Piston Detection Using YOLOv8

Let's make sure that we have access to GPU. We can use `nvidia-smi` command to do that. In case of any problems navigate to `Edit` -> `Notebook settings` -> `Hardware accelerator`, set it to `GPU`, and then click `Save`.

```
!nvidia-smi
```

```
Mon Jul  3 06:25:01 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.85.12    Driver Version: 525.85.12    CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   56C    P8    10W /  70W |      0MiB / 15360MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

```
import os
HOME = os.getcwd()
print(HOME)
```

```
/content
```

Install YOLOv8

```
# Pip install method (recommended)

!pip install ultralytics==8.0.20

from IPython import display
display.clear_output()

import ultralytics
ultralytics.checks()
```

```
Ultralytics YOLOv8.0.20 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 24.2/78.2 GB disk)
```

```
from ultralytics import YOLO

from IPython.display import display, Image
```

Inference with Pre-trained COCO Model

💻 CLI

`yolo mode=predict` runs YOLOv8 inference on a variety of sources, downloading models automatically from the latest YOLOv8 release, and saving results to `runs/predict`.

```
%cd {HOME}
!yolo task=detect mode=predict model=yolov8n.pt conf=0.25 source='3.jpg' save=True
```

```
/content
Downloading https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8n.pt to yolov8n.pt...
100% 6.23M/6.23M [00:00<00:00, 58.4MB/s]

2023-07-03 06:27:14.970085: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instruct
To enable the following instructions: AVX2 AVX512F FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-07-03 06:27:15.799035: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
Ultralytics YOLOv8.0.20 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8n summary (fused): 168 layers, 3151904 parameters, 0 gradients, 8.7 GFLOPs
image 1/1 /content/3.jpg: 640x640 1 dining table, 3 clocks, 7.8ms
Speed: 0.6ms pre-process, 7.8ms inference, 322.1ms postprocess per image at shape (1, 3, 640, 640)
Results saved to runs/detect/predict
```

```
%cd {HOME}
Image(filename='runs/detect/predict/3.jpg', height=600)
```

```
/content
```

Roboflow Universe

Roboflow Universe is a repository of more than 110,000 open-source datasets that you can use in your projects. You'll find datasets containing everything from annotated cracks in concrete to plant images with disease annotations.

Preparing a custom dataset

Building a custom dataset can be a painful process. It might take dozens or even hundreds of hours to collect images, label them, and export them in the proper format. Fortunately, Roboflow makes this process as straightforward and fast as possible. Let me show you how!

Step 1: Creating project

Step 2: Uploading images

Step 3: Labeling

Step 4: Generate new dataset version

Step 5: Exporting dataset

Once the dataset version is generated, we have a hosted dataset we can load directly into our notebook for easy training. Click `Export` and select the `YOLO v5 PyTorch` dataset format and finally download the snipped importing the dataset.

```
!mkdir {HOME}/datasets
%cd {HOME}/datasets

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="8PoV0vGFuX4ullVvzlbd")
project = rf.workspace("amrita-vishwa-vidyapeetham-wtgwo").project("piston-detection")
dataset = project.version(1).download("yolov8")
```

```
/content/datasets
Collecting roboflow
  Downloading roboflow-1.1.0-py3-none-any.whl (57 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 57.0/57.0 kB 4.6 MB/s eta 0:00:00
Collecting certifi==2022.12.7 (from roboflow)
  Downloading certifi-2022.12.7-py3-none-any.whl (155 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 155.3/155.3 kB 14.2 MB/s eta 0:00:00
Requirement already satisfied: chardet==4.0.0 in /usr/local/lib/python3.10/dist-packages (from roboflow
Collecting cycler==0.10.0 (from roboflow)
  Downloading cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Collecting idna==2.10 (from roboflow)
  Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.8/58.8 kB 8.0 MB/s eta 0:00:00
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.10/dist-packages (from robof
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from roboflow) (3
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from roboflow)
Requirement already satisfied: opencv-python>=4.1.2 in /usr/local/lib/python3.10/dist-packages (from ro
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from roboflow)
Collecting pyparsing==2.4.7 (from roboflow)
  Downloading pyparsing-2.4.7-py2.py3-none-any.whl (67 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.8/67.8 kB 8.3 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from roboflo
Collecting python-dotenv (from roboflow)
  Downloading python_dotenv-1.0.0-py3-none-any.whl (19 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.2
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.16.0)
Collecting supervision (from roboflow)
  Downloading supervision-0.11.1-py3-none-any.whl (55 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 55.6/55.6 kB 5.6 MB/s eta 0:00:00
Requirement already satisfied: urllib3>=1.26.6 in /usr/local/lib/python3.10/dist-packages (from roboflo
Collecting wget (from roboflow)
  Downloading wget-3.2.zip (10 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.10/dist-packages (from roboflow)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow)
Collecting requests-toolbelt (from roboflow)
  Downloading requests_toolbelt-1.0.0-py2.py3-none-any.whl (54 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 54.5/54.5 kB 7.3 MB/s eta 0:00:00
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplo
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matpl
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplot
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (fr
Building wheels for collected packages: wget
```

Custom Training

```
Installing collected packages: wget, python-dotenv, pyparsing, idna, cycler, certifi, supervision, requ
```

```
%cd {HOME}

!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=25 imgsz=800 plots=True
```

```
/content
Downloading https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8s.pt to yolov8s.pt...
100% 21.5M/21.5M [00:00<00:00, 205MB/s]

Ultralytics YOLOv8.0.20 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
yolo/engine/trainer: task=detect, mode=train, model=yolov8s.yaml, data=/content/datasets/Piston-Detection-1/data.yaml, epochs=25, patience=50,
Downloading https://ultralytics.com/assets/Arial.ttf to /root/.config/Ultralytics/Arial.ttf...
100% 755k/755k [00:00<00:00, 48.1MB/s]
2023-07-03 06:37:01.649992: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instr
To enable the following instructions: AVX2 AVX512F FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-07-03 06:37:02.528221: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
Overriding model.yaml nc=80 with nc=1

                   from  n    params  module                                       arguments
  0                  -1  1       928  ultralytics.nn.modules.Conv                  [3, 32, 3, 2]
  1                  -1  1     18560  ultralytics.nn.modules.Conv                  [32, 64, 3, 2]
  2                  -1  1     29056  ultralytics.nn.modules.C2f                   [64, 64, 1, True]
  3                  -1  1     73984  ultralytics.nn.modules.Conv                  [64, 128, 3, 2]
  4                  -1  2    197632  ultralytics.nn.modules.C2f                   [128, 128, 2, True]
  5                  -1  1    295424  ultralytics.nn.modules.Conv                  [128, 256, 3, 2]
  6                  -1  2    788480  ultralytics.nn.modules.C2f                   [256, 256, 2, True]
  7                  -1  1   1180672  ultralytics.nn.modules.Conv                  [256, 512, 3, 2]
  8                  -1  1   1838080  ultralytics.nn.modules.C2f                   [512, 512, 1, True]
  9                  -1  1    656896  ultralytics.nn.modules.SPPF                  [512, 512, 5]
 10                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 11             [-1, 6]  1         0  ultralytics.nn.modules.Concat                [1]
 12                  -1  1    591360  ultralytics.nn.modules.C2f                   [768, 256, 1]
 13                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 14             [-1, 4]  1         0  ultralytics.nn.modules.Concat                [1]
 15                  -1  1    148224  ultralytics.nn.modules.C2f                   [384, 128, 1]
 16                  -1  1    147712  ultralytics.nn.modules.Conv                  [128, 128, 3, 2]
 17            [-1, 12]  1         0  ultralytics.nn.modules.Concat                [1]
 18                  -1  1    493056  ultralytics.nn.modules.C2f                   [384, 256, 1]
 19                  -1  1    590336  ultralytics.nn.modules.Conv                  [256, 256, 3, 2]
 20             [-1, 9]  1         0  ultralytics.nn.modules.Concat                [1]
 21                  -1  1   1969152  ultralytics.nn.modules.C2f                   [768, 512, 1]
 22        [15, 18, 21]  1   2116435  ultralytics.nn.modules.Detect                [1, [128, 256, 512]]
Model summary: 225 layers, 11135987 parameters, 11135971 gradients, 28.6 GFLOPs

Transferred 349/355 items from pretrained weights
optimizer: SGD(lr=0.01) with parameter groups 57 weight(decay=0.0), 64 weight(decay=0.001), 63 bias
train: Scanning /content/datasets/Piston-Detection-1/train/labels... 156 images, 0 backgrounds, 0 corrupt: 100% 156/156 [00:00<00:00, 1373.10it
train: New cache created: /content/datasets/Piston-Detection-1/train/labels.cache
```

```
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile
val: Scanning /content/datasets/Piston-Detection-1/valid/labels... 10 images, 0 backgrounds, 0 corrupt: 100% 10/10 [00:00<00:00, 1235.14it/s]
val: New cache created: /content/datasets/Piston-Detection-1/valid/labels.cache
Image sizes 800 train, 800 val
Using 2 dataloader workers
Logging results to runs/detect/train
Starting training for 25 epochs...

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       1/25       6.4G      1.613      2.911      1.859         89        800: 100% 10/10 [00:15<00:00,  1.60s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:01<00:00,  1.54s/it]
                   all         10         59      0.386      0.508      0.416      0.377
```

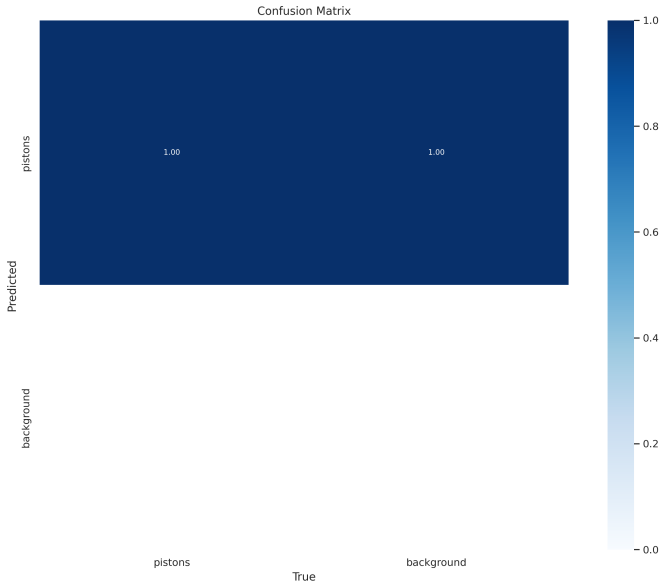
```
!rm -rf *
```

```
!ls {HOME}/runs/detect/train/
```

```
args.yaml                                       train_batch0.jpg
confusion_matrix.png                            train_batch150.jpg
events.out.tfevents.1688366224.fcf5f29d5498.3917.0  train_batch151.jpg
F1_curve.png                                    train_batch152.jpg
P_curve.png                                     train_batch1.jpg
PR_curve.png                                    train_batch2.jpg
R_curve.png                                     val_batch0_labels.jpg
results.csv                                     val_batch0_pred.jpg
results.png                                     weights
```

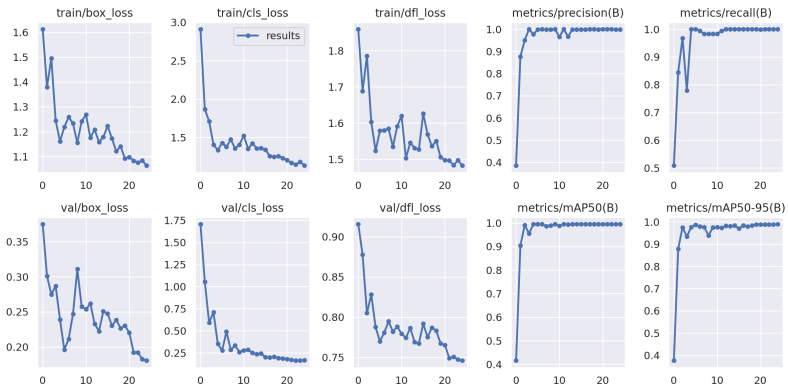
```
%cd {HOME}
Image(filename=f'{HOME}/runs/detect/train/confusion_matrix.png', width=600)
```

```
/content
```



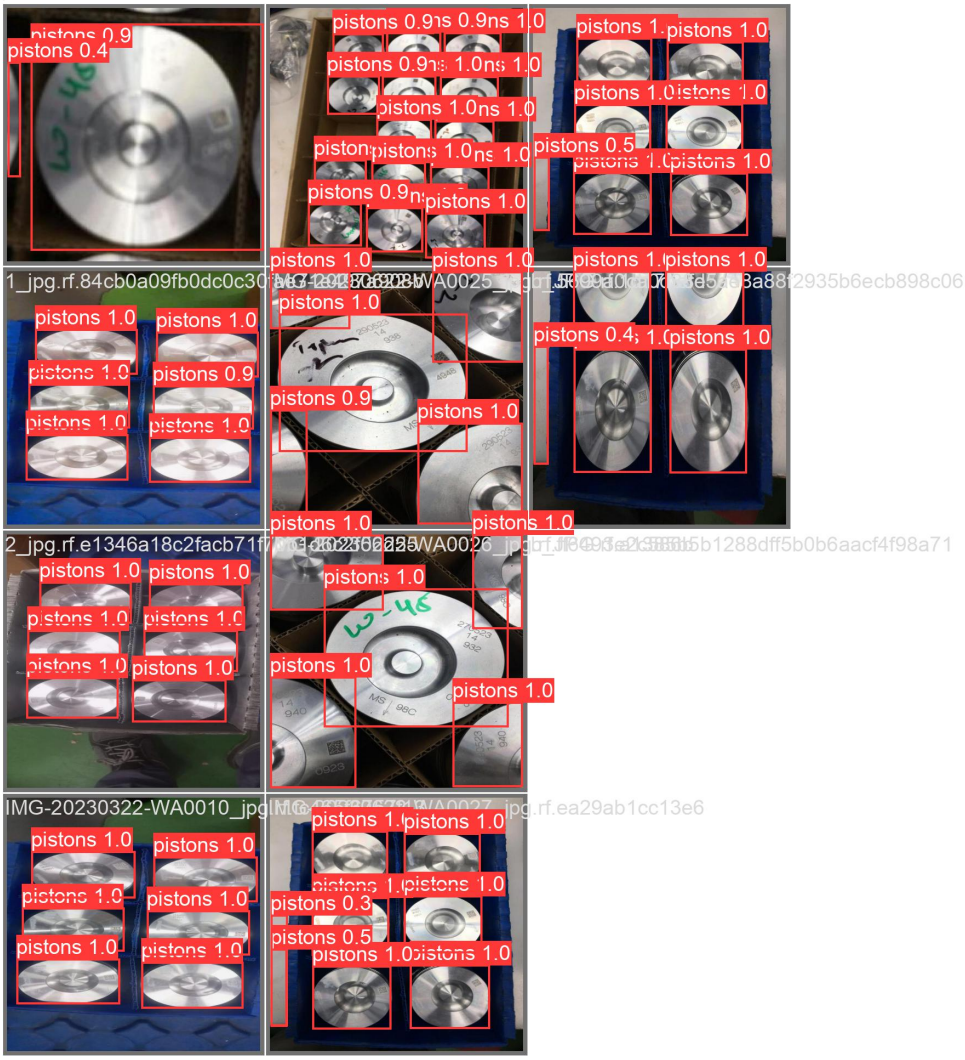

```
%cd {HOME}
Image(filename=f'{HOME}/runs/detect/train/results.png', width=600)
```

```
/content
```




```
%cd {HOME}
Image(filename=f'{HOME}/runs/detect/train/val_batch0_pred.jpg', width=600)
```

/content



Validate Custom Model

```
%cd {HOME}

!yolo task=detect mode=val model={HOME}/runs/detect/train/weights/best.pt data={dataset.location}/data.yaml
```

```
/content
2023-07-03 06:47:49.745427: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instruct
To enable the following instructions: AVX2 AVX512F FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-07-03 06:47:50.597959: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
Ultralytics YOLOv8.0.20 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients, 28.4 GFLOPs
val: Scanning /content/datasets/Piston-Detection-1/valid/labels.cache... 10 images, 0 backgrounds, 0 corrupt: 100% 10/10 [00:00<?, ?it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:00<00:00,  1.18it/s]
                   all         10         59      0.998          1      0.995      0.989
Speed: 0.4ms pre-process, 20.9ms inference, 0.0ms loss, 12.3ms post-process per image
```

Inference with Custom Model

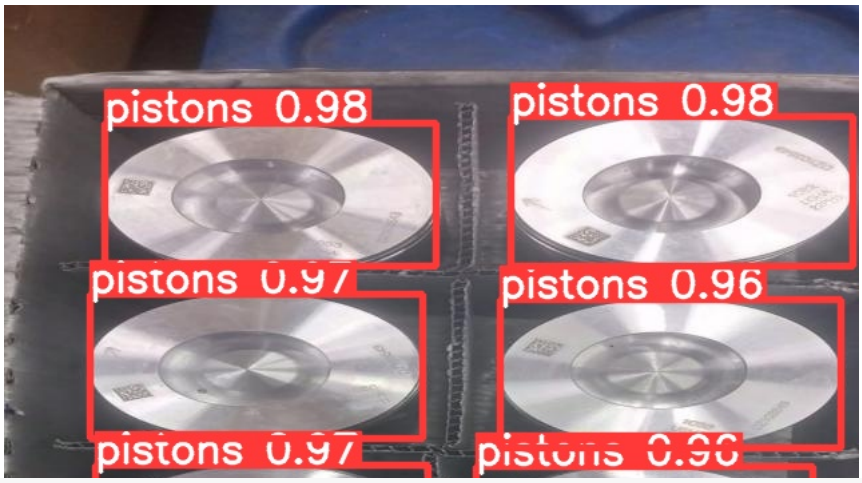
```
%cd {HOME}
!yolo task=detect mode=predict model={HOME}/runs/detect/train/weights/best.pt conf=0.25 source={dataset.location}/test/images save=True
```

```
/content
2023-07-03 06:48:09.467625: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instruct
To enable the following instructions: AVX2 AVX512F FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-07-03 06:48:10.312223: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
Ultralytics YOLOv8.0.20 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients, 28.4 GFLOPs
image 1/4 /content/datasets/Piston-Detection-1/test/images/IMG-20230322-WA0013_jpg.rf.f12e489123121a6b29c5372d24a5e14d.jpg: 800x800 6 pistonss, 24
image 2/4 /content/datasets/Piston-Detection-1/test/images/IMG-20230322-WA0017_jpg.rf.5f8ebe0bb8e25672b09d29d0add8ae74.jpg: 800x800 6 pistonss, 24
image 3/4 /content/datasets/Piston-Detection-1/test/images/IMG-20230622-WA0003_jpg.rf.213a9c3bb85ee3b0c2b2b69e97f1c5b8.jpg: 800x800 3 pistonss, 24
image 4/4 /content/datasets/Piston-Detection-1/test/images/IMG-20230622-WA0005_jpg.rf.7d7b6d55ba2e1cf5f8ee0f6914775cc2.jpg: 800x800 5 pistonss, 24
Speed: 0.7ms pre-process, 24.6ms inference, 22.1ms postprocess per image at shape (1, 3, 800, 800)
Results saved to runs/detect/predict2
```

NOTE: Let's take a look at few results.

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

for image_path in glob.glob(f'{HOME}/runs/detect/predict2/*.jpg')[:3]:
      display(Image(filename=image_path, width=600))
      print("\n")
```




```
model = YOLO(f'{HOME}/yolov8n.pt')
results = model.predict(source=f'{HOME}/runs/detect/predict2/*.jpg', conf=0.25)
```

```
Ultralytics YOLOv8.0.20 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8n summary (fused): 168 layers, 3151904 parameters, 0 gradients, 8.7 GFLOPs
```

```
#Run inference on your model on a persistant, auto-scaling, cloud API

#load model
model = project.version(dataset.version).model

#choose random test set image
import os, random
test_set_loc = dataset.location + "/test/images/"
random_test_image = random.choice(os.listdir(test_set_loc))
print("running inference on " + random_test_image)

pred = model.predict(test_set_loc + random_test_image, confidence=40, overlap=30).json()
pred
```

```
running inference on IMG-20230322-WA0013_jpg.rf.f12e489123121a6b29c5372d24a5e14d.jpg
{'predictions': [{'x': 458.5,
   'y': 430.0,
   'width': 279.0,
   'height': 124.0,
   'confidence': 0.972584068775177,
   'class': 'pistons',
   'image_path': '/content/datasets/Piston-Detection-1/test/images/IMG-20230322-WA0013_jpg.rf.f12e489123121a6b29c5372d24a5e14d.jpg',
   'prediction_type': 'ObjectDetectionModel'},
  {'x': 503.5,
   'y': 144.5,
   'width': 273.0,
   'height': 117.0,
   'confidence': 0.9721883535385132,
```

```
  'class': 'pistons',
  'image_path': '/content/datasets/Piston-Detection-1/test/images/IMG-20230322-WA0013_jpg.rf.f12e489123121a6b29c5372d24a5e14d.jpg',
  'prediction_type': 'ObjectDetectionModel'},
 {'x': 179.0,
  'y': 141.5,
  'width': 274.0,
  'height': 121.0,
  'confidence': 0.9664465188980103,
  'class': 'pistons',
  'image_path': '/content/datasets/Piston-Detection-1/test/images/IMG-20230322-WA0013_jpg.rf.f12e489123121a6b29c5372d24a5e14d.jpg',
  'prediction_type': 'ObjectDetectionModel'},
 {'x': 145.0,
  'y': 424.0,
  'width': 268.0,
  'height': 124.0,
  'confidence': 0.9604313373565674,
  'class': 'pistons',
  'image_path': '/content/datasets/Piston-Detection-1/test/images/IMG-20230322-WA0013_jpg.rf.f12e489123121a6b29c5372d24a5e14d.jpg',
  'prediction_type': 'ObjectDetectionModel'},
 {'x': 492.5,
  'y': 284.5,
  'width': 281.0,
  'height': 121.0,
  'confidence': 0.9573260545730591,
  'class': 'pistons',
  'image_path': '/content/datasets/Piston-Detection-1/test/images/IMG-20230322-WA0013_jpg.rf.f12e489123121a6b29c5372d24a5e14d.jpg',
  'prediction_type': 'ObjectDetectionModel'},
 {'x': 147.0,
  'y': 284.0,
  'width': 272.0,
  'height': 120.0,
  'confidence': 0.9497633576393127,
  'class': 'pistons',
  'image_path': '/content/datasets/Piston-Detection-1/test/images/IMG-20230322-WA0013_jpg.rf.f12e489123121a6b29c5372d24a5e14d.jpg',
  'prediction_type': 'ObjectDetectionModel'}],
'image': {'width': '640', 'height': '640'}}
```