflow wrote for us that we're loading into this notebook with our data
%cat {dataset.location}/data.yaml

names:
- Gun
- Knife
- Pliers
- Scissors
- Wrench
nc: 5
train: Sixray-4/train/images
val: Sixray-4/valid/images
```

▼ Define Model Configuration and Architecture

We will write a yaml script that defines the parameters for our model like the number of classes, anchors, and each layer.

You do not need to edit these cells, but you may.

```
# 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'])

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

# YOLOv5 🚀 by Ultralytics, GPL-3.0 license

# Parameters
nc: 80 # number of classes
depth_multiple: 0.33 # model depth multiple
width_multiple: 0.50 # layer channel multiple
anchors:
  - [10,13, 16,30, 33,23] # P3/8
  - [30,61, 62,45, 59,119] # P4/16
  - [116,90, 156,198, 373,326] # P5/32

# YOLOv5 v6.0 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [64, 6, 2, 2]], # 0-P1/2
  [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
  [-1, 3, C3, [128]],
  [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
  [-1, 6, C3, [256]],
  [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
  [-1, 9, C3, [512]],
  [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
  [-1, 3, C3, [1024]],
  [-1, 1, SPPF, [1024, 5]], # 9
  ]

# YOLOv5 v6.0 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [[-1, 6], 1, Concat, [1]], # cat backbone P4
  [-1, 3, C3, [512, False]], # 13

  [-1, 1, Conv, [256, 1, 1]],
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [[-1, 4], 1, Concat, [1]], # cat backbone P3
  [-1, 3, C3, [256, False]], # 17 (P3/8-small)

  [-1, 1, Conv, [256, 3, 2]],
  [[-1, 14], 1, Concat, [1]], # cat head P4
  [-1, 3, C3, [512, False]], # 20 (P4/16-medium)

  [-1, 1, Conv, [512, 3, 2]],
  [[-1, 10], 1, Concat, [1]], # cat head P5
  [-1, 3, C3, [1024, False]], # 23 (P5/32-large)

  [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)
  ]

#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()))

%%writetemplate /content/yolov5/models/custom_yolov5s.yaml

# parameters
nc: {num_classes} # number of classes
depth_multiple: 0.33 # model depth multiple
width_multiple: 0.50 # layer channel multiple

# anchors
anchors:
  - [10,13, 16,30, 33,23] # P3/8
  - [30,61, 62,45, 59,119] # P4/16
  - [116,90, 156,198, 373,326] # P5/32

# YOLOv5 backbone
backbone:
```

```
# [from, number, module, args]
[[-1, 1, Focus, [64, 3]], # 0-P1/2
 [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
 [-1, 3, BottleneckCSP, [128]],
 [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
 [-1, 9, BottleneckCSP, [256]],
 [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
 [-1, 9, BottleneckCSP, [512]],
 [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
 [-1, 1, SPP, [1024, [5, 9, 13]]],
 [-1, 3, BottleneckCSP, [1024, False]], # 9
]

# YOLOv5 head
head:
[[-1, 1, Conv, [512, 1, 1]],
 [-1, 1, nn.Upsample, [None, 2, 'nearest']],
 [[-1, 6], 1, Concat, [1]], # cat backbone P4
 [-1, 3, BottleneckCSP, [512, False]], # 13

 [-1, 1, Conv, [256, 1, 1]],
 [-1, 1, nn.Upsample, [None, 2, 'nearest']],
 [[-1, 4], 1, Concat, [1]], # cat backbone P3
 [-1, 3, BottleneckCSP, [256, False]], # 17 (P3/8-small)

 [-1, 1, Conv, [256, 3, 2]],
 [[-1, 14], 1, Concat, [1]], # cat head P4
 [-1, 3, BottleneckCSP, [512, False]], # 20 (P4/16-medium)

 [-1, 1, Conv, [512, 3, 2]],
 [[-1, 10], 1, Concat, [1]], # cat head P5
 [-1, 3, BottleneckCSP, [1024, False]], # 23 (P5/32-large)

 [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)
]
```

▼ Train Custom YOLOv5 Detector

- **img:** define input image size
- **batch:** determine batch size
- **epochs:** define the number of training epochs. (Note: often, 3000+ are common here!)
- **data:** set the path to our yaml file
- **cfg:** specify our model configuration
- **weights:** specify a custom path to weights. (Note: you can download weights from the Ultralytics Google Drive [folder](#))
- **name:** result names
- **nosave:** only save the final checkpoint
- **cache:** cache images for faster training

```
# train yolov5s on custom data for 100 epochs
# time its performance
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 32 --epochs 200 --data {dataset.location}/data.yaml --cfg ./models/custom_yolov5s.yaml --we
```

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	
197/199	4.91G	0.02977	0.01342	0.004308	86	416:	100% 182/182 [00:44<00:00, 4.07it/s]
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100% 26/26 [00:10<00:00, 2.53i
	all	1662	3176	0.91	0.82	0.879	0.578
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	
198/199	4.91G	0.02961	0.01339	0.004337	108	416:	100% 182/182 [00:44<00:00, 4.05it/s]
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100% 26/26 [00:10<00:00, 2.49i
	all	1662	3176	0.911	0.82	0.879	0.578
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	
199/199	4.91G	0.02972	0.01333	0.004362	96	416:	100% 182/182 [00:44<00:00, 4.08it/s]
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100% 26/26 [00:10<00:00, 2.43i
	all	1662	3176	0.908	0.82	0.879	0.579

200 epochs completed in 3.287 hours.

Optimizer stripped from runs/train/yolov5s_results/weights/last.pt, 14.8MB

Optimizer stripped from runs/train/yolov5s_results/weights/best.pt, 14.8MB

Validating runs/train/yolov5s_results/weights/best.pt...

Fusing layers...

custom_YOLOv5s summary: 182 layers, 7257306 parameters, 0 gradients

Class	Images	Instances	P	R	mAP50	mAP50-95: 100% 26/26 [00:14<00:00, 1.78i
all	1662	3176	0.908	0.82	0.879	0.579
Gun	1662	888	0.955	0.944	0.973	0.7
Knife	1662	442	0.927	0.726	0.833	0.521
Pliers	1662	1110	0.895	0.837	0.905	0.599
Scissors	1662	206	0.915	0.801	0.853	0.531
Wrench	1662	530	0.846	0.791	0.829	0.541

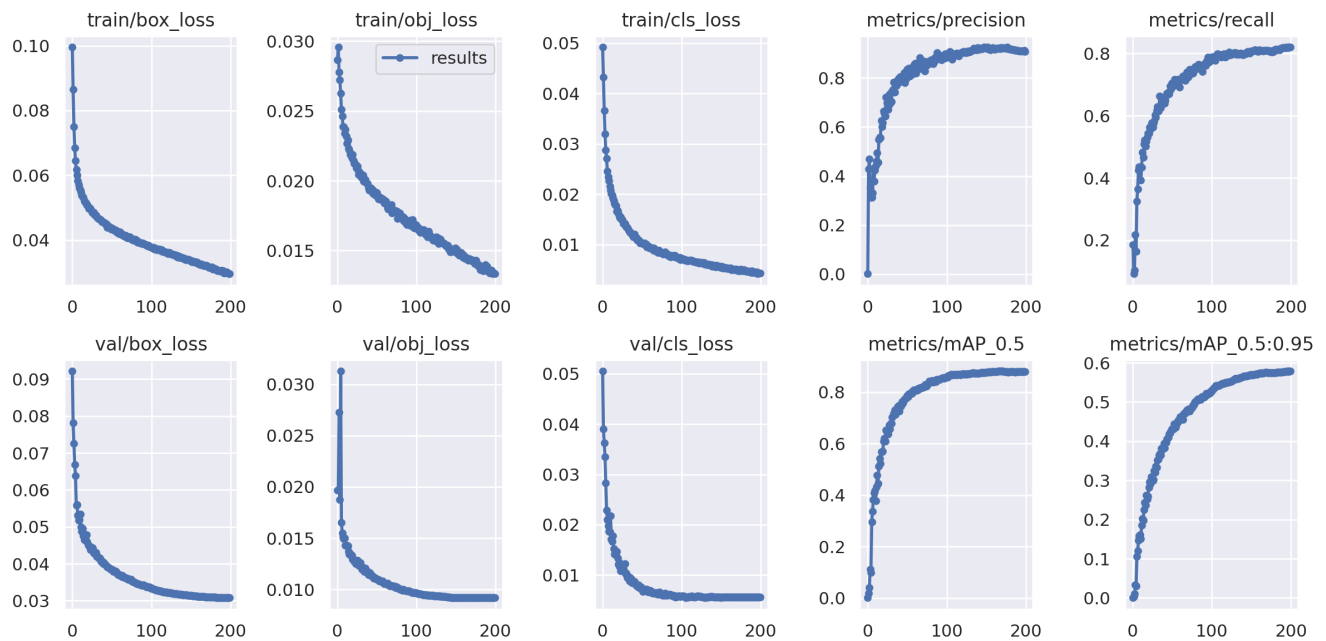
Results saved to runs/train/yolov5s_results

CPU times: user 2min 4s, sys: 14.3 s, total: 2min 18s

Wall time: 3h 18min 53s

▼ Evaluate Custom YOLOv5 Detector Performance

```
#
from utils.plots import plot_results # plot results.txt as results.png
Image(filename='/content/yolov5/runs/train/yolov5s_results/results.png', width=1000) # view results.png
```



▼ Curious? Visualize Our Training Data with Labels

After training starts, view train*.jpg images to see training images, labels and augmentation effects.

```
# first, display our ground truth data
# print("GROUND TRUTH TRAINING DATA:")
# Image(filename='/content/yolov5/runs/train/yolov5s_results/val_batch0_labels.jpg', width=900)

# print out an augmented training example
# print("GROUND TRUTH AUGMENTED TRAINING DATA:")
# Image(filename='/content/yolov5/runs/train/yolov5s_results/train_batch0.jpg', width=900)
```

Run Inference With Trained Weights

Run inference with a pretrained checkpoint on contents of `test/images` folder downloaded from Roboflow.

```
%ls runs/train/yolov5s_results/weights

best.pt  last.pt

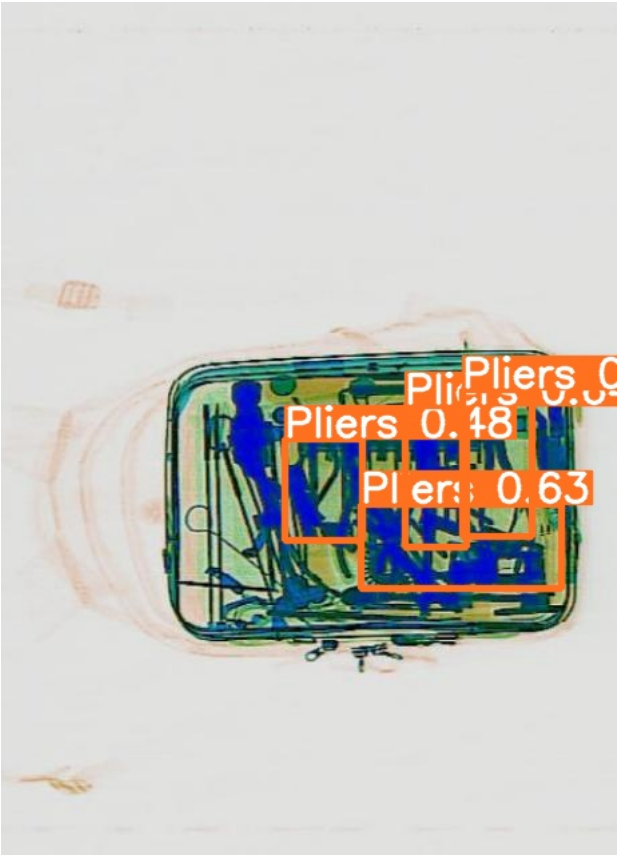
%cd /content/yolov5/
!python detect.py --weights runs/train/yolov5s_results/weights/best.pt --img 416 --conf 0.4 --source /content/yolov5/Sixray-4/

image 174/831 /content/yolov5/Sixray-4/test/images/P01752_jpg.rf.fe502ddec0e2f019b8378d2bef2e930b.jpg: 416x416 1 Gun, 7.8
image 175/831 /content/yolov5/Sixray-4/test/images/P01755_jpg.rf.bed6084ecd1e4670fbb55ed57fa58343.jpg: 416x384 1 Gun, 7.0
image 176/831 /content/yolov5/Sixray-4/test/images/P01761_jpg.rf.61cd010816c9357a2521ca6839e9af48.jpg: 416x384 2 Guns, 6.1
image 177/831 /content/yolov5/Sixray-4/test/images/P01764_jpg.rf.54bfccaca3052b538fba38fef63eceb52.jpg: 320x416 2 Guns, 6.1
image 178/831 /content/yolov5/Sixray-4/test/images/P01771_jpg.rf.fd290c9c8ad49c4ffbb3964d4f37d9d9.jpg: 416x384 4 Guns, 6.1
image 179/831 /content/yolov5/Sixray-4/test/images/P01779_jpg.rf.c1069b39cacbaa9e7ab3e6a36739bd8e.jpg: 288x416 1 Gun, 6.5
image 180/831 /content/yolov5/Sixray-4/test/images/P01786_jpg.rf.ecec7165d0038818ff08eb0a48e25d0e.jpg: 288x416 1 Gun, 5.9
image 181/831 /content/yolov5/Sixray-4/test/images/P01789_jpg.rf.2b60c73868fb34271ecfd165a72d82aa.jpg: 320x416 2 Guns, 6.1
image 182/831 /content/yolov5/Sixray-4/test/images/P01794_jpg.rf.0d3d7fa1803f3317e727f793785dee48.jpg: 416x384 2 Guns, 6.1
image 183/831 /content/yolov5/Sixray-4/test/images/P01802_jpg.rf.0cb438c82c314450c0baae7c8b10abca.jpg: 352x416 2 Guns, 6.1
image 184/831 /content/yolov5/Sixray-4/test/images/P01806_jpg.rf.fe6ef7786959e27432bfdc32a540d0f5.jpg: 416x384 2 Guns, 7.7
image 185/831 /content/yolov5/Sixray-4/test/images/P01809_jpg.rf.9cec5c7f82af3c5285eee7d1729ecb05.jpg: 256x416 2 Guns, 6.1
image 186/831 /content/yolov5/Sixray-4/test/images/P01812_jpg.rf.c95ae7d4d7d8c58626d4caa85bbcc6c50.jpg: 416x320 2 Guns, 6.1
image 187/831 /content/yolov5/Sixray-4/test/images/P01839_jpg.rf.3d74b148e2842d69ea606b831884ab67.jpg: 416x416 1 Gun, 1.1
image 188/831 /content/yolov5/Sixray-4/test/images/P01850_jpg.rf.2632dac8903e50a97606b19885651150.jpg: 288x416 1 Gun, 1.1
image 189/831 /content/yolov5/Sixray-4/test/images/P01851_jpg.rf.687582d03c5e3021e998549f73f558fd.jpg: 384x416 1 Gun, 1.1
image 190/831 /content/yolov5/Sixray-4/test/images/P01888_jpg.rf.ccc303cd83711434bcc939910eb1cfb1.jpg: 416x416 1 Gun, 7.9
image 191/831 /content/yolov5/Sixray-4/test/images/P01891_jpg.rf.96423bf5e51121ba4640c6f03602d70.jpg: 384x416 1 Gun, 7.0
image 192/831 /content/yolov5/Sixray-4/test/images/P01899_jpg.rf.ac749809c8d883f3d1f3ad430ea4a568.jpg: 288x416 1 Gun, 6.1
image 193/831 /content/yolov5/Sixray-4/test/images/P01924_jpg.rf.0700f549543c3ab5d94cfc15e78ab6a.jpg: 384x416 1 Gun, 7.0
image 194/831 /content/yolov5/Sixray-4/test/images/P01929_jpg.rf.1fce12d7750b5a012e237e02063c859a.jpg: 384x416 1 Gun, 6.9
image 195/831 /content/yolov5/Sixray-4/test/images/P01938_jpg.rf.fcd9f1a90017ba5120c0bd01928fd5d7.jpg: 352x416 1 Gun, 6.5
image 196/831 /content/yolov5/Sixray-4/test/images/P01939_jpg.rf.78a93c87f843cd1cc4a1427f0002bd1b.jpg: 288x416 1 Gun, 7.9
image 197/831 /content/yolov5/Sixray-4/test/images/P01946_jpg.rf.625d743bdbd8cf2ab6b6eadaaa015f7e.jpg: 352x416 1 Gun, 6.5
image 198/831 /content/yolov5/Sixray-4/test/images/P01949_jpg.rf.62398debff98b004019e009ed1f47e9a.jpg: 288x416 1 Gun, 6.1
image 199/831 /content/yolov5/Sixray-4/test/images/P01968_jpg.rf.84074a4a1d46712a3611b5e0855362a5.jpg: 352x416 1 Gun, 6.5
image 200/831 /content/yolov5/Sixray-4/test/images/P01969_jpg.rf.24a119692e68a28ecf8e8783387440b5.jpg: 352x416 1 Gun, 6.4
image 201/831 /content/yolov5/Sixray-4/test/images/P01970_jpg.rf.247294df772b454bd820b5c6c8ac7054.jpg: 384x416 6 Guns, 7.7
image 202/831 /content/yolov5/Sixray-4/test/images/P01973_jpg.rf.1ce8f07e8808e1d6fadf1fa24654d757.jpg: 416x416 6 Guns, 9.1
image 203/831 /content/yolov5/Sixray-4/test/images/P01991_jpg.rf.5fe6db72a0a5b541d3a29ce23a42ea5d.jpg: 352x416 2 Guns, 6.1
image 204/831 /content/yolov5/Sixray-4/test/images/P01992_jpg.rf.f426ad88c02ab53bda0a6461517c8de4.jpg: 416x416 2 Guns, 7.7
image 205/831 /content/yolov5/Sixray-4/test/images/P02002_jpg.rf.a8611f4054667d35fe33dbb23bb074d3.jpg: 416x416 2 Guns, 7.1
image 206/831 /content/yolov5/Sixray-4/test/images/P02027_jpg.rf.d107709d94fc65e5fa3c5c22b42c7844.jpg: 416x416 1 Gun, 7.8
image 207/831 /content/yolov5/Sixray-4/test/images/P02041_jpg.rf.3cc240ac8577a780bb64bd3dc12ff00f.jpg: 416x416 1 Gun, 7.8
image 208/831 /content/yolov5/Sixray-4/test/images/P02044_jpg.rf.fdce7814a7a220356c9f651828418ebf.jpg: 352x416 1 Gun, 6.5
image 209/831 /content/yolov5/Sixray-4/test/images/P02052_jpg.rf.b905f7b9879e3dala1ae5dc7dedc598a2.jpg: 416x288 2 Guns, 6.1
image 210/831 /content/yolov5/Sixray-4/test/images/P02060_jpg.rf.dbb6a67ad6c332c7dbcea333e2f8012.jpg: 416x320 2 Guns, 6.1
image 211/831 /content/yolov5/Sixray-4/test/images/P02063_jpg.rf.9904a35d862eb1f6850e2960e5851d8d.jpg: 416x384 1 Gun, 1.1
image 212/831 /content/yolov5/Sixray-4/test/images/P02081_jpg.rf.4d50695751f7f468fab8260e4e624806.jpg: 288x416 2 Guns, 1.1
image 213/831 /content/yolov5/Sixray-4/test/images/P02096_jpg.rf.2c114ef7b953d13c490dd74d9df3b6fb.jpg: 352x416 2 Guns, 1.1
image 214/831 /content/yolov5/Sixray-4/test/images/P02105_jpg.rf.0998752003a30071867947b0d410d1c5.jpg: 416x384 1 Gun, 7.0
image 215/831 /content/yolov5/Sixray-4/test/images/P02107_jpg.rf.938eff8730fd4e23999d00279e359e8f.jpg: 416x416 1 Gun, 7.9
image 216/831 /content/yolov5/Sixray-4/test/images/P02116_jpg.rf.b0d7db47b04b13a66d3b9e1abfb6df34.jpg: 352x416 1 Gun, 6.5
image 217/831 /content/yolov5/Sixray-4/test/images/P02120_jpg.rf.e5d9355db45f06405623342e65effef5.jpg: 288x416 1 Gun, 6.5
image 218/831 /content/yolov5/Sixray-4/test/images/P02121_jpg.rf.7c9ac9afae4931e29c5af01560771719.jpg: 256x416 1 Gun, 6.2
image 219/831 /content/yolov5/Sixray-4/test/images/P02145_jpg.rf.fbf0133027d78c639a9110f67c000a84.jpg: 416x288 2 Guns, 1.1
image 220/831 /content/yolov5/Sixray-4/test/images/P02155_jpg.rf.cd455852da9eac0cc33a3fae98e6155d.jpg: 352x416 2 Guns, 1.1
image 221/831 /content/yolov5/Sixray-4/test/images/P02162_jpg.rf.122083acb9ab94369fb9a70b340f80b2.jpg: 416x288 6 Guns, 2.1
image 222/831 /content/yolov5/Sixray-4/test/images/P02163_jpg.rf.2ccad0516177a07640346f3562bc44d2.jpg: 384x416 5 Guns, 2.1
image 223/831 /content/yolov5/Sixray-4/test/images/P02174_jpg.rf.a9beff0f95aaf7cf81cdd4ecc6c284852.jpg: 416x416 1 Gun, 7.9
image 224/831 /content/yolov5/Sixray-4/test/images/P02176_jpg.rf.63d011577973e622799e1963bed6c8dd.jpg: 416x384 1 Gun, 7.0
image 225/831 /content/yolov5/Sixray-4/test/images/P02193_jpg.rf.517cb7c372566c9d626d9005fb3f0c24.jpg: 416x288 2 Guns, 9.1
image 226/831 /content/yolov5/Sixray-4/test/images/P02202_jpg.rf.7c0467cbd6bd3a8a7b751a1f3776599b.jpg: 416x288 2 Guns, 6.1
image 227/831 /content/yolov5/Sixray-4/test/images/P02209_jpg.rf.fd73d93784db0e77861ce3c81f7c162b.jpg: 416x416 2 Guns, 7.7
image 228/831 /content/yolov5/Sixray-4/test/images/P02230_jpg.rf.42bc0df362e2e8078a6f2ce9627aba92.jpg: 416x416 2 Guns, 7.1
image 229/831 /content/yolov5/Sixray-4/test/images/P02251_jpg.rf.059ec725cecf8661d5faf000b1587b95.jpg: 416x416 2 Guns, 7.1
image 230/831 /content/yolov5/Sixray-4/test/images/P02260_jpg.rf.f14a5b2bade25935b7e7066bb020271d.jpg: 320x416 2 Guns, 6.1
image 231/831 /content/yolov5/Sixray-4/test/images/P02267_jpg.rf.9004f27e3a56f95257407274c39d1fde.jpg: 384x416 1 Gun, 7.0
image 232/831 /content/yolov5/Sixray-4/test/images/P02275_img.rf.11e15040017dc47a79597b1fb1aca8d3_img: 416x416 1 Gun, 7.0

#display inference on ALL test images
#this looks much better with longer training above

import glob
from IPython.display import Image, display

for imageName in glob.glob('/content/yolov5/runs/detect/exp/*'):
    print(display(Image(filename=imageName)))
    print("\n")
    print("Mukund's custom x-ray tested images!")
```



None

Mukund's custom x-ray tested images!



None

Mukund's custom x-ray tested images!

