Im following this tutorial: https://learnopencv.com/fine-tuning-yolov7-on-custom-dataset/

```
import os
import torch

if torch.cuda.is_available():
    print("CUDA is available! 🎉")
    print(f"CUDA version: {torch.version.cuda}")
    print(f"Number of GPUs: {torch.cuda.device_count()}")
else:
    print("CUDA is not available. 😔")

!nvidia-smi
```

```
CUDA is available! 🎉
CUDA version: 12.1
Number of GPUs: 1
Sat Jan 25 09:45:48 2025
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| 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   34C    P8              10W /  70W |      3MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

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

```
# Download the dataset. ifnot already there.
if not os.path.exists('pothole_dataset.zip'):
    !wget https://learnopencv.s3.us-west-2.amazonaws.com/pothole_dataset.zip

```

Download dataset, extract code and view direcotry tree

```
# Extract the dataset.
if not os.path.exists('pothole_dataset'):
    !unzip -q pothole_dataset.zip
```

```
!ls -R pothole_dataset
```

Show hidden output

We need the above output of the direcotry tree to look something like this.

NOTE TO SELF: Check later if this is an actual requirement of training a YOLOv7 model.

```
pothole_dataset/
├── images
│   ├── test [118 entries exceeds filelimit, not opening dir]
│   ├── train [1265 entries exceeds filelimit, not opening dir]
│   └── valid [401 entries exceeds filelimit, not opening dir]
└── labels
    ├── test [118 entries exceeds filelimit, not opening dir]
    ├── train [1265 entries exceeds filelimit, not opening dir]
    ├── valid [401 entries exceeds filelimit, not opening dir]
```

Below are some interesting yaps about how the boundary boxes are labeled along with the images.

Each image comes along with a text file that contains the annotations of the boundary boxes for that image.

```
0 0.5497282608695652 0.5119565217391304 0.017934782608695653 0.005072463768115942
0 0.41032608695652173 0.5253623188405797 0.025 0.005797101449275362
```

```
0 0.30842391304347827 0.5282608695652173 0.014673913043478261 0.005797101449275362
0 0.1654891304347826 0.5224637681159421 0.027717391304347826 0.005797101449275362
0 0.10163043478260869 0.5286231884057971 0.01956521739130435 0.006521739130434782
0 0.07907608695652174 0.5293478260869565 0.01576086956521739 0.007971014492753623
```

- First number: Class (category)
- Next four floats: <x_center, y_center, width, height>

All coordinates are noramlised to the image.

CLONE YOLO!!!

```
if not os.path.exists('yolov7'):
    !git clone https://github.com/WongKinYiu/yolov7.git

%cd yolov7

!pip install -r requirements.txt
```

Show hidden output

Clone if not already there then cd into it and install requirements.

FEED THE CLONE YALM!!!

Fuck the fluff

To train YOLO models you need configured .yalm files.

lets see how thats done

> "This .yaml file contains the paths to the image sets, the number of classes, and the name of the classes."

```
%%writefile data/pothole.yaml
train: ../pothole_dataset/images/train
val: ../pothole_dataset/images/valid
test: ../pothole_dataset/images/test

# Classes
nc: 1  # number of classes
names: ['pothole']  # class names
```

```
Writing data/pothole.yaml
```

Remember this is still in the yolov7 directory.

Right now i cant see it tho...hmm weird, strange behaviour...

Lets start small: Tiny YOLOv7 Fixed Resolution Training

OOhhhh 6 million parameters, ahhh wooowww so impressive...... idk, actually? is it?

native base resolution: 640x640

1. First get the model:

```
# Download the Tiny model weights.
!wget https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7-tiny.pt
```

Show hidden output

This has been pre-trained on the COCO dataset. Ill link that later on... if i remember.

YOOO, little side note, i found the crab mode feature under the misc settings. and this combo feature is fiiiiirrrrreeeeeee my high score right now is like 250.

fun

2. Configuration: they are vague on what were configuring but config files for the model can be found here: yolov7/cfg/training/

The high score is now 600...

Anyway, we need to change the configuration file current present there with the below code. Actually, just for comparison, here is the current code:

```
# parameters
nc: 80  # number of classes
depth_multiple: 1.0  # model depth multiple
width_multiple: 1.0  # 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

# yolov7-tiny backbone
backbone:
  # [from, number, module, args] c2, k=1, s=1, p=None, g=1, act=True
  [[-1, 1, Conv, [32, 3, 2, None, 1, nn.LeakyReLU(0.1)]],  # 0-P1/2

   [-1, 1, Conv, [64, 3, 2, None, 1, nn.LeakyReLU(0.1)]],  # 1-P2/4

   [-1, 1, Conv, [32, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-2, 1, Conv, [32, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [32, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [32, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [[-1, -2, -3, -4], 1, Concat, [1]],
   [-1, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 7

   [-1, 1, MP, []],  # 8-P3/8
   [-1, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-2, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [64, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [64, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [[-1, -2, -3, -4], 1, Concat, [1]],
   [-1, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 14

   [-1, 1, MP, []],  # 15-P4/16
   [-1, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-2, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [128, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [128, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [[-1, -2, -3, -4], 1, Concat, [1]],
   [-1, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 21

   [-1, 1, MP, []],  # 22-P5/32
   [-1, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-2, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [256, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [256, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [[-1, -2, -3, -4], 1, Concat, [1]],
   [-1, 1, Conv, [512, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 28
  ]

# yolov7-tiny head
head:
  [[-1, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-2, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, SP, [5]],
   [-2, 1, SP, [9]],
   [-3, 1, SP, [13]],
   [[-1, -2, -3, -4], 1, Concat, [1]],
   [-1, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [[-1, -7], 1, Concat, [1]],
   [-1, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 37

   [-1, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [21, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]], # route backbone P4
   [[-1, -2], 1, Concat, [1]],

   [-1, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-2, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [64, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [64, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [[-1, -2, -3, -4], 1, Concat, [1]],
   [-1, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 47

   [-1, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [14, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]], # route backbone P3
   [[-1, -2], 1, Concat, [1]],

   [-1, 1, Conv, [32, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-2, 1, Conv, [32, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [32, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [32, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [[-1, -2, -3, -4], 1, Concat, [1]],
   [-1, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 57

   [-1, 1, Conv, [128, 3, 2, None, 1, nn.LeakyReLU(0.1)]],
   [[-1, 47], 1, Concat, [1]],

   [-1, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-2, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [64, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [64, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [[-1, -2, -3, -4], 1, Concat, [1]],
   [-1, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 65

   [-1, 1, Conv, [256, 3, 2, None, 1, nn.LeakyReLU(0.1)]],
   [[-1, 37], 1, Concat, [1]],

   [-1, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-2, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [128, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [-1, 1, Conv, [128, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [[-1, -2, -3, -4], 1, Concat, [1]],
   [-1, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 73

   [57, 1, Conv, [128, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [65, 1, Conv, [256, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
   [73, 1, Conv, [512, 3, 1, None, 1, nn.LeakyReLU(0.1)]],

   [[74,75,76], 1, IDetect, [nc, anchors]],   # Detect(P3, P4, P5)
  ]
```

```
%%writefile cfg/training/yolov7_pothole-tiny.yaml
# parameters
nc: 1  # number of classes
depth_multiple: 1.0  # model depth multiple
width_multiple: 1.0  # 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

# yolov7-tiny backbone
backbone:
  # [from, number, module, args] c2, k=1, s=1, p=None, g=1, act=True
  [[-1, 1, Conv, [32, 3, 2, None, 1, nn.LeakyReLU(0.1)]],  # 0-P1/2

  [-1, 1, Conv, [64, 3, 2, None, 1, nn.LeakyReLU(0.1)]],  # 1-P2/4

  [-1, 1, Conv, [32, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-2, 1, Conv, [32, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [32, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [32, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [[-1, -2, -3, -4], 1, Concat, [1]],
  [-1, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 7

  [-1, 1, MP, []],  # 8-P3/8
  [-1, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-2, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [64, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [64, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [[-1, -2, -3, -4], 1, Concat, [1]],
  [-1, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 14

  [-1, 1, MP, []],  # 15-P4/16
  [-1, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-2, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [128, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [128, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [[-1, -2, -3, -4], 1, Concat, [1]],
  [-1, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 21

  [-1, 1, MP, []],  # 22-P5/32
  [-1, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-2, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [256, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [256, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [[-1, -2, -3, -4], 1, Concat, [1]],
  [-1, 1, Conv, [512, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 28
  ]

# yolov7-tiny head
head:
  [[-1, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-2, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, SP, [5]],
  [-2, 1, SP, [9]],
  [-3, 1, SP, [13]],
  [[-1, -2, -3, -4], 1, Concat, [1]],
  [-1, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [[-1, -7], 1, Concat, [1]],
  [-1, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 37

  [-1, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [21, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]], # route backbone P4
  [[-1, -2], 1, Concat, [1]],

  [-1, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-2, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [64, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [64, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [[-1, -2, -3, -4], 1, Concat, [1]],
  [-1, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 47

  [-1, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [14, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]], # route backbone P3
  [[-1, -2], 1, Concat, [1]],

  [-1, 1, Conv, [32, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-2, 1, Conv, [32, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [32, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [32, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [[-1, -2, -3, -4], 1, Concat, [1]],
  [-1, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 57

  [-1, 1, Conv, [128, 3, 2, None, 1, nn.LeakyReLU(0.1)]],
  [[-1, 47], 1, Concat, [1]],

  [-1, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-2, 1, Conv, [64, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [64, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [64, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [[-1, -2, -3, -4], 1, Concat, [1]],
  [-1, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 65

  [-1, 1, Conv, [256, 3, 2, None, 1, nn.LeakyReLU(0.1)]],
  [[-1, 37], 1, Concat, [1]],

  [-1, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-2, 1, Conv, [128, 1, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [128, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
  [-1, 1, Conv, [128, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
[[-1, -2, -3, -4], 1, Concat, [1]],
[-1, 1, Conv, [256, 1, 1, None, 1, nn.LeakyReLU(0.1)]],  # 73

[57, 1, Conv, [128, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
[65, 1, Conv, [256, 3, 1, None, 1, nn.LeakyReLU(0.1)]],
[73, 1, Conv, [512, 3, 1, None, 1, nn.LeakyReLU(0.1)]],

[[74,75,76], 1, IDetect, [nc, anchors]],   # Detect(P3, P4, P5)
]
```

```
Writing cfg/training/yolov7_pothole-tiny.yaml
```

Alright, here we have some model architecture, this is the good stuff. NICCEE

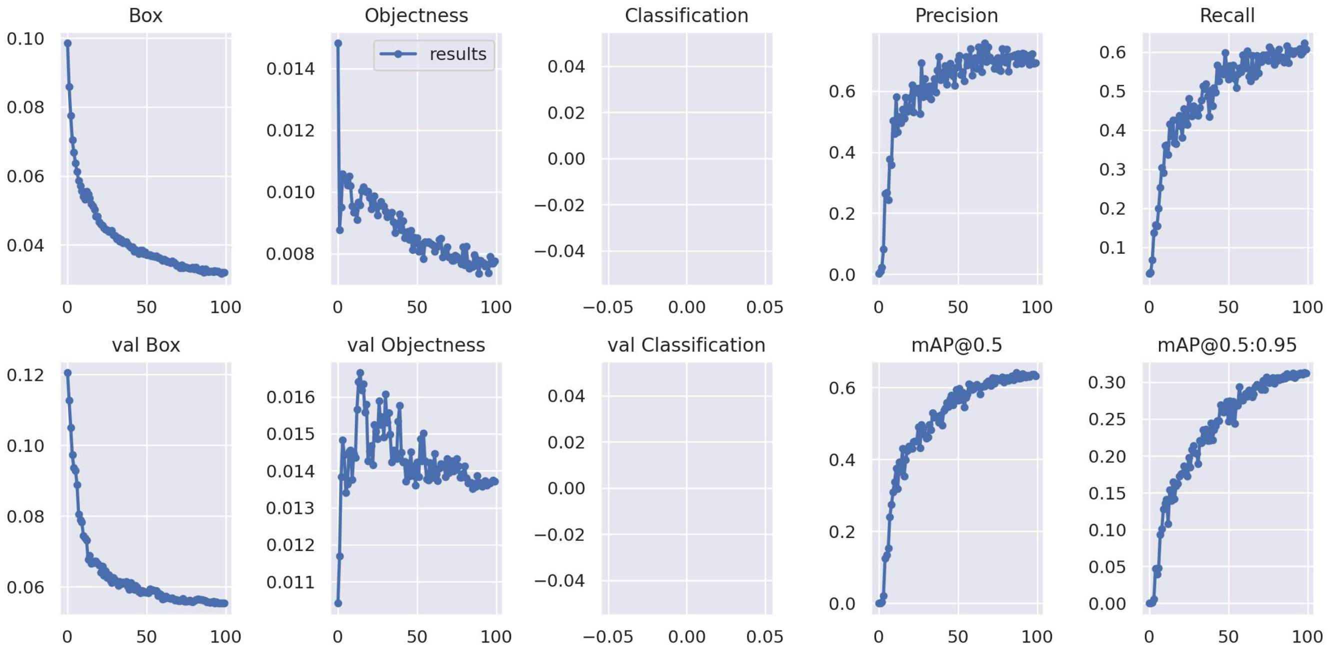
We needed to change the number of classes to 1.

Let the Training BEGIN!

```
!python train.py --epochs 100 --workers 8 --device 0 --batch-size 32 \
--data data/pothole.yaml --img 640 640 --cfg cfg/training/yolov7_pothole-tiny.yaml \
--weights 'yolov7-tiny.pt' --name yolov7_tiny_pothole_fixed_res --hyp data/hyp.scratch.tiny.yaml
```

Show hidden output

For a small model this thing does relatively fine. Here are some results:



Now we get to run a test:

```
!python test.py --weights runs/train/yolov7_tiny_pothole_fixed_res/weights/best.pt --task test --data data/pothole.yaml
```

Show hidden output

Tiny Multi-Resolution Training

Right, our next step is doing multi resolution training.

Here is what is going to be happening:

1. we set a base resolution like 640x640
2. Training then happens in batches
3. Images in the batches can be +-50% of base resolution
4. so between 320x320 and 1280x1280 (according to learnopenvc) 640+50%=960... not 1280.
5. Results in a more robust model that deals with small details
6. Need to train for longer as data is more diverse

```
!python train.py --epochs 100 --workers 4 --device 0 --batch-size 32 \
--data data/pothole.yaml --img 640 640 --cfg cfg/training/yolov7_pothole-tiny.yaml \
--weights 'yolov7-tiny.pt' --name yolov7_tiny_pothole_multi_res --hyp data/hyp.scratch.tiny.yaml \
--multi-scale
```

```
      2/99    0.776G     0.079  0.009544         0   0.08855        58       352: 100% 40/40 [01:11<00:00,  1.79s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 7/7 [00:07<00:00,  1.04s/it]
                 all         401        1034     0.00989      0.0571     0.00131    0.000252

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      3/99     1.16G   0.07209   0.01257         0   0.08466        44       928: 100% 40/40 [01:14<00:00,  1.87s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 7/7 [00:04<00:00,  1.40it/s]
                 all         401        1034      0.0655      0.0687     0.00974     0.00209

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      4/99    0.795G   0.06639   0.01319         0   0.07958        88       736: 100% 40/40 [01:09<00:00,  1.74s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 7/7 [00:07<00:00,  1.04s/it]
                 all         401        1034       0.174       0.113      0.0497      0.0162

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      5/99     11.7G   0.06205   0.01245         0   0.07451        81       352: 100% 40/40 [01:08<00:00,  1.71s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 7/7 [00:08<00:00,  1.19s/it]
                 all         401        1034       0.244       0.158       0.102      0.0364

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      6/99     11.7G   0.05889   0.01134         0   0.07023        46       352: 100% 40/40 [01:08<00:00,  1.71s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 7/7 [00:06<00:00,  1.05it/s]
                 all         401        1034       0.255       0.176        0.14      0.0511

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      7/99     1.26G   0.05662   0.01225         0   0.06887        68       896: 100% 40/40 [01:12<00:00,  1.81s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 7/7 [00:06<00:00,  1.07it/s]
                 all         401        1034       0.372       0.215       0.189      0.0688

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      8/99    0.468G   0.05545   0.01275         0   0.06819        54       512: 100% 40/40 [01:11<00:00,  1.80s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 7/7 [00:05<00:00,  1.26it/s]
                 all         401        1034       0.402       0.283       0.277       0.102

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      9/99     1.47G   0.05287   0.01223         0    0.0651        68       960: 100% 40/40 [01:14<00:00,  1.86s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 7/7 [00:06<00:00,  1.08it/s]
                 all         401        1034       0.477       0.262        0.26      0.0909

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     10/99    0.575G   0.05168   0.01274         0   0.06442        80       384: 100% 40/40 [01:09<00:00,  1.75s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 7/7 [00:08<00:00,  1.20s/it]
                 all         401        1034       0.446        0.35       0.325        0.13

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     11/99     0.82G   0.05071   0.01152         0   0.06223        37       640: 100% 40/40 [01:09<00:00,  1.75s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 7/7 [00:06<00:00,  1.12it/s]
                 all         401        1034       0.474       0.325       0.323       0.132

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     12/99     11.9G   0.05186   0.01097         0   0.06283        76       736: 100% 40/40 [01:05<00:00,  1.65s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 7/7 [00:09<00:00,  1.35s/it]
                 all         401        1034       0.363        0.32       0.255      0.0884

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     13/99    0.778G    0.0503    0.0102         0    0.0605        51       672: 100% 40/40 [01:13<00:00,  1.84s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95:   0% 0/7 [00:00<?, ?it/s]
```

```
!python train.py \
    --weights last.pt \
    --cfg cfg/training/yolov7-tiny.yaml \
    --data data.yaml \
    --epochs 100 --resume

```

```
python3: can't open file '/content/train.py': [Errno 2] No such file or directory
```