```
   # clone YOLOv5 repository
!git clone https://github.com/ultralytics/yolov5  # clone repo
%cd yolov5
```

```
Cloning into 'yolov5'...
remote: Enumerating objects: 16982, done.
remote: Counting objects: 100% (177/177), done.
remote: Compressing objects: 100% (124/124), done.
remote: Total 16982 (delta 90), reused 109 (delta 53), pack-reused 16805 (from 1)
Receiving objects: 100% (16982/16982), 15.72 MiB | 17.67 MiB/s, done.
Resolving deltas: 100% (11623/11623), done.
/content/yolov5
```

```
# 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() else 'CPU')
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.3/207.3 kB 9.5 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 870.5/870.5 kB 34.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.7/62.7 kB 5.8 MB/s eta 0:00:00
Setup complete. Using torch 2.4.1+cu121 _CudaDeviceProperties(name='Tesla T4', major=7, minor=5, total_memory=15102MB, multi_processor_c
```

```
!pip install -q roboflow
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 80.4/80.4 kB 5.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 66.8/66.8 kB 5.5 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 54.5/54.5 kB 4.4 MB/s eta 0:00:00
```

```
!pip install roboflow
```

```
from roboflow import Roboflow
rf = Roboflow(api_key="Kz0naz6hl8OK2iVgRPyU")
project = rf.workspace("jota-xkeph").project("number-water-meter")
version = project.version(6)
dataset = version.download("yolov7")
```

```
Requirement already satisfied: roboflow in /usr/local/lib/python3.10/dist-packages (1.1.47)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from roboflow) (2024.8.30)
Requirement already satisfied: idna==3.7 in /usr/local/lib/python3.10/dist-packages (from roboflow) (3.7)
Requirement already satisfied: cycler in /usr/local/lib/python3.10/dist-packages (from roboflow) (0.12.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.4.7)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from roboflow) (3.7.1)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.26.4)
Requirement already satisfied: opencv-python-headless==4.10.0.84 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.10.0.84)
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from roboflow) (10.4.0)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.8.2)
Requirement already satisfied: python-dotenv in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.0.1)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.32.3)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.16.0)
Requirement already satisfied: urllib3>=1.26.6 in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.2.3)
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.66.5)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (6.0.2)
Requirement already satisfied: requests-toolbelt in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.0.0)
Requirement already satisfied: filetype in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.2.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (1.3.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (4.54.1)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (24.1)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (3.1.4)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->roboflow) (3.4.0)
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in Number-Water-Meter-6 to yolov7pytorch:: 100%|██████████| 30384/30384 [00:01<00:00, 29300.97it/s]

Extracting Dataset Version Zip to Number-Water-Meter-6 in yolov7pytorch:: 100%|██████████| 1738/1738 [00:00<00:00, 7683.46it/s]
```

```
pwd
```

```
'/content/yolov5'
```

```
dataset.location
```

```
'/content/yolov5/Number-Water-Meter-6'
```

```
%cat
```

```
%cd /content/yolov5
```

```
/content/yolov5
```

```
%cat {dataset.location}/data.yaml
```

```
names:
- '0'
- '10'
- '20'
- '30'
- '40'
- '50'
- '60'
- '70'
- '80'
- '90'
nc: 10
roboflow:
  license: CC BY 4.0
  project: number-water-meter
  url: https://universe.roboflow.com/jota-xkeph/number-water-meter/dataset/6
  version: 6
  workspace: jota-xkeph
test: ../test/images
train: Number-Water-Meter-6/train/images
val: Number-Water-Meter-6/valid/images
```

```
# 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'])


num_classes
```

```
'10'
```

```
%cat /content/yolov5/models/yolov5s.yaml
```

```
# Ultralytics YOLOv5 🚀, AGPL-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]],
```

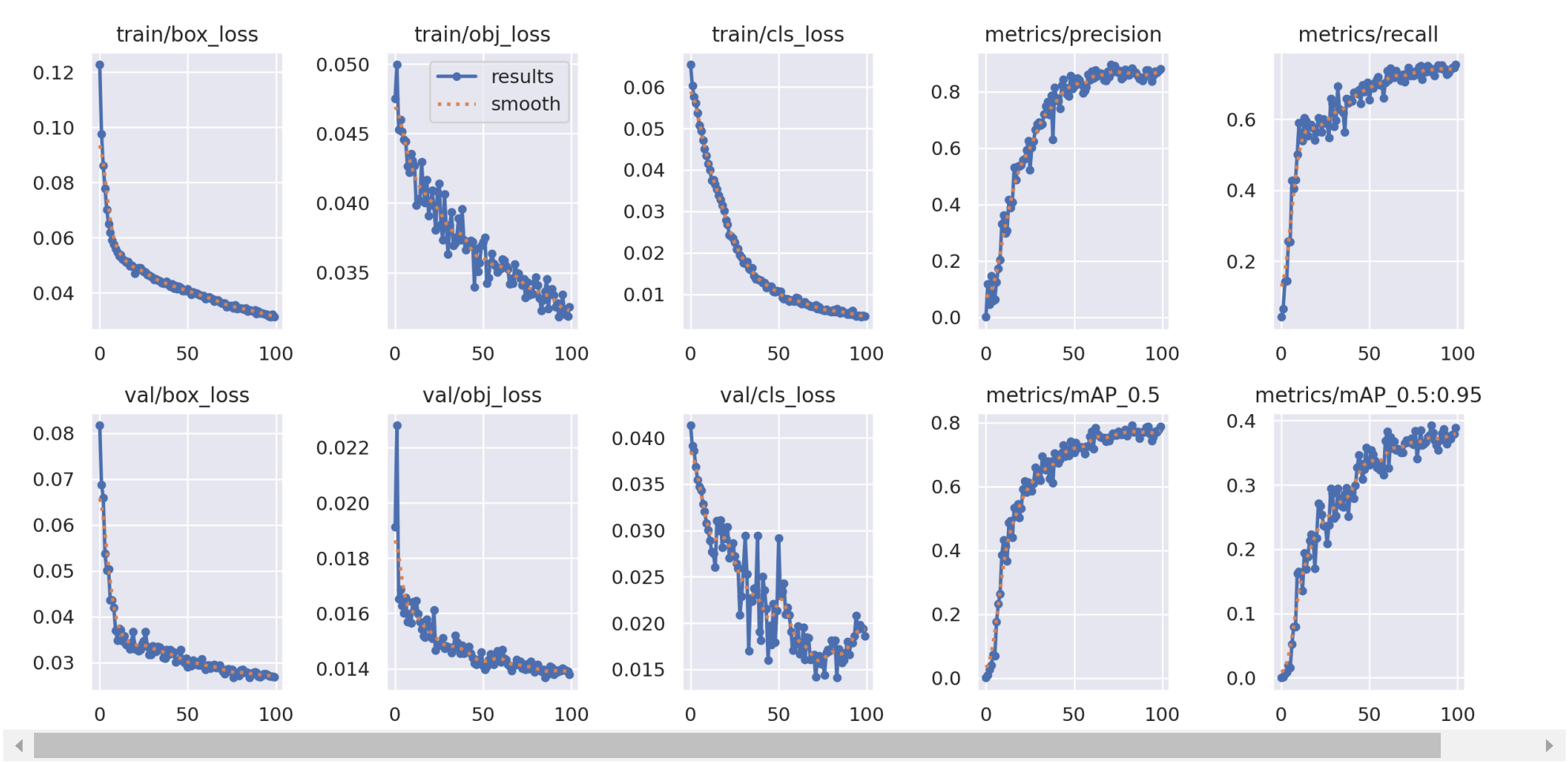
```
            [-1, 1, nn.Upsample, [None, 2, "nearest"]],
            [[-1, 6], 1, Concat, [1]], # cat backbone P4
            [-1, 3, C3, [512, False]], # 13

            [-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()))


    # Ultralytics YOLOv5 🚀, AGPL-3.0 license
    %%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:
      - [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)
  ]
```

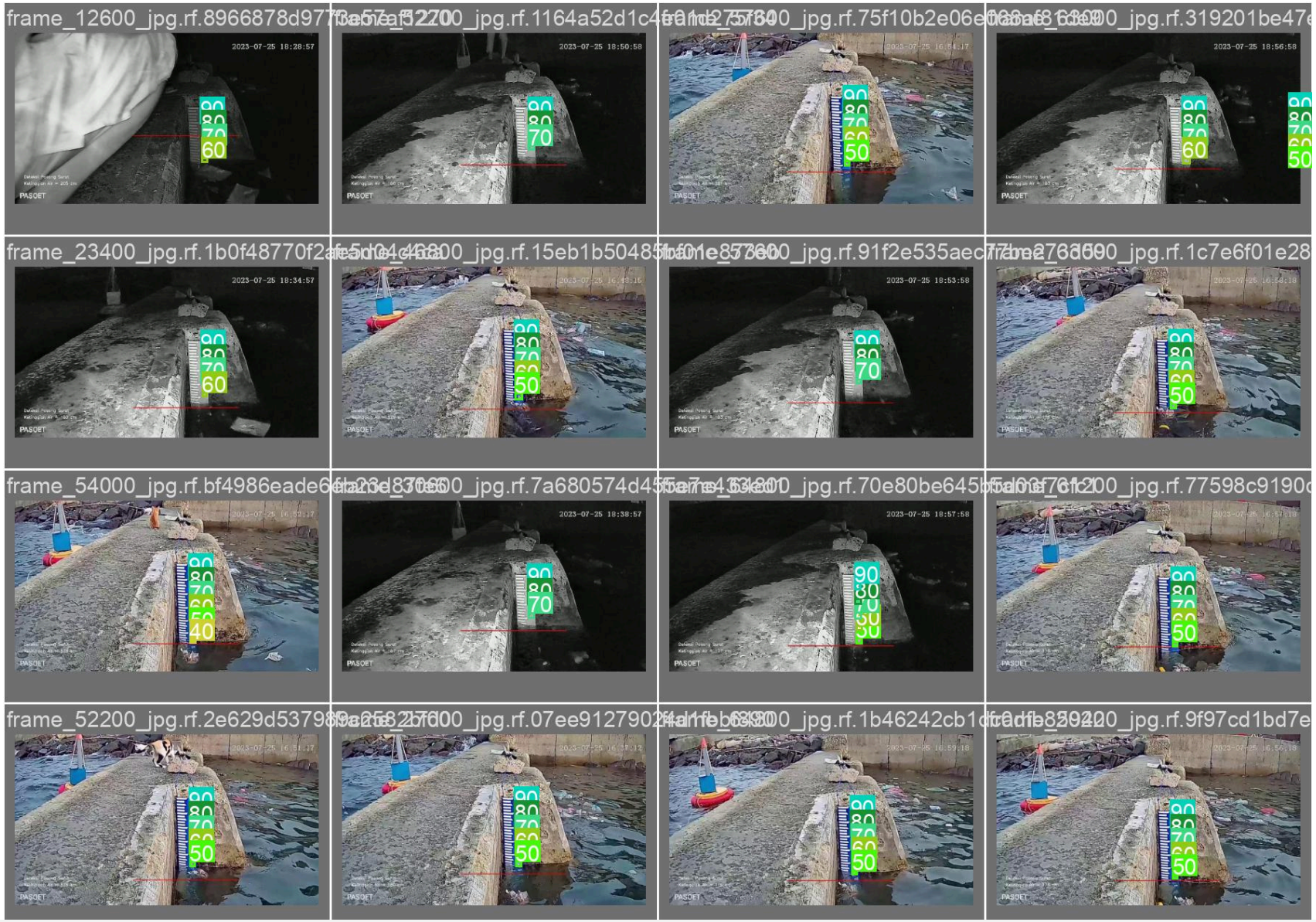
```
# train yolov5s on custom data for 200 epochs
# time its performance
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 16 --epochs 100 --data {dataset.location}/data.yaml --cfg ./models/custom_yolov5s.yaml --weights 'yolov5s.p
```

```
100 epochs completed in 0.312 hours.
Optimizer stripped from runs/train/yolov5s_results2/weights/last.pt, 14.4MB
Optimizer stripped from runs/train/yolov5s_results2/weights/best.pt, 14.4MB

Validating runs/train/yolov5s_results2/weights/best.pt...
Fusing layers...
custom_YOLOv5s summary: 157 layers, 7037095 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 2/2 [00:01<00:00,  1.95it/s]
                   all         55        374      0.868      0.733       0.77      0.392
                     0         55          8      0.727       0.75      0.637      0.223
                    10         55         17      0.657      0.706      0.584      0.293
                    20         55         19       0.94      0.684       0.75      0.356
                    30         55         22      0.887      0.591      0.668      0.319
                    40         55         29      0.772      0.586      0.642      0.341
                    50         55         43      0.934      0.659      0.842      0.412
                    60         55         53      0.892      0.811      0.835      0.494
                    70         55         60      0.947      0.889      0.917       0.52
                    80         55         60      0.967      0.833      0.931      0.479
                    90         55         63      0.957      0.825      0.893      0.487
Results saved to runs/train/yolov5s_results2
CPU times: user 12.9 s, sys: 1.37 s, total: 14.2 s
Wall time: 19min 13s
```

```
from utils.plots import plot_results  # plot results.txt as results.png
Image(filename='/content/yolov5/runs/train/yolov5s_results2/results.png', width=1000)  # view results.png
```

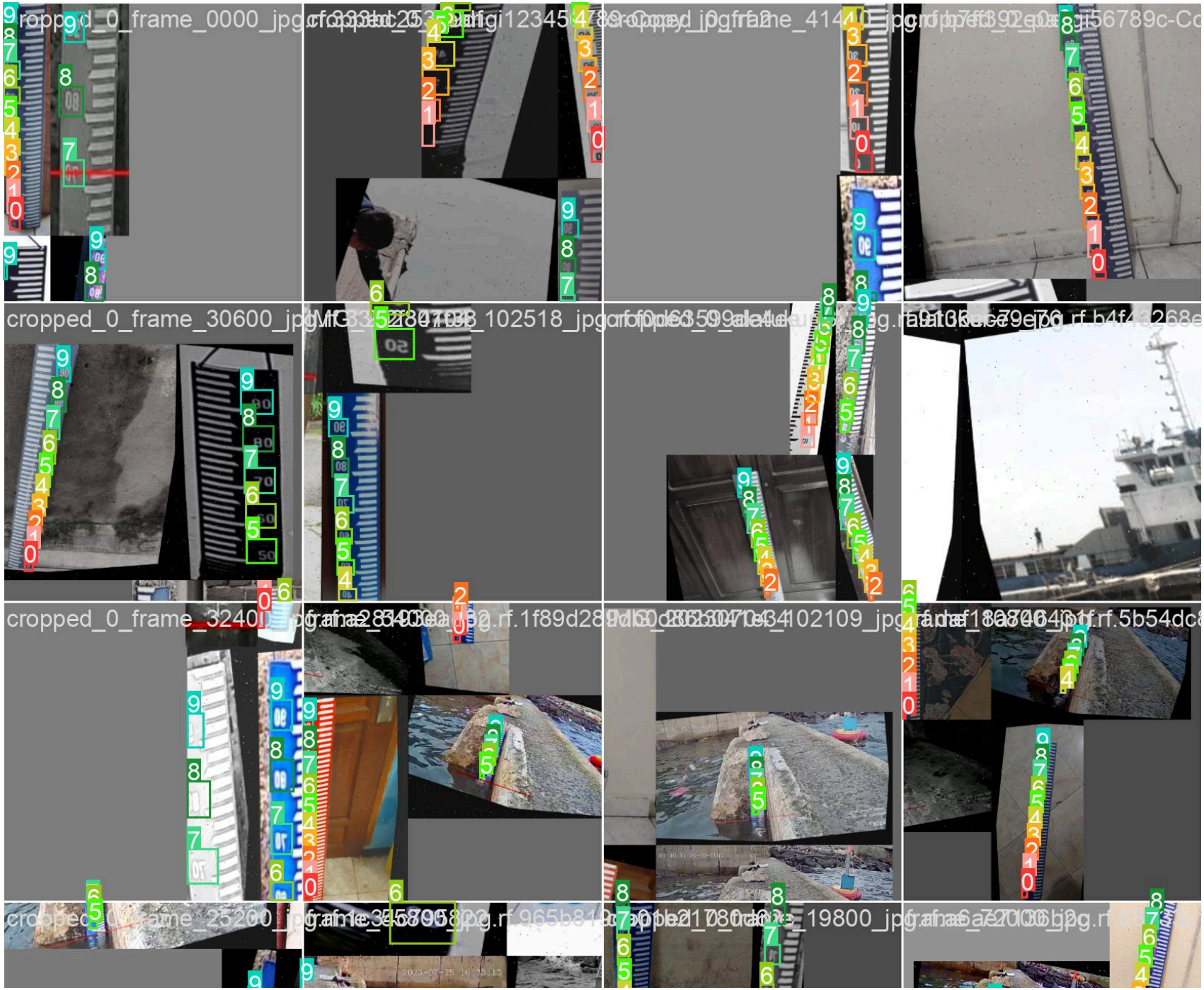


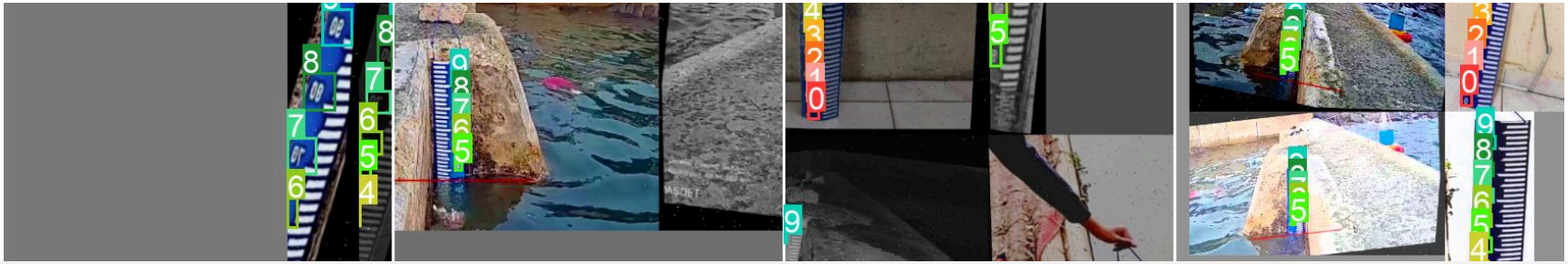

```
Image(filename='/content/yolov5/runs/train/yolov5s_results2/val_batch0_labels.jpg', width=900)
```




```
# print out an augmented training example
print("GROUND TRUTH AUGMENTED TRAINING DATA:")
Image(filename='/content/yolov5/runs/train/yolov5s_results2/train_batch0.jpg', width=900)
```

GROUND TRUTH AUGMENTED TRAINING DATA:





```
%ls runs/
```

```
train/
```

```
%ls runs/train/yolov5s_results2/weights
```

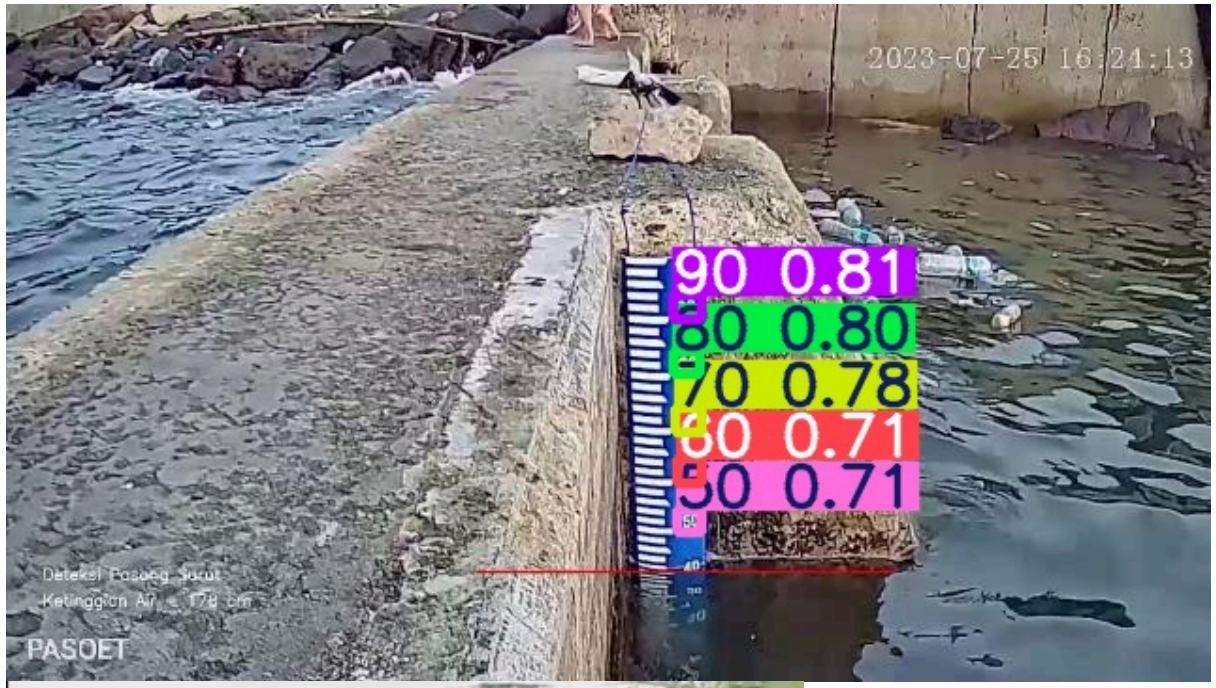
```
best.pt  last.pt
```

```
%cd /content/yolov5/
!python detect.py --weights runs/train/yolov5s_results2/weights/best.pt --img 416 --conf 0.4 --source /content/yolov5/Number-Water-Meter-6/tes
```

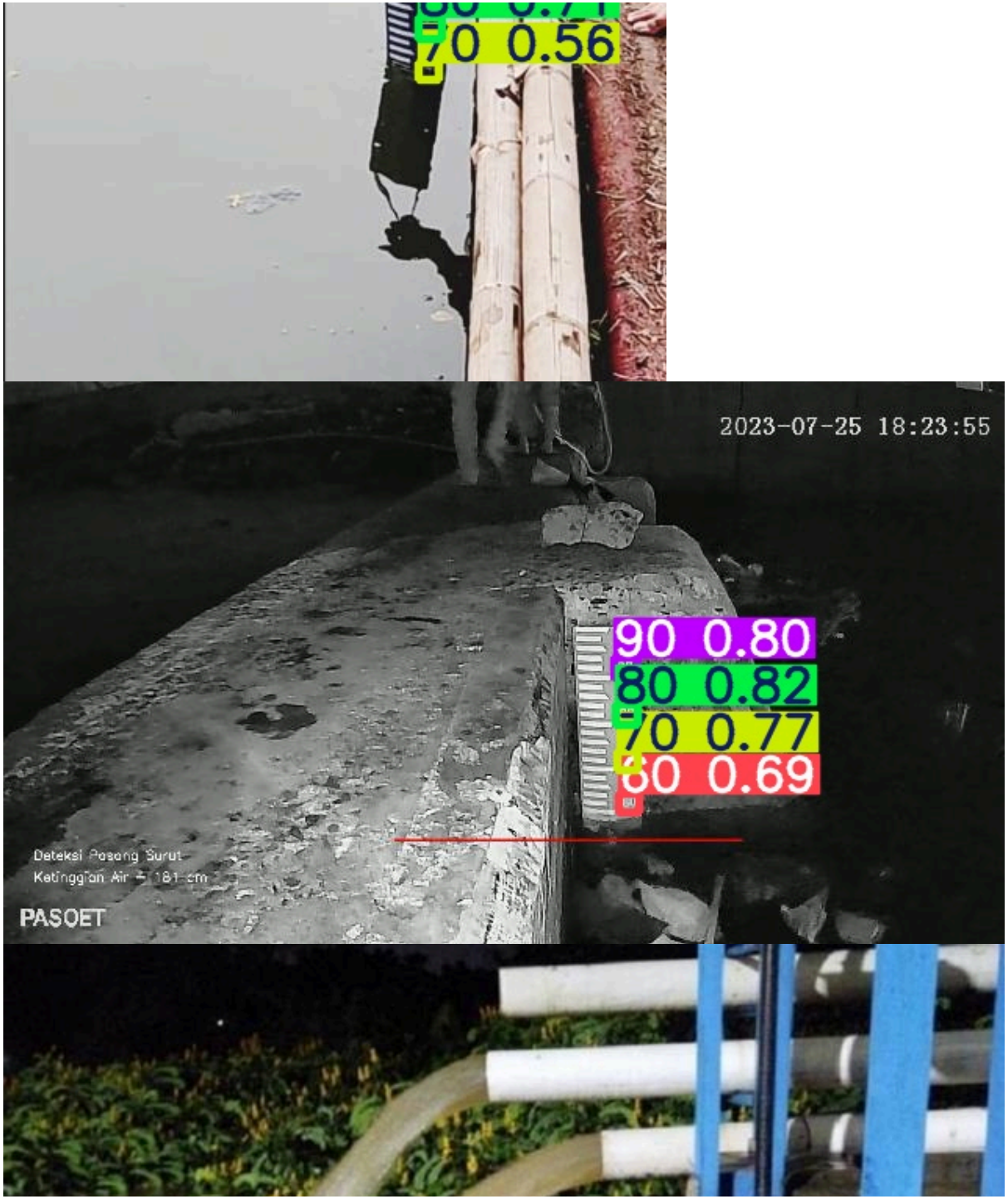
```
/content/yolov5
detect: weights=['runs/train/yolov5s_results2/weights/best.pt'], source=/content/yolov5/Number-Water-Meter-6/test/images, data=data/coco
YOLOv5 🚀 v7.0-372-ga3555241 Python-3.10.12 torch-2.4.1+cu121 CUDA:0 (Tesla T4, 15102MiB)

Fusing layers...
custom_YOLOv5s summary: 157 layers, 7037095 parameters, 0 gradients, 15.8 GFLOPs
image 1/22 /content/yolov5/Number-Water-Meter-6/test/images/4parigi456789-Copy_jpg.rf.5a9fdd551a363b5475aedc1fe5fa29ce.jpg: 416x288 1 40
image 2/22 /content/yolov5/Number-Water-Meter-6/test/images/5parigi123456789-Copy_jpg.rf.8d411f032322313ef509e338555ac34b.jpg: 416x288 1
image 3/22 /content/yolov5/Number-Water-Meter-6/test/images/baru-3_jpg.rf.65ffffdcf8cbcd726e22f180f8334134.jpg: 416x416 1 10, 1 20, 1 30
image 4/22 /content/yolov5/Number-Water-Meter-6/test/images/frame_0000_jpg.rf.35d09ea39d2554c31a19e17a0886e730.jpg: 256x416 1 50, 1 60,
image 5/22 /content/yolov5/Number-Water-Meter-6/test/images/frame_14400_jpg.rf.1ffeb18b9d433aaf09ed4efcbe66d84f.jpg: 256x416 1 60, 1 70,
image 6/22 /content/yolov5/Number-Water-Meter-6/test/images/frame_16200_jpg.rf.231d09569c2118cd1b15439f897619e7.jpg: 256x416 1 40, 1 50,
image 7/22 /content/yolov5/Number-Water-Meter-6/test/images/frame_1800_jpg.rf.effeb0dff192716492fa276fd4b05d09.jpg: 256x416 1 60, 1 70,
image 8/22 /content/yolov5/Number-Water-Meter-6/test/images/frame_21600_jpg.rf.a2fb682a0d23bec73313468b653302f8.jpg: 256x416 1 60, 1 70,
image 9/22 /content/yolov5/Number-Water-Meter-6/test/images/frame_25200_jpg.rf.a6a469471e3d1d677dce117051c9b071.jpg: 256x416 1 60, 1 70,
image 10/22 /content/yolov5/Number-Water-Meter-6/test/images/frame_28800_jpg.rf.4aed8a34abced7af7d0639935eec8793.jpg: 256x416 1 60, 1 70
```

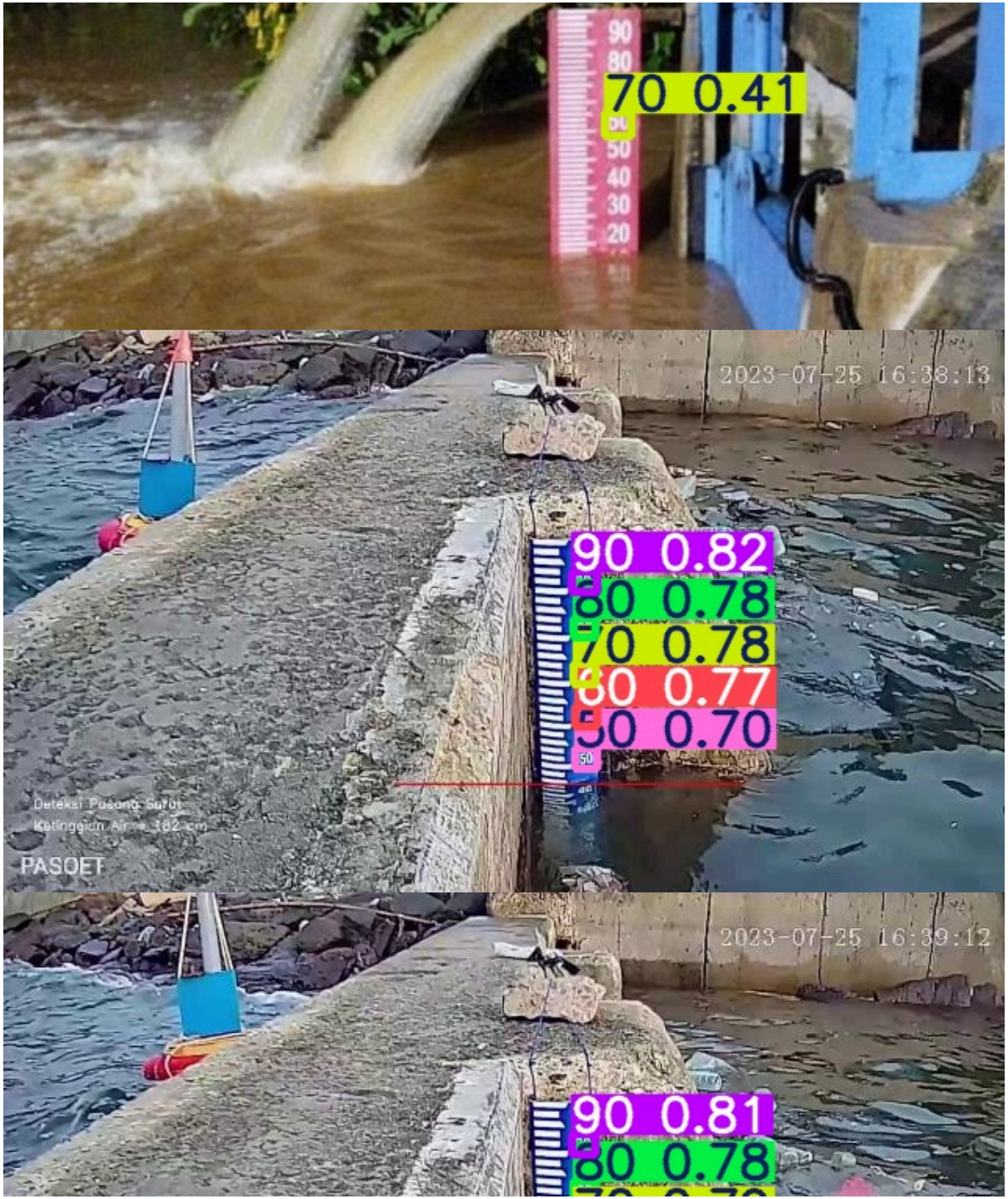
```
image 11/22 /content/yolov5/Number-Water-Meter-6/test/images/frame_28800_jpg.rf.b26906946d5f40aa448fed2a59fab099.jpg: 256x416 1 50, 1 60
image 12/22 /content/yolov5/Number-Water-Meter-6/test/images/frame_30600_jpg.rf.b50c4807a983ef576e88f2d501ee06d9.jpg: 256x416 1 50, 1 60
image 13/22 /content/yolov5/Number-Water-Meter-6/test/images/frame_32400_jpg.rf.4ce703876e2e2f058dc0056f658941bd.jpg: 256x416 1 60, 1 70
image 14/22 /content/yolov5/Number-Water-Meter-6/test/images/frame_3600_jpg.rf.51567bf4980931bce8488d726a47d454.jpg: 256x416 1 60, 1 70,
image 15/22 /content/yolov5/Number-Water-Meter-6/test/images/frame_3600_jpg.rf.af91ddeeb6cb92648101a68b332e1ff4.jpg: 256x416 1 50, 1 60,
image 16/22 /content/yolov5/Number-Water-Meter-6/test/images/frame_39600_jpg.rf.5bea214feb14cafbbd659e733030958b.jpg: 256x416 1 70, 1 80
image 17/22 /content/yolov5/Number-Water-Meter-6/test/images/frame_41400_jpg.rf.8f4852c15b2af2741adba0fcb3b84717.jpg: 256x416 1 60, 1 70
image 18/22 /content/yolov5/Number-Water-Meter-6/test/images/parigi6789_jpg.rf.62d8d83d0b10d346360a18b6d762f85e.jpg: 416x288 1 60, 1 70,
image 19/22 /content/yolov5/Number-Water-Meter-6/test/images/parigi6789c-Copy_jpg.rf.351648c0b1d6f64e27391c3c0d0eebfc.jpg: 416x224 1 60,
image 20/22 /content/yolov5/Number-Water-Meter-6/test/images/parigi89-Copy_jpg.rf.83c319aaff5e8176d9db64dab2d38890.jpg: 416x288 1 70, 1
image 21/22 /content/yolov5/Number-Water-Meter-6/test/images/pengukur-136-_jpg.rf.9df033b3f6ce92c7be457066e9b2959a.jpg: 288x416 1 20, 1
image 22/22 /content/yolov5/Number-Water-Meter-6/test/images/pengukur-144-_jpg.rf.a2028077014884872851 7a9822a31de1.jpg: 256x416 1 70, 5.
Speed: 0.2ms pre-process, 9.5ms inference, 23.5ms NMS per image at shape (1, 3, 416, 416)
Results saved to runs/detect/exp
```

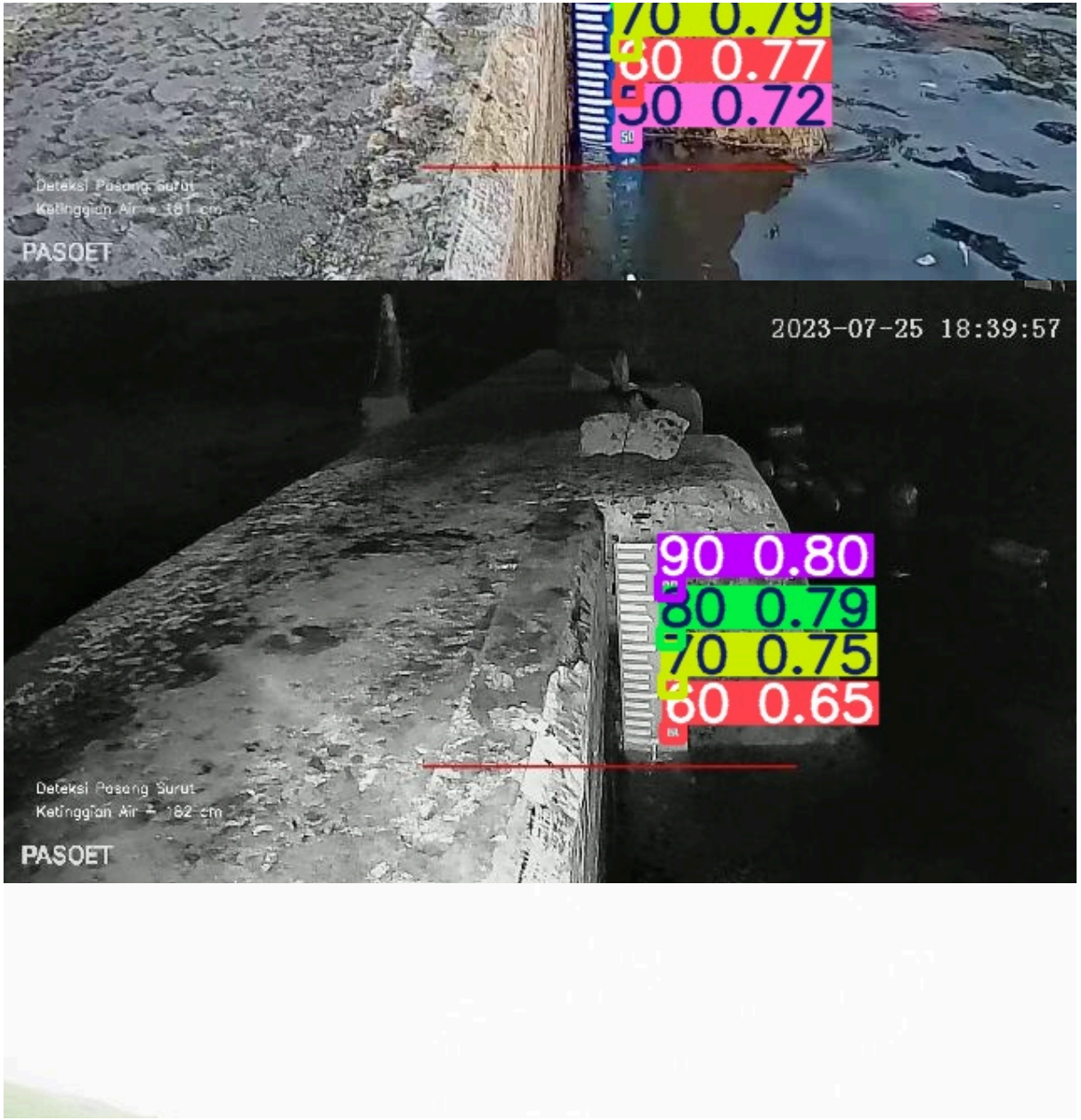
```
import glob
from IPython.display import Image, display

for imageName in glob.glob('/content/yolov5/runs/detect/exp/*.jpg')[:10]: #assuming JPG
    display(Image(filename=imageName))
```


2023-07-25 16:24:13
90 0.81
80 0.80
70 0.78
60 0.71
50 0.71
PASOET


90 0.74


70 0.56
2023-07-25 18:23:55
90 0.80
80 0.82
70 0.77
60 0.69
Deteksi Pasang Surut
Ketinggian Air = 181 cm
PASOET


70 0.41
2023-07-25 16:38:13
90 0.82
80 0.78
70 0.78
60 0.77
50 0.70
Deteksi Pusong Surut
PASOET
2023-07-25 16:39:12
90 0.81
80 0.78


70 0.79
80 0.77
50 0.72
Deteksi Pasang Surut
PASOET
2023-07-25 18:39:57
90 0.80
80 0.79
70 0.75
60 0.65
Deteksi Pasang Surut
PASOET