

```
In [ ]:
```

```
import numpy as np
import pandas as pd
import cv2
from PIL import Image
from tqdm import tqdm
```

```
In [ ]:
```

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
In [ ]:
```

```
df = pd.read_csv('drive/MyDrive/CS2/dataframe.csv')
df = df.drop(['Unnamed: 0'], axis=1)
df.head()
```

```
Out[ ]:
```

	filename	width	height	class	xmin	ymin	xmax	ymax
0	Czech_000512.jpg	600	600	D20	403	467	559	538
1	Czech_000512.jpg	600	600	D40	321	473	398	522
2	Czech_000646.jpg	600	600	D00	250	400	287	515
3	Czech_003379.jpg	600	600	D00	298	389	354	515
4	Czech_000809.jpg	600	600	D00	264	453	391	527

```
In [ ]:
```

```
from shutil import copyfile

with open('darknet/data/train.txt', 'r') as f:
    for line in f:
        line = line.rstrip()
        fName = line.split('/')[-1]
        fName = fName.split('.')[0]
        srcImg = 'darknet/' + line
        destImg = 'yolov5/data/images/train/' + fName + '.jpg'

        srcLbl = 'darknet/data/labels/' + fName + '.txt'
        destLbl = 'yolov5/data/labels/train/' + fName + '.txt'

        copyfile(srcImg, destImg)
        copyfile(srcLbl, destLbl)

with open('darknet/data/val.txt', 'r') as f:
    for line in f:
        line = line.rstrip()

        fName = line.split('/')[-1]
        fName = fName.split('.')[0]
        srcImg = 'darknet/' + line
        destImg = 'yolov5/data/images/valid/' + fName + '.jpg'

        srcLbl = 'darknet/data/labels/' + fName + '.txt'
        destLbl = 'yolov5/data/labels/valid/' + fName + '.txt'

        copyfile(srcImg, destImg)
        copyfile(srcLbl, destLbl)
```

```
In [ ]:
```

```
!unzip "/content/drive/My Drive/CS2/yolov5.zip"
```



```
inflating: yolov5/data/labels/valid/Japan_013099.txt
inflating: yolov5/data/labels/valid/Japan_013110.txt
inflating: yolov5/data/labels/valid/Japan_013116.txt
inflating: yolov5/data/labels/valid/Japan_013122.txt
inflating: yolov5/data/labels/valid/Japan_013129.txt
inflating: yolov5/data/labels/valid/Japan_013131.txt
inflating: yolov5/data/objects365.yaml
  creating: yolov5/data/scripts/
inflating: yolov5/data/scripts/get_argoverse_hd.sh
inflating: yolov5/data/scripts/get_coco.sh
inflating: yolov5/data/scripts/get_coco128.sh
inflating: yolov5/data/scripts/get_voc.sh
inflating: yolov5/data/SKU-110K.yaml
inflating: yolov5/data/VisDrone.yaml
inflating: yolov5/data/voc.yaml
inflating: yolov5/detect.py
inflating: yolov5/Dockerfile
inflating: yolov5/hubconf.py
inflating: yolov5/LICENSE
  creating: yolov5/models/
inflating: yolov5/models/common.py
inflating: yolov5/models/experimental.py
inflating: yolov5/models/export.py
  creating: yolov5/models/hub/
inflating: yolov5/models/hub/anchors.yaml
inflating: yolov5/models/hub/yolov3-spp.yaml
inflating: yolov5/models/hub/yolov3-tiny.yaml
inflating: yolov5/models/hub/yolov3.yaml
inflating: yolov5/models/hub/yolov5-fpn.yaml
inflating: yolov5/models/hub/yolov5-p2.yaml
inflating: yolov5/models/hub/yolov5-p6.yaml
inflating: yolov5/models/hub/yolov5-p7.yaml
inflating: yolov5/models/hub/yolov5-panet.yaml
inflating: yolov5/models/hub/yolov516.yaml
inflating: yolov5/models/hub/yolov5m6.yaml
inflating: yolov5/models/hub/yolov5s-transformer.yaml
inflating: yolov5/models/hub/yolov5s6.yaml
inflating: yolov5/models/hub/yolov5x6.yaml
inflating: yolov5/models/yolo.py
inflating: yolov5/models/yolov5l.yaml
inflating: yolov5/models/yolov5m.yaml
inflating: yolov5/models/yolov5s.yaml
inflating: yolov5/models/yolov5x.yaml
extracting: yolov5/models/__init__.py
inflating: yolov5/README.md
inflating: yolov5/requirements.txt
inflating: yolov5/test.py
inflating: yolov5/train.py
inflating: yolov5/tutorial.ipynb
  creating: yolov5/utils/
inflating: yolov5/utils/activations.py
inflating: yolov5/utils/autoanchor.py
  creating: yolov5/utils/aws/
inflating: yolov5/utils/aws/mime.sh
inflating: yolov5/utils/aws/resume.py
inflating: yolov5/utils/aws/userdata.sh
extracting: yolov5/utils/aws/__init__.py
inflating: yolov5/utils/datasets.py
  creating: yolov5/utils/flask_rest_api/
inflating: yolov5/utils/flask_rest_api/example_request.py
inflating: yolov5/utils/flask_rest_api/README.md
inflating: yolov5/utils/flask_rest_api/restapi.py
inflating: yolov5/utils/general.py
  creating: yolov5/utils/google_app_engine/
inflating: yolov5/utils/google_app_engine/additional_requirements.txt
inflating: yolov5/utils/google_app_engine/app.yaml
inflating: yolov5/utils/google_app_engine/Dockerfile
inflating: yolov5/utils/google_utils.py
inflating: yolov5/utils/loss.py
inflating: yolov5/utils/metrics.py
inflating: yolov5/utils/plots.py
inflating: yolov5/utils/torch_utils.py
  creating: yolov5/utils/wandb_logging/
inflating: yolov5/utils/wandb_logging/log_dataset.py
inflating: yolov5/utils/wandb_logging/wandb_utils.py
extracting: yolov5/utils/wandb_logging/__init__.py
extracting: yolov5/utils/__init__.py
```

```
creating: yolov5/weights/
inflating: yolov5/weights/download_weights.sh
inflating: yolov5/yolov5l.yaml
```

In []:

```
%cd /content/yolov5

/content/yolov5
```

In []:

```
!rm /content/yolov5/backup -r
!ln -s /content/drive/'My Drive'/CS2/YOLov5_weight/backup /content/yolov5
```

In []:

```
%pip install -qr requirements.txt
```

In []:

```
# Tensorboard (optional)
%load_ext tensorboard
%tensorboard --logdir runs/train
```

In []:

```
import torch
from IPython.display import Image, clear_output # to display images

clear_output()
print(f"Setup complete. Using torch {torch.__version__} ({torch.cuda.get_device_properties(0).name
if torch.cuda.is_available() else 'CPU'})")
```

Setup complete. Using torch 1.8.1+cu101 (Tesla K80)

In []:

```
!python train.py --img 640 --batch 10 --epochs 100 --data dataset.yaml --cfg yolov5l.yaml --
weights runs/train/exp8/weights/best.pt --device 0
```

```
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), done.
From https://github.com/ultralytics/yolov5
 4c5d9bf..7d3686a master      -> origin/master
github: △ WARNING: code is out of date by 2 commits. Use 'git pull' to update or 'git clone
https://github.com/ultralytics/yolov5' to download latest.
YOLOv5 □ v5.0-189-gdaab682 torch 1.8.1+cu101 CUDA:0 (Tesla K80, 11441.1875MB)
```

```
Namespace(adam=False, artifact_alias='latest', batch_size=10, bbox_interval=-1, bucket='',
cache_images=False, cfg='yolov5l.yaml', data='dataset.yaml', device='0', entity=None, epochs=100,
evolve=False, exist_ok=False, global_rank=-1, hyp='data/hyp.scratch.yaml', image_weights=False,
img_size=[640, 640], label_smoothing=0.0, linear_lr=False, local_rank=-1, multi_scale=False,
name='exp', noautoanchor=False, nosave=False, notest=False, project='runs/train', quad=False,
rect=False, resume=False, save_dir='runs/train/exp9', save_period=-1, single_cls=False,
sync_bn=False, total_batch_size=10, upload_dataset=False,
weights='runs/train/exp8/weights/best.pt', workers=8, world_size=1)
tensorboard: Start with 'tensorboard --logdir runs/train', view at http://localhost:6006/
2021-06-15 12:05:25.539728: I tensorflow/stream_executor/platform/default/dso_loader.cc:53]
Successfully opened dynamic library libcudart.so.11.0
```

```
hyperparameters: lr0=0.01, lrf=0.2, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0,
warmup_momentum=0.8, warmup_bias_lr=0.1, box=0.05, cls=0.5, cls_pw=1.0, obj=1.0, obj_pw=1.0, iou_t
=0.2, anchor_t=4.0, fl_gamma=0.0, hsv_h=0.015, hsv_s=0.7, hsv_v=0.4, degrees=0.0, translate=0.1,
scale=0.5, shear=0.0, perspective=0.0, flipud=0.0, fliplr=0.5, mosaic=1.0, mixup=0.0
wandb: (1) Create a W&B account
wandb: (2) Use an existing W&B account
```

```

-----: ., ... an existing user account
wandb: (3) Don't visualize my results
wandb: Enter your choice: 3
wandb: You chose 'Don't visualize my results'
2021-06-15 12:05:34.925567: I tensorflow/stream_executor/platform/default/dso_loader.cc:53]
Successfully opened dynamic library libcudart.so.11.0
wandb: W&B syncing is set to `offline` in this directory. Run `wandb online` or set
WANDB_MODE=online to enable cloud syncing.

      from    n      params      module           arguments
0          -1    1        7040  models.common.Focus      [3, 64, 3]
1          -1    1       73984  models.common.Conv      [64, 128, 3, 2]
2          -1    1      156928  models.common.C3      [128, 128, 3]
3          -1    1      295424  models.common.Conv      [128, 256, 3, 2]
4          -1    1     1611264  models.common.C3      [256, 256, 9]
5          -1    1     1180672  models.common.Conv      [256, 512, 3, 2]
6          -1    1     6433792  models.common.C3      [512, 512, 9]
7          -1    1     4720640  models.common.Conv      [512, 1024, 3, 2]
8          -1    1     2624512  models.common.SPP      [1024, 1024, [5, 9, 13]]
9          -1    1     9971712  models.common.C3      [1024, 1024, 3, False]
10         -1   1     525312  models.common.Conv      [1024, 512, 1, 1]
11         -1   1         0  torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
12        [-1, 6]   1         0  models.common.Concat      [1]
13         -1   1     2757632  models.common.C3      [1024, 512, 3, False]
14         -1   1     131584  models.common.Conv      [512, 256, 1, 1]
15         -1   1         0  torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
16        [-1, 4]   1         0  models.common.Concat      [1]
17         -1   1     690688  models.common.C3      [512, 256, 3, False]
18         -1   1     590336  models.common.Conv      [256, 256, 3, 2]
19        [-1, 14]  1         0  models.common.Concat      [1]
20         -1   1     2495488  models.common.C3      [512, 512, 3, False]
21         -1   1     2360320  models.common.Conv      [512, 512, 3, 2]
22        [-1, 10]  1         0  models.common.Concat      [1]
23         -1   1     9971712  models.common.C3      [1024, 1024, 3, False]
24        [17, 20, 23]  1     59235  models.yolo.Detect      [6, [[10, 13, 16, 30, 3, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [256, 512, 1024]]
```

Model Summary: 499 layers, 46658275 parameters, 46658275 gradients, 114.3 GFLOPs

```

Transferred 648/650 items from runs/train/exp8/weights/best.pt
Scaled weight_decay = 0.00046875
Optimizer groups: 110 .bias, 110 conv.weight, 107 other
train: Scanning 'data/labels/train.cache' images and labels... 11260 found, 0 missing, 0 empty, 93
40 corrupted: 100% 11260/11260 [00:00<00:00, 58814275.27it/s]
val: Scanning 'data/labels/valid.cache' images and labels... 2815 found, 0 missing, 0 empty, 2536
corrupted: 100% 2815/2815 [00:00<00:00, 3588743.39it/s]
Plotting labels...

autoanchor: Analyzing anchors... anchors/target = 4.76, Best Possible Recall (BPR) = 0.9976
Image sizes 640 train, 640 test
Using 2 dataloader workers
Logging results to runs/train/exp9
Starting training for 100 epochs...

```

Epoch	gpu_mem	box	obj	cls	total	labels	img_size
17/99	6.89G	0.06665	0.02207	0.02196	0.1107	29	640: 100% 192/192 [08:2<00:00, 2.72s/it]
00:41<00:00, 2.99s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.68	0.138	0.128	0.0425
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
18/99	6.75G	0.06626	0.02144	0.02121	0.1089	20	640: 100% 192/192 [08:9<00:00, 2.65s/it]
00:26<00:00, 1.91s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.656	0.153	0.133	0.0409
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
19/99	6.75G	0.06518	0.02209	0.02019	0.1075	32	640: 100% 192/192 [08:9<00:00, 2.65s/it]
00:29<00:00, 2.11s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.678	0.207	0.152	0.0462
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
20/99	6.75G	0.06478	0.02092	0.01986	0.1056	29	640: 100% 192/192 [08:0<00:00, 2.66s/it]
00:26<00:00, 1.91s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.492	0.201	0.148	0.0475
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
21/99	6.75G	0.06399	0.02134	0.02012	0.1055	20	640: 100% 192/192 [08:0<00:00, 2.66s/it]
00:26<00:00, 1.91s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.727	0.158	0.2	0.0632
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
22/99	6.75G	0.06275	0.02106	0.01927	0.1031	25	640: 100% 192/192 [08:9<00:00, 2.66s/it]
00:26<00:00, 1.91s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.489	0.186	0.163	0.053
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
23/99	6.75G	0.06163	0.02116	0.01801	0.1008	23	640: 100% 192/192 [08:0<00:00, 2.66s/it]
00:26<00:00, 1.91s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.417	0.284	0.183	0.064
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
24/99	6.75G	0.06113	0.02116	0.01802	0.1003	29	640: 100% 192/192 [08:9<00:00, 2.65s/it]
00:26<00:00, 1.92s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.744	0.177	0.174	0.0578
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
25/99	6.75G	0.06114	0.02076	0.01831	0.1002	28	640: 100% 192/192 [08:9<00:00, 2.66s/it]
00:26<00:00, 1.91s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.712	0.194	0.225	0.0709
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
26/99	6.75G	0.06152	0.02118	0.01803	0.1007	23	640: 100% 192/192 [08:9<00:00, 2.65s/it]
00:26<00:00, 1.91s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.465	0.326	0.217	0.0699
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
27/99	6.75G	0.05978	0.02034	0.01777	0.09789	22	640: 100% 192/192 [08:9<00:00, 2.66s/it]
00:26<00:00, 1.91s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.697	0.185	0.177	0.0568

Epoch	gpu_mem	box	obj	cls	total	labels	img_size
28/99	6.75G	0.05996	0.02055	0.01674	0.09724	24	640: 100% 192/192 [08:00:00, 2.65s/it]
00:26<00:00, 1.91s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.761	0.17	0.208	0.0719
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
29/99	6.75G	0.06003	0.02077	0.01649	0.0973	23	640: 100% 192/192 [08:00:00, 2.65s/it]
00:29<00:00, 2.10s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.341	0.24	0.206	0.0677
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
30/99	6.75G	0.05923	0.02075	0.01672	0.0967	27	640: 100% 192/192 [08:00:00, 2.65s/it]
00:26<00:00, 1.91s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.356	0.22	0.185	0.067
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
31/99	6.75G	0.05841	0.02047	0.01616	0.09503	24	640: 100% 192/192 [08:00:00, 2.65s/it]
00:26<00:00, 1.91s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.457	0.326	0.25	0.0842
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
32/99	6.75G	0.0588	0.02052	0.01663	0.09595	27	640: 100% 192/192 [08:00:00, 2.65s/it]
00:26<00:00, 1.91s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.334	0.349	0.208	0.0869
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
33/99	6.75G	0.05777	0.02017	0.01555	0.09349	23	640: 100% 192/192 [08:00:00, 2.65s/it]
00:26<00:00, 1.91s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.436	0.31	0.267	0.102
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
34/99	6.75G	0.05724	0.02002	0.01525	0.09251	18	640: 100% 192/192 [08:00:00, 2.65s/it]
00:26<00:00, 1.91s/it]	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 14/14
	all	279	409	0.255	0.412	0.257	0.102
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
35/99	6.75G	0.0597	0.01911	0.01039	0.0892	18	640: 3% 5/192 [00:13:08:17, 2.66s/it]Traceback (most recent call last):
File "train.py", line 542, in <module>							
train(hyp, opt, device, tb_writer)							
File "train.py", line 305, in train							
pred = model(imgs) # forward							
File "/usr/local/lib/python3.7/dist-packages/torch/nn/modules/module.py", line 889, in _call_impl							
result = self.forward(*input, **kwargs)							
File "/content/yolov5/models/yolo.py", line 123, in forward							
return self.forward_once(x, profile) # single-scale inference, train							
File "/content/yolov5/models/yolo.py", line 154, in forward_once							
x = m(x) # run							
File "/usr/local/lib/python3.7/dist-packages/torch/nn/modules/module.py", line 889, in _call_impl							
result = self.forward(*input, **kwargs)							
File "/content/yolov5/models/common.py", line 138, in forward							
return self.cv3(torch.cat((self.m(self.cv1(x)), self.cv2(x)), dim=1))							
File "/usr/local/lib/python3.7/dist-packages/torch/nn/modules/module.py", line 889, in _call_impl							
result = self.forward(*input, **kwargs)							
File "/usr/local/lib/python3.7/dist-packages/torch/nn/modules/container.py", line 119, in forward							
input = module(input)							
File "/usr/local/lib/python3.7/dist-packages/torch/nn/modules/module.py", line 889, in _call_impl							
result = self.forward(*input, **kwargs)							

```

result = self.forward(*input, **kwargs)
File "/content/yolov5/models/common.py", line 104, in forward
    return x + self.cv2(self.cv1(x)) if self.add else self.cv2(self.cv1(x))
File "/usr/local/lib/python3.7/dist-packages/torch/nn/modules/module.py", line 889, in
__call_impl
    result = self.forward(*input, **kwargs)
File "/content/yolov5/models/common.py", line 42, in forward
    return self.act(self.bn(self.conv(x)))
File "/usr/local/lib/python3.7/dist-packages/torch/nn/modules/module.py", line 889, in
__call_impl
    result = self.forward(*input, **kwargs)
File "/usr/local/lib/python3.7/dist-packages/torch/nn/modules/conv.py", line 399, in forward
    return self._conv_forward(input, self.weight, self.bias)
File "/usr/local/lib/python3.7/dist-packages/torch/nn/modules/module.py", line 935, in
__getattr__
    if '_parameters' in self.__dict__:
KeyboardInterrupt

wandb: Waiting for W&B process to finish, PID 5435
wandb: Program failed with code 255.
wandb: Find user logs for this run at: /content/yolov5/wandb/offline-run-20210615_120533-
3i7xv1sb/logs/debug.log
wandb: Find internal logs for this run at: /content/yolov5/wandb/offline-run-20210615_120533-3i7xv
1sb/logs/debug-internal.log
wandb: Run summary:
wandb:           train/box_loss 0.05724
wandb:           train/obj_loss 0.02002
wandb:           train/cls_loss 0.01525
wandb:           metrics/precision 0.25495
wandb:           metrics/recall 0.41191
wandb:           metrics/mAP_0.5 0.25702
wandb:           metrics/mAP_0.5:0.95 0.10239
wandb:           val/box_loss 0.04858
wandb:           val/obj_loss 0.01257
wandb:           val/cls_loss 0.01338
wandb:           x/lr0 0.00804
wandb:           x/lr1 0.00804
wandb:           x/lr2 0.00804
wandb:           runtime 9735
wandb:           timestamp 1623768469
wandb:           step 17
wandb: Run history:
wandb:           train/box_loss [REDACTED]
wandb:           train/obj_loss [REDACTED]
wandb:           train/cls_loss [REDACTED]
wandb:           metrics/precision [REDACTED]
wandb:           metrics/recall [REDACTED]
wandb:           metrics/mAP_0.5 [REDACTED]
wandb:           metrics/mAP_0.5:0.95 [REDACTED]
wandb:           val/box_loss [REDACTED]
wandb:           val/obj_loss [REDACTED]
wandb:           val/cls_loss [REDACTED]
wandb:           x/lr0 [REDACTED]
wandb:           x/lr1 [REDACTED]
wandb:           x/lr2 [REDACTED]
wandb:           runtime [REDACTED]
wandb:           timestamp [REDACTED]
wandb:           step [REDACTED]
wandb: You can sync this run to the cloud by running:
wandb: wandb sync /content/yolov5/wandb/offline-run-20210615_120533-3i7xv1sb

```

In []:

```
!python detect.py --weights runs/train/exp9/weights/best.pt --img 640 --conf 0.1 --source
testImages/
```

```

Namespace(agnostic_nms=False, augment=False, classes=None, conf_thres=0.1, device='',
exist_ok=False, half=False, hide_conf=False, hide_labels=False, imgsz=640, iou_thres=0.45,
line_thickness=3, max_det=1000, name='exp', nosave=False, project='runs/detect', save_conf=False,
save_crop=False, save_txt=False, source='testImages/', update=False, view_img=False, weights=
['runs/train/exp9/weights/best.pt'])
YOLOv5 □ v5.0-189-gdaab682 torch 1.8.1+cu101 CUDA:0 (Tesla K80, 11441.1875MB)
```

Fusing layers...

Model Summary: 392 layers, 46627491 parameters, 0 gradients, 114.2 GFLOPs

```
image 1/15 /content/yolov5/testImages/Czech_000396.jpg: 640x640 3 D20s, Done. (0.145s)
image 2/15 /content/yolov5/testImages/Czech_000460.jpg: 640x640 Done. (0.144s)
image 3/15 /content/yolov5/testImages/Czech_000487.jpg: 640x640 2 D00s, Done. (0.145s)
image 4/15 /content/yolov5/testImages/Czech_000491.jpg: 640x640 2 D40s, 3 D00s, Done. (0.145s)
image 5/15 /content/yolov5/testImages/Czech_000672.jpg: 640x640 Done. (0.145s)
image 6/15 /content/yolov5/testImages/Czech_001220.jpg: 640x640 Done. (0.144s)
image 7/15 /content/yolov5/testImages/Czech_001921.jpg: 640x640 1 D40, 1 D00, 2 D20s, Done.
(0.145s)
image 8/15 /content/yolov5/testImages/Japan_000249.jpg: 640x640 1 D00, 3 D44s, Done. (0.146s)
image 9/15 /content/yolov5/testImages/Japan_000288.jpg: 640x640 1 D40, 1 D20, Done. (0.145s)
image 10/15 /content/yolov5/testImages/Japan_000563.jpg: 640x640 3 D20s, Done. (0.145s)
image 11/15 /content/yolov5/testImages/Japan_001229.jpg: 640x640 2 D00s, 1 D44, Done. (0.145s)
image 12/15 /content/yolov5/testImages/Japan_001959.jpg: 640x640 Done. (0.144s)
image 13/15 /content/yolov5/testImages/Japan_003142.jpg: 640x640 2 D20s, 1 D44, Done. (0.145s)
image 14/15 /content/yolov5/testImages/Japan_003356.jpg: 640x640 1 D00, 1 D20, 4 D44s, Done.
(0.145s)
image 15/15 /content/yolov5/testImages/Japan_003362.jpg: 640x640 1 D20, Done. (0.145s)
Results saved to runs/detect/exp21
Done. (2.501s)
```

In []:

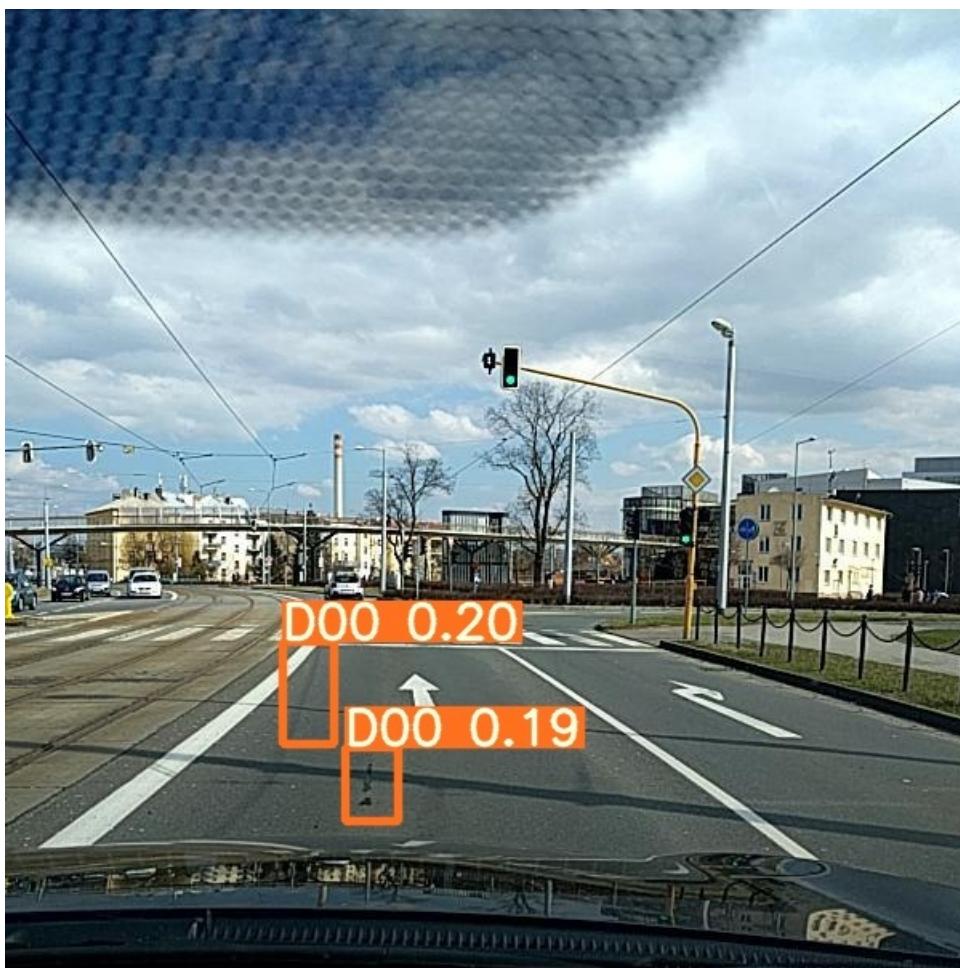
```
import os
import cv2
from google.colab.patches import cv2_imshow
```

In []:

```
results = os.listdir('testImages')
testDir = 'testImages/'
resultDir = 'runs/detect/exp21/'
for img in results:
    if '.jpg' in img:
        dirTest = testDir + img
        dirRes = resultDir + img
        print(dirTest)
        image1 = cv2.imread(dirTest)
        cv2_imshow(image1)
        image2 = cv2.imread(dirRes)
        cv2_imshow(image2)
```

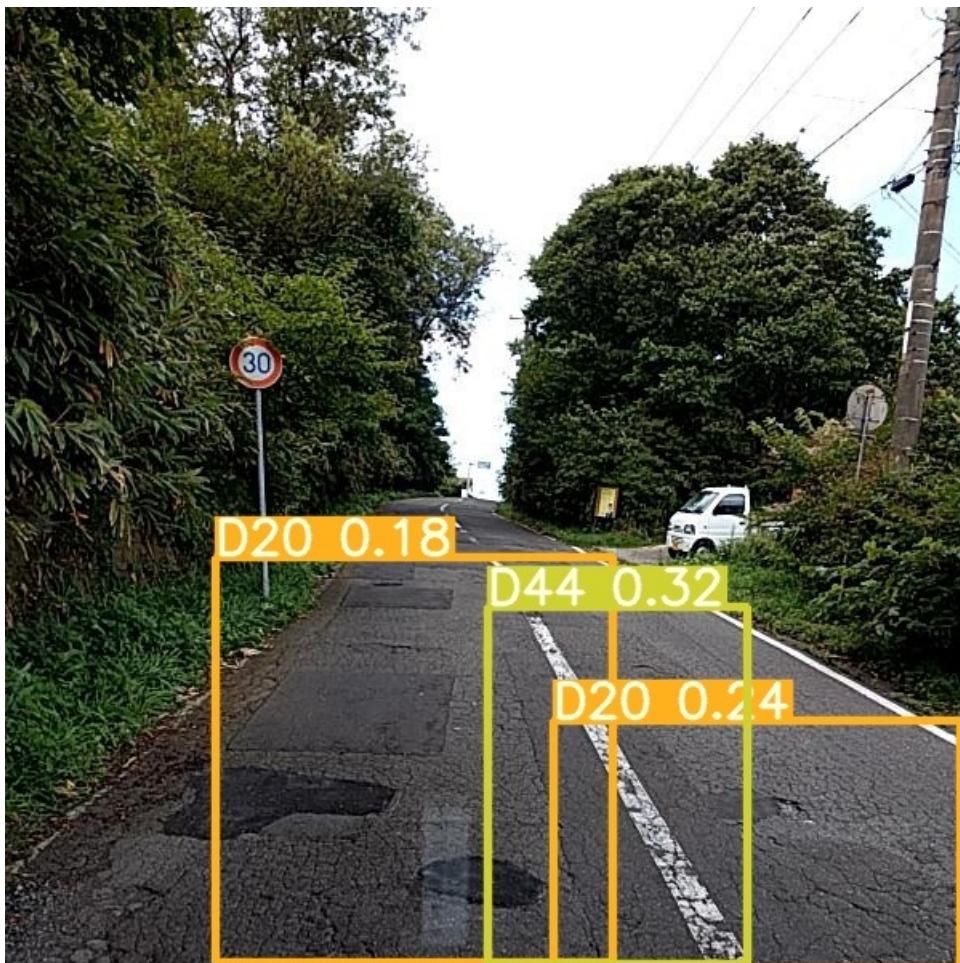
testImages/Czech_000487.jpg





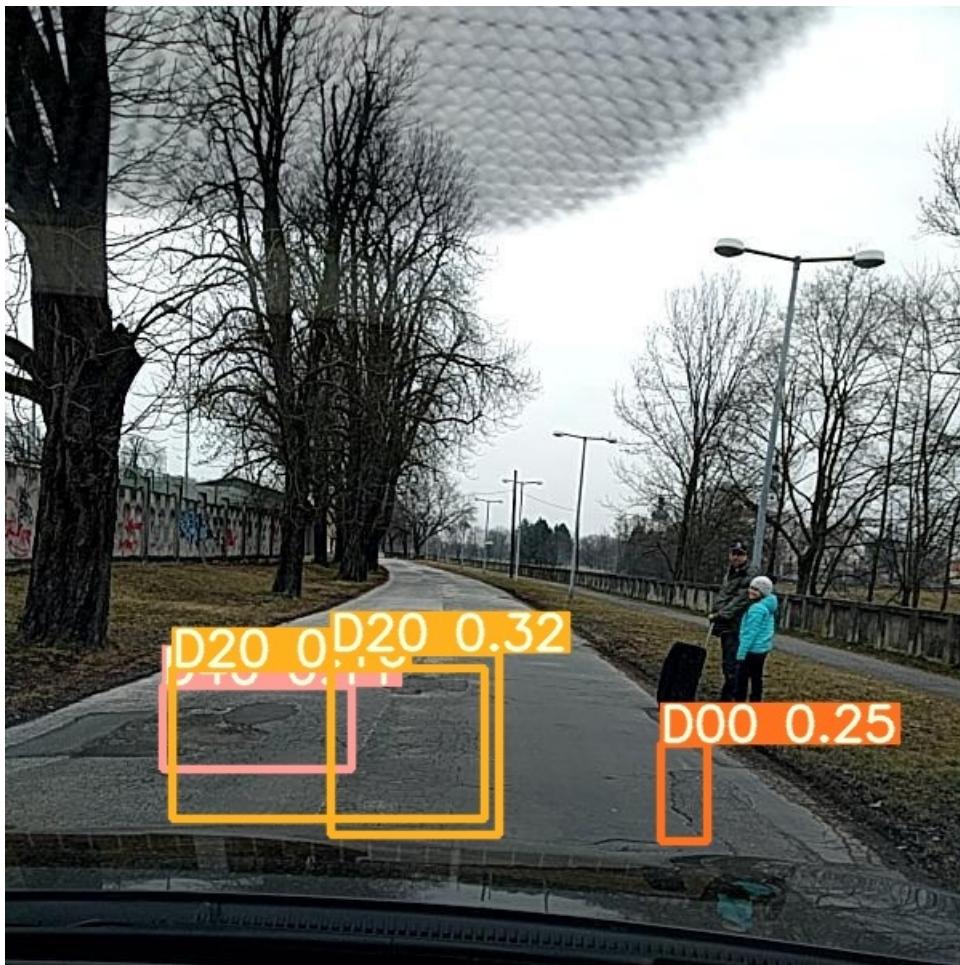
testImages/Japan_003142.jpg





testImages/Czech_001921.jpg





testImages/Czech_000396.jpg





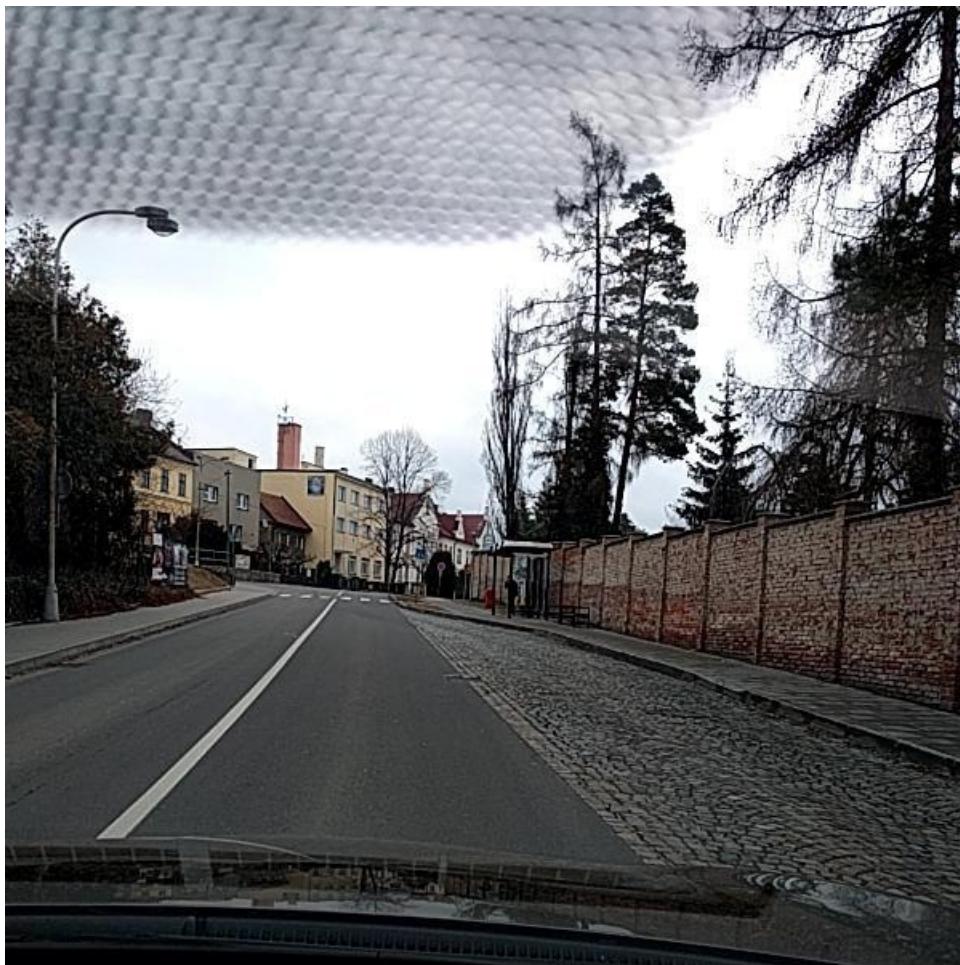
testImages/Japan_001959.jpg





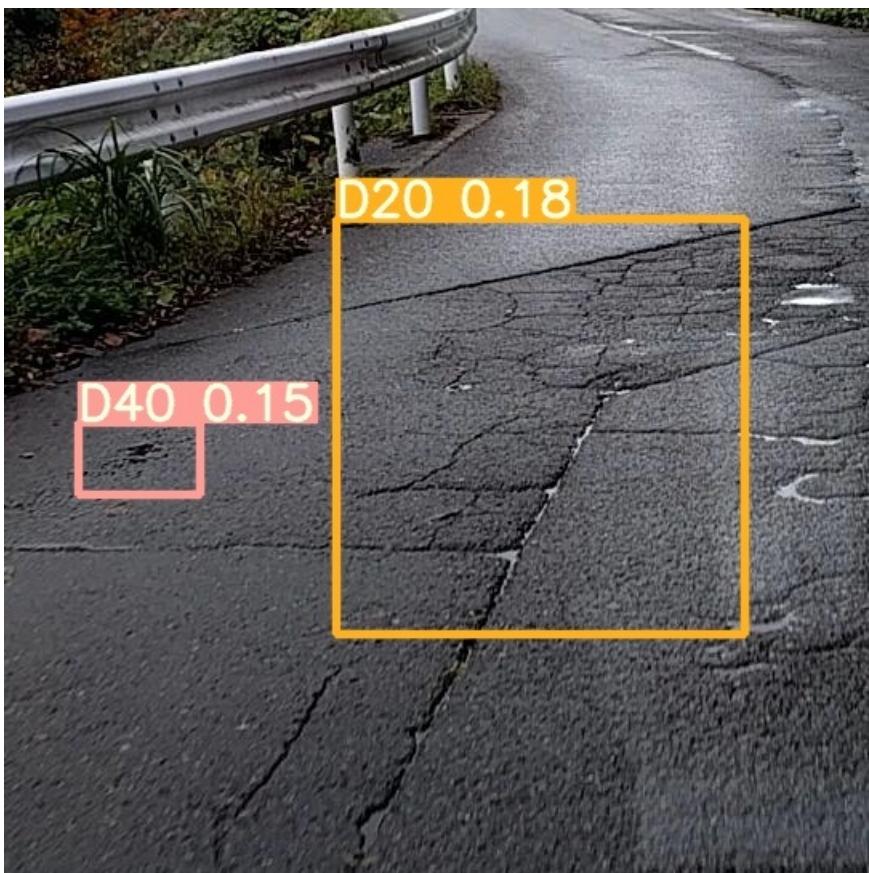
testImages/Czech_000672.jpg





testImages/Japan_000288.jpg





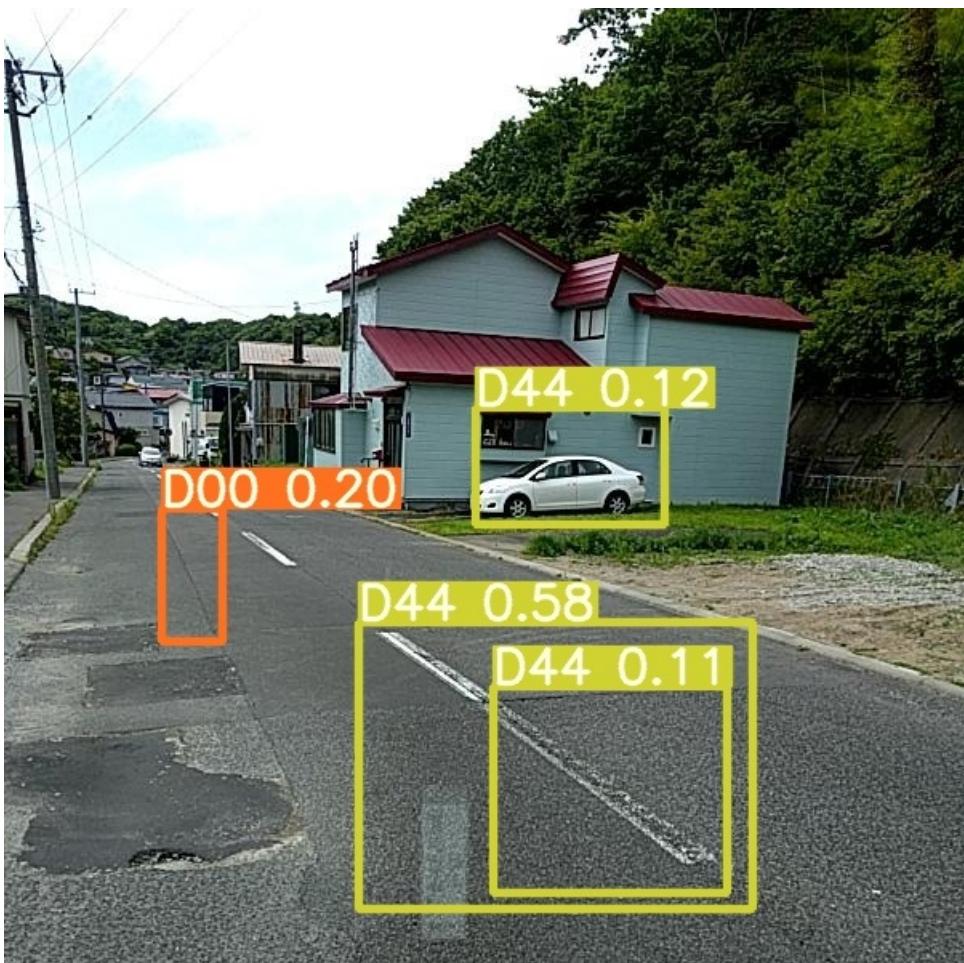
testImages/Czech_001220.jpg





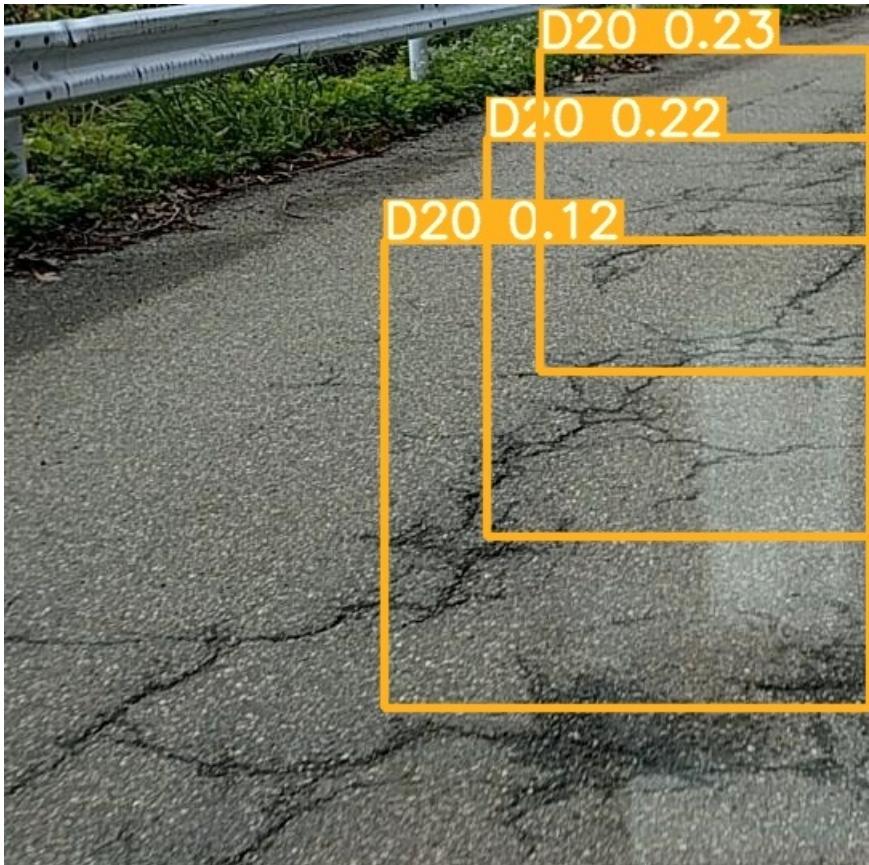
testImages/Japan_000249.jpg





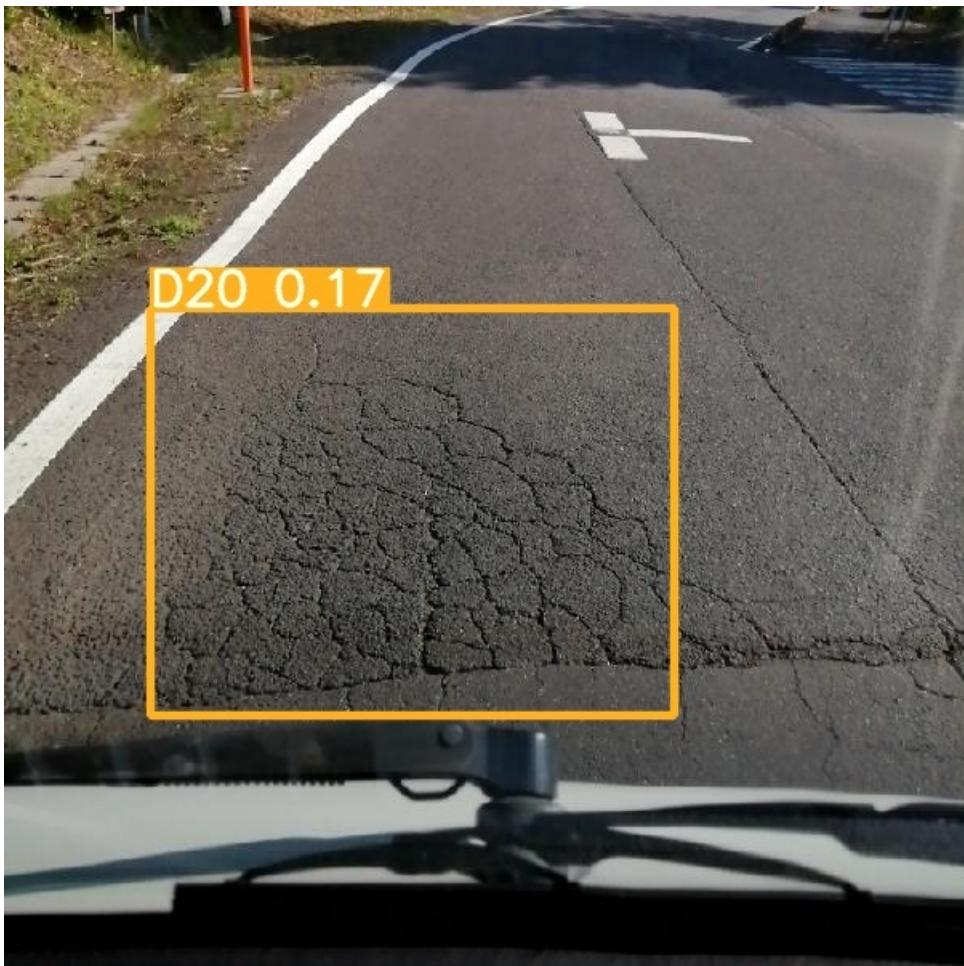
testImages/Japan_000563.jpg





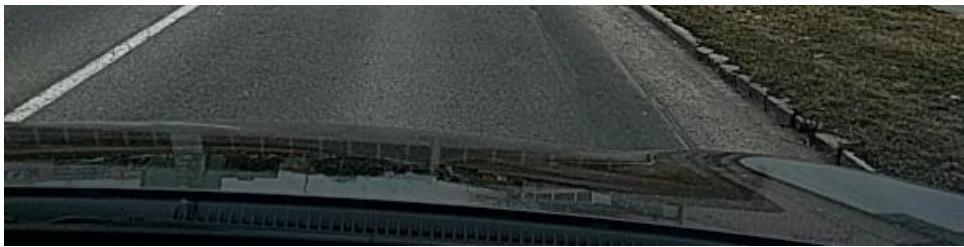
testImages/Japan_003362.jpg





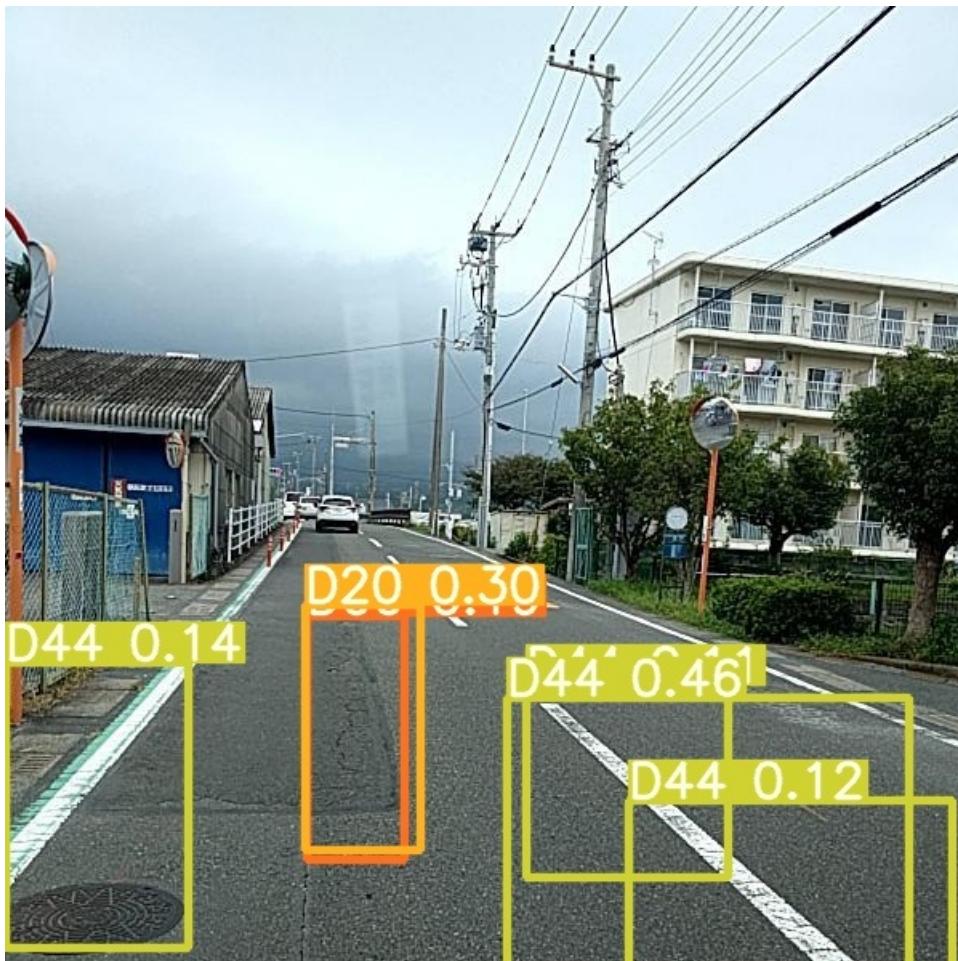
testImages/Czech_000460.jpg





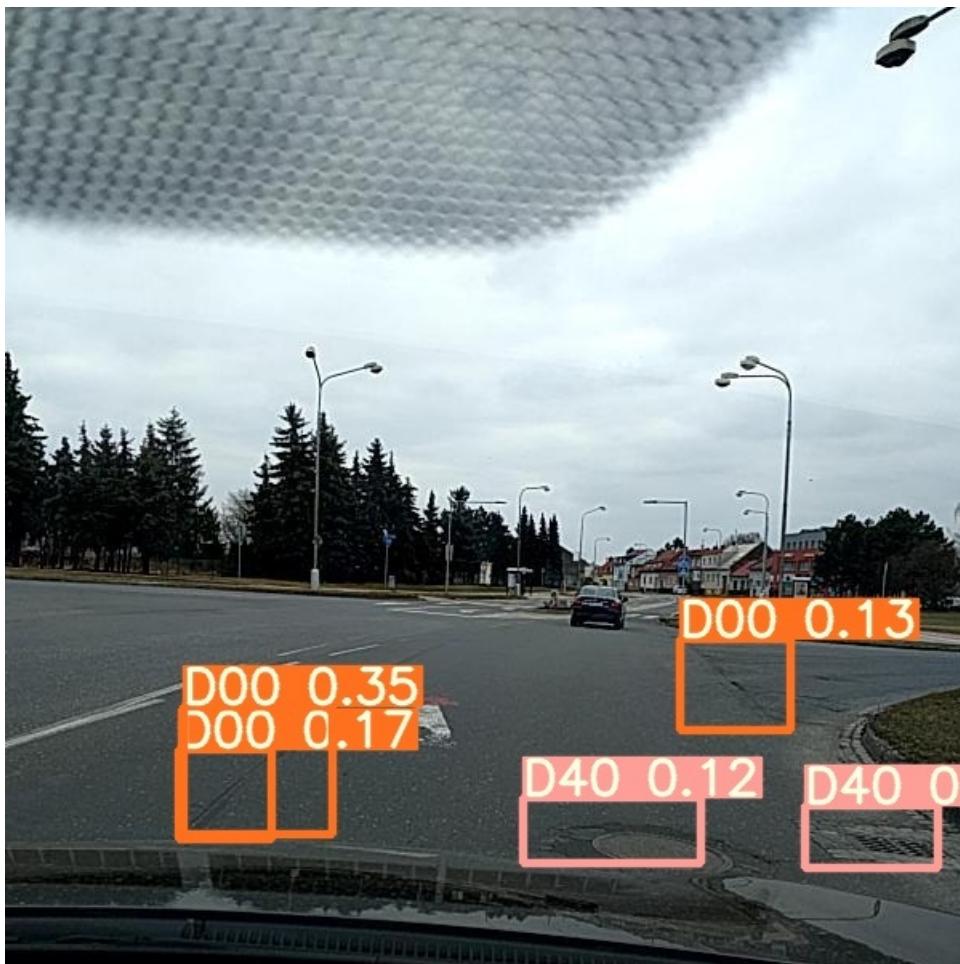
testImages/Japan_003356.jpg





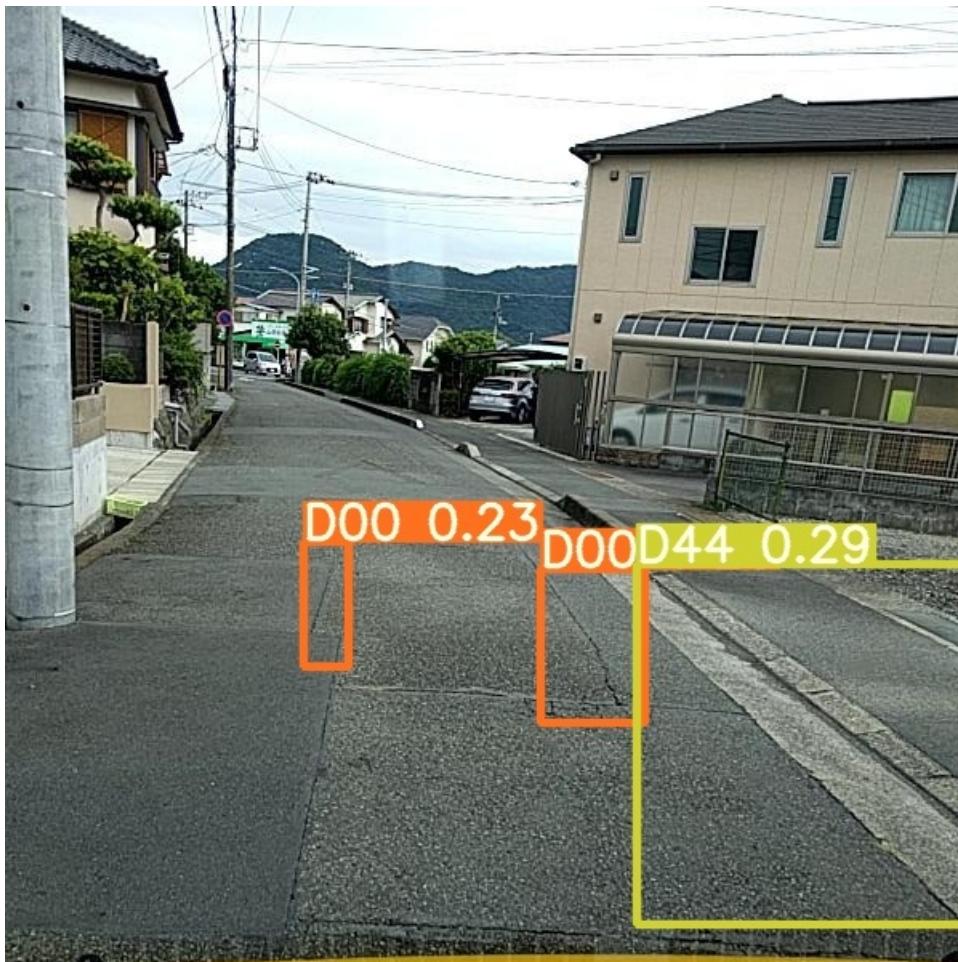
testImages/Czech_000491.jpg





testImages/Japan_001229.jpg





Conclusion:

- With yoloV5, results are really satisfactory as compared to yoloV3.
- I trained it for 34 epochs and observed significant change with increasing the number of epochs.
- Since I am performing a custom training, it is taking lots of time but results are worth it.
- After training with yoloV5, I also tried with EfficientDet3 model from Tensorflow 2.0.
- Custom Training with EfficientDet was real slow on Colab gpu. Firstly, training was not starting for any batch_size >1.
- So I trained with batch_size=1. It took multiple hours to train.
- I was getting losses as NaN for many epochs and for some it was in between 1 to 2. But when I tried to predict boxes with trained weights, results were not good as compared to yoloV5.
- So overall, yoloV5 was the best which I could get.