Final Model

This notebook describes:

1. Setup (dependencies and imports)
2. Dataset (definitions of train and test sets)
3. Training (how the training was performed)
4. Error analysis (comparison with ground truths from test set and model results)
5. Export (of predicitons on the test set)

In [238…
```
%load_ext autoreload
%autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

1. Setup

Dependencies

Tested using torch.**version** = 1.8.0. To install other dependencies run:

```
$ pip install --upgrade pip
$ pip install torchvision
$ pip install cython
$ pip install pycocotools
$ pip install torch-summary
```

Helper functions

We will import some helper functions to simplify training and evaluating detection model. We will use `/references/detection/{engine, utils, transforms}.py` from `pytorch.vision` repository:

```
$ git clone https://github.com/pytorch/vision.git
$ mkdir pytorch-helpers
$ cd vision
$ git checkout v0.3.0
$ cp references/detection/utils.py ../pytorch-helpers/
$ cp references/detection/transforms.py ../pytorch-helpers/
$ cp references/detection/coco_eval.py ../pytorch-helpers/
$ cp references/detection/engine.py ../pytorch-helpers/
$ cp references/detection/coco_utils.py ../pytorch-helpers/
$ cd ../pytorch-helpers/
$ chmod +x *
```

Imports

In [239…
```
import os
import sys
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import time
```

In [240…
```
import scipy.io
import torch
import torch.utils.data
import torchvision
```

In [241…
```
# ../src/pytorch-helpers/
module_path = os.path.abspath(os.path.join('../src/pytorch-helpers/'))
if module_path not in sys.path:
    sys.path.append(module_path)

from engine import train_one_epoch, evaluate # test_one_epoch
import utils
import transforms as T
from torchsummary import summary
# summary(model,input_size=(3,2048,1024))
```

In [242…
```
base_model_path = '../src/main-model/'

module_path = os.path.abspath(os.path.join(base_model_path))
if module_path not in sys.path:
    sys.path.append(module_path)

import fasterutils
import fasterrcnn
```

2. Dataset

From:

```
* 2975 (2500 with persons) in train (citypersons train) set
* 500 (441 with persons) in test (citypersons validation) set
```

we create:

```
anno_train
anno_test

imgs_path_train
imgs_path_test

imgs_paths_train
imgs_paths_test
```

```
anno_dict
anno_dict_val

imgs_train
imgs_test
```

In [243…
```
imgs_path_train = './datasets/cityscapes/train/'
imgs_path_test = './datasets/cityscapes/test/'
```

In [245…
```
anno_dict = np.load('anno-all.npy', allow_pickle='TRUE').item()
anno_dict_val = np.load('anno-val.npy', allow_pickle='TRUE').item()
# anno_dict = {**anno_dict, **anno_dict_val}

len(list(anno_dict.keys())), len(list(anno_dict_val.keys()))
```

Out[245… (2500, 441)

In [246…
```
imgs1 = os.listdir(imgs_path_train)
imgs2 = list(anno_dict.keys())
imgs3 = list(set(imgs1) & set(imgs2))

anno_train = {}
imgs_train = []
for img in imgs3:
    anno_train[img] = anno_dict[img]
    imgs_train.append(img)
```

In [247…
```
imgs1 = os.listdir(imgs_path_test)
imgs2 = list(anno_dict_val.keys())
imgs3 = list(set(imgs1) & set(imgs2))

anno_test = {}
imgs_test = []
for img in imgs3:
    anno_test[img] = anno_dict_val[img]
    imgs_test.append(img)
```

In [248…
```
for img in imgs_test:
    anno_test[img]

for img in imgs_train:
    anno_train[img]
```

In [249…
```
len(imgs_train), len(imgs_test)
```

Out[249… (2500, 441)

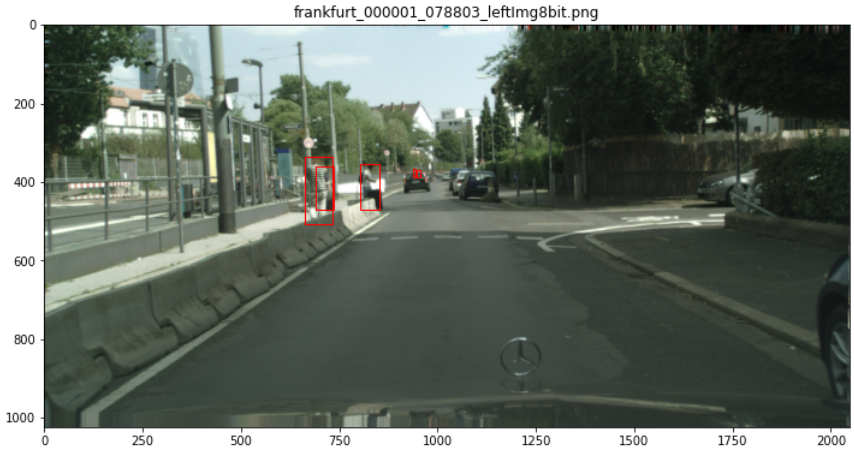
In [250…
```
i = 0
img_name = imgs_test[i]
img_path = imgs_path_test + img_name
img = Image.open(img_path)
bboxes = anno_test[img_name]
```

In [251…
```
imgs_test[:5]
```

Out[251…
```
['frankfurt_000001_078803_leftImg8bit.png',
 'frankfurt_000001_023769_leftImg8bit.png',
 'frankfurt_000000_012121_leftImg8bit.png',
 'frankfurt_000001_064651_leftImg8bit.png',
 'frankfurt_000001_046504_leftImg8bit.png']
```

In [252…
```
plt.rcParams['figure.figsize'] = [12, 8]
```

In [253…
```
fig, ax = plt.subplots()
ax.imshow(img)

for bbox in bboxes:
    rect = patches.Rectangle(
        (bbox[0], bbox[1]), bbox[2], bbox[3],
        linewidth=1, edgecolor='r', facecolor='none')

    ax.add_patch(rect)

plt.title(img_name)
plt.show()
```



In [254…

```
img_np = np.array(img)
H, W = img_np.shape[:2]
H, W
```

Out[254… (1024, 2048)

In [255…
```
imgs_paths_train = [(imgs_path_train + img_name) for img_name in imgs_train]
imgs_paths_test = [(imgs_path_test + img_name) for img_name in imgs_test]
```

In [257…
```
dataset = fasterrcnn.Dataset(imgs_paths_train, imgs_train, anno_train)
dataset_test = fasterrcnn.Dataset(imgs_paths_test, imgs_test, anno_test)
```

In [258…
```
## test it
data = []
for i in range(10):
    data.append(dataset_test[i])
```

In [259…
```
## test it
data = []
for i in range(10):
    data.append(dataset[i])
```

In [260…
```
data[0][0]
```

Out[260…

3. Training

We do not need to add a mean/std normalization nor image rescaling in the data transforms, as those are handled internally by the Faster R-CNN model.

In [261…
```
# split the dataset into train and test
dataset = fasterrcnn.Dataset(
    imgs_paths_train,
    imgs_train,
    anno_train,
    fasterutils.get_transform(train=True))

dataset_test = fasterrcnn.Dataset(
    imgs_paths_test,
    imgs_test,
    anno_test,
    fasterutils.get_transform(train=False))

## permute the indices
torch.manual_seed(1)
indices = torch.randperm(len(dataset)).tolist()
```

In [1]:
```
## check the results
# print(len(indices), len(dataset_test))
# print(dataset[0],  dataset_test[0])
```

In [263…
```
## define train and test sets
# dataset = torch.utils.data.Subset(dataset, indices[:-36])
# dataset_test = torch.utils.data.Subset(dataset_test, indices[-36:])

## define training and validation data loaders
data_loader = torch.utils.data.DataLoader(
    dataset, batch_size=2, shuffle=True, num_workers=4,
    collate_fn=utils.collate_fn)

data_loader_test = torch.utils.data.DataLoader(
    dataset_test, batch_size=1, shuffle=False, num_workers=4,
    collate_fn=utils.collate_fn)
```

In [264…
```
## check the results
print('Batch size = %d' % (data_loader.batch_size))
```

```
data_loader.dataset[0], data_loader_test.dataset[0]
```

```
Batch size = 2
```

Out[264…

```
((tensor([[[0.3412, 0.3529, 0.3647,  ..., 0.0000, 0.0000, 0.0000],
           [0.4510, 0.4471, 0.4392,  ..., 0.0000, 0.0000, 0.0000],
           [0.5804, 0.5647, 0.5529,  ..., 0.0000, 0.0000, 0.0000],
           ...,
           [0.2157, 0.2196, 0.2157,  ..., 0.1804, 0.1804, 0.1804],
           [0.2196, 0.2196, 0.2157,  ..., 0.1765, 0.1725, 0.1725],
           [0.2196, 0.2196, 0.2157,  ..., 0.1647, 0.1647, 0.1647]],

          [[0.3882, 0.4039, 0.4157,  ..., 0.0000, 0.1569, 0.0000],
           [0.4745, 0.4706, 0.4706,  ..., 0.0000, 0.1686, 0.0000],
           [0.6000, 0.5843, 0.5725,  ..., 0.0000, 0.1765, 0.0000],
           ...,
           [0.2745, 0.2784, 0.2745,  ..., 0.2196, 0.2196, 0.2196],
           [0.2745, 0.2784, 0.2745,  ..., 0.2157, 0.2157, 0.2157],
           [0.2745, 0.2784, 0.2745,  ..., 0.2118, 0.2118, 0.2118]],

          [[0.4000, 0.4039, 0.4078,  ..., 0.0000, 0.0745, 0.0000],
           [0.5255, 0.5137, 0.5020,  ..., 0.0000, 0.0824, 0.0000],
           [0.5922, 0.5804, 0.5647,  ..., 0.0000, 0.0941, 0.0000],
           ...,
           [0.2510, 0.2510, 0.2431,  ..., 0.1725, 0.1725, 0.1725],
           [0.2510, 0.2510, 0.2431,  ..., 0.1725, 0.1686, 0.1686],
           [0.2510, 0.2510, 0.2431,  ..., 0.1725, 0.1725, 0.1725]]]),
  {'boxes': tensor([[1135.,  451., 1167.,  530.],
           [1179.,  432., 1232.,  562.],
           [1229.,  429., 1284.,  564.],
           [1322.,  438., 1372.,  560.],
           [1393.,  434., 1446.,  564.]]),
   'labels': tensor([1, 1, 1, 1, 1]),
   'image_id': tensor([0]),
   'area': tensor([2528., 6890., 7425., 6100., 6890.]),
   'iscrowd': tensor([0, 0, 0, 0, 0])}),
 (tensor([[[0.0000, 0.0000, 0.0000,  ..., 0.0196, 0.0471, 0.0000],
           [0.1490, 0.1490, 0.1490,  ..., 0.0235, 0.0471, 0.0000],
           [0.0000, 0.0000, 0.0000,  ..., 0.0275, 0.0471, 0.0000],
           ...,
           [0.2863, 0.2863, 0.2863,  ..., 0.2667, 0.2667, 0.2627],
           [0.2863, 0.2863, 0.2863,  ..., 0.1451, 0.2706, 0.2706],
           [0.2863, 0.2863, 0.2863,  ..., 0.1490, 0.1490, 0.1490]],

          [[0.0902, 0.0980, 0.1098,  ..., 0.1098, 0.0784, 0.0000],
           [0.2667, 0.2667, 0.2667,  ..., 0.1098, 0.0784, 0.0000],
           [0.0000, 0.0000, 0.0157,  ..., 0.1098, 0.0784, 0.0000],
           ...,
           [0.3373, 0.3373, 0.3373,  ..., 0.3333, 0.3333, 0.3294],
           [0.3373, 0.3373, 0.3373,  ..., 0.1843, 0.3333, 0.3333],
           [0.3373, 0.3373, 0.3373,  ..., 0.1922, 0.1882, 0.1882]],

          [[0.0000, 0.0000, 0.0000,  ..., 0.0863, 0.0549, 0.0000],
           [0.0000, 0.0000, 0.0157,  ..., 0.0863, 0.0549, 0.0000],
           [0.0000, 0.0000, 0.0000,  ..., 0.0863, 0.0549, 0.0000],
           ...,
           [0.2980, 0.2941, 0.2941,  ..., 0.2863, 0.2863, 0.2863],
           [0.2980, 0.2941, 0.2941,  ..., 0.1647, 0.2863, 0.2863],
           [0.2980, 0.2941, 0.2980,  ..., 0.1725, 0.1725, 0.1686]]]),
  {'boxes': tensor([[937., 368., 945., 388.],
           [948., 369., 955., 387.],
           [663., 337., 733., 508.],
           [691., 359., 737., 470.],
           [804., 355., 852., 471.]]),
   'labels': tensor([1, 1, 1, 1, 1]),
   'image_id': tensor([0]),
   'area': tensor([  160.,   126., 11970.,  5106.,  5568.]),
   'iscrowd': tensor([0, 0, 0, 0, 0])}))
```

In [265…

```
device = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
device
```

Out[265…

```
device(type='cuda')
```

In [266…

```
model = fasterutils.get_model()
model.to(device)
```

Out[266…

```
FasterRCNN(
  (transform): GeneralizedRCNNTransform(
      Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
      Resize(min_size=(800,), max_size=1333, mode='bilinear')
  )
  (backbone): BackboneWithFPN(
    (body): IntermediateLayerGetter(
      (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
      (bn1): FrozenBatchNorm2d(64, eps=0.0)
      (relu): ReLU(inplace=True)
      (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
      (layer1): Sequential(
        (0): Bottleneck(
          (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): FrozenBatchNorm2d(64, eps=0.0)
          (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
          (bn2): FrozenBatchNorm2d(64, eps=0.0)
          (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): FrozenBatchNorm2d(256, eps=0.0)
          (relu): ReLU(inplace=True)
          (downsample): Sequential(
            (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): FrozenBatchNorm2d(256, eps=0.0)
          )
        )
        (1): Bottleneck(
          (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): FrozenBatchNorm2d(64, eps=0.0)
          (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
          (bn2): FrozenBatchNorm2d(64, eps=0.0)
          (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): FrozenBatchNorm2d(256, eps=0.0)
          (relu): ReLU(inplace=True)
        )
        (2): Bottleneck(
          (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): FrozenBatchNorm2d(64, eps=0.0)
          (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
          (bn2): FrozenBatchNorm2d(64, eps=0.0)
          (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): FrozenBatchNorm2d(256, eps=0.0)
          (relu): ReLU(inplace=True)
        )
      )
```

```
  (layer2): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(128, eps=0.0)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(128, eps=0.0)
      (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(512, eps=0.0)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): FrozenBatchNorm2d(512, eps=0.0)
      )
    )
    (1): Bottleneck(
      (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(128, eps=0.0)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(128, eps=0.0)
      (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(512, eps=0.0)
      (relu): ReLU(inplace=True)
    )
    (2): Bottleneck(
      (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(128, eps=0.0)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(128, eps=0.0)
      (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(512, eps=0.0)
      (relu): ReLU(inplace=True)
    )
    (3): Bottleneck(
      (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(128, eps=0.0)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(128, eps=0.0)
      (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(512, eps=0.0)
      (relu): ReLU(inplace=True)
    )
  )
  (layer3): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(256, eps=0.0)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(256, eps=0.0)
      (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(1024, eps=0.0)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(512, 1024, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): FrozenBatchNorm2d(1024, eps=0.0)
      )
    )
    (1): Bottleneck(
      (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(256, eps=0.0)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(256, eps=0.0)
      (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(1024, eps=0.0)
      (relu): ReLU(inplace=True)
    )
    (2): Bottleneck(
      (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(256, eps=0.0)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(256, eps=0.0)
      (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(1024, eps=0.0)
      (relu): ReLU(inplace=True)
    )
    (3): Bottleneck(
      (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(256, eps=0.0)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(256, eps=0.0)
      (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(1024, eps=0.0)
      (relu): ReLU(inplace=True)
    )
    (4): Bottleneck(
      (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(256, eps=0.0)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(256, eps=0.0)
      (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(1024, eps=0.0)
      (relu): ReLU(inplace=True)
    )
    (5): Bottleneck(
      (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(256, eps=0.0)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(256, eps=0.0)
      (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(1024, eps=0.0)
      (relu): ReLU(inplace=True)
    )
  )
  (layer4): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(512, eps=0.0)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(512, eps=0.0)
      (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(2048, eps=0.0)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(1024, 2048, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): FrozenBatchNorm2d(2048, eps=0.0)
      )
    )
    (1): Bottleneck(
      (conv1): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): FrozenBatchNorm2d(512, eps=0.0)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): FrozenBatchNorm2d(512, eps=0.0)
      (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): FrozenBatchNorm2d(2048, eps=0.0)
      (relu): ReLU(inplace=True)
```

```
        )
        (2): Bottleneck(
          (conv1): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): FrozenBatchNorm2d(512, eps=0.0)
          (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
          (bn2): FrozenBatchNorm2d(512, eps=0.0)
          (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): FrozenBatchNorm2d(2048, eps=0.0)
          (relu): ReLU(inplace=True)
        )
      )
    )
    (fpn): FeaturePyramidNetwork(
      (inner_blocks): ModuleList(
        (0): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
        (1): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        (2): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1))
        (3): Conv2d(2048, 256, kernel_size=(1, 1), stride=(1, 1))
      )
      (layer_blocks): ModuleList(
        (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (3): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      )
      (extra_blocks): LastLevelMaxPool()
    )
  )
  (rpn): RegionProposalNetwork(
    (anchor_generator): AnchorGenerator()
    (head): RPNHead(
      (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (cls_logits): Conv2d(256, 3, kernel_size=(1, 1), stride=(1, 1))
      (bbox_pred): Conv2d(256, 12, kernel_size=(1, 1), stride=(1, 1))
    )
  )
  (roi_heads): RoIHeads(
    (box_roi_pool): MultiScaleRoIAlign(featmap_names=['0', '1', '2', '3'], output_size=(7, 7), sampling_ratio=2)
    (box_head): TwoMLPHead(
      (fc6): Linear(in_features=12544, out_features=1024, bias=True)
      (fc7): Linear(in_features=1024, out_features=1024, bias=True)
    )
    (box_predictor): FastRCNNPredictor(
      (cls_score): Linear(in_features=1024, out_features=2, bias=True)
      (bbox_pred): Linear(in_features=1024, out_features=8, bias=True)
    )
  )
)
```

In [267…

```
# summary(model,input_size=(3,2048,1024))
```

In [286…

```
# construct an optimizer
params = [p for p in model.parameters() if p.requires_grad]
optimizer = torch.optim.SGD(params, lr=0.005,
                            momentum=0.9, weight_decay=0.0005)
```

In [287…

```
# and a learning rate scheduler which decreases the learning rate by
# 10x every 3 epochs
lr_scheduler = torch.optim.lr_scheduler.StepLR(optimizer,
                                               step_size=3,
                                               gamma=0.1)
```

Let's train the model for 10 epochs, evaluating at the end of every epoch.

In [288…

```
num_epochs = 10

tstart = time.time()
for epoch in range(num_epochs):
    # train for one epoch, printing every 10 iterations
    train_one_epoch(model, optimizer, data_loader, device, epoch, print_freq=100)
    # update the learning rate
    lr_scheduler.step()
    # evaluate on the test dataset
    evaluate(model, data_loader_test, device=device)

tend = time.time()
```

```
Epoch: [0]  [   0/1250]  eta: 0:14:16  lr: 0.000010  loss: 0.2760 (0.2760)  loss_classifier: 0.0981 (0.0981)  loss_box_reg: 0.1372 (0.1372)  loss_objectne
ss: 0.0177 (0.0177)  loss_rpn_box_reg: 0.0230 (0.0230)  time: 0.6852  data: 0.5501  max mem: 2825
Epoch: [0]  [ 100/1250]  eta: 0:03:12  lr: 0.000509  loss: 0.3715 (0.4192)  loss_classifier: 0.0991 (0.1027)  loss_box_reg: 0.1496 (0.1642)  loss_objectne
ss: 0.0344 (0.0512)  loss_rpn_box_reg: 0.0553 (0.1011)  time: 0.1616  data: 0.0472  max mem: 2825
Epoch: [0]  [ 200/1250]  eta: 0:02:50  lr: 0.001009  loss: 0.5698 (0.4482)  loss_classifier: 0.1441 (0.1129)  loss_box_reg: 0.2675 (0.1842)  loss_objectne
ss: 0.0427 (0.0487)  loss_rpn_box_reg: 0.1085 (0.1024)  time: 0.1672  data: 0.0515  max mem: 2825
Epoch: [0]  [ 300/1250]  eta: 0:02:34  lr: 0.001508  loss: 0.4671 (0.4452)  loss_classifier: 0.1084 (0.1111)  loss_box_reg: 0.1921 (0.1822)  loss_objectne
ss: 0.0399 (0.0488)  loss_rpn_box_reg: 0.0812 (0.1031)  time: 0.1761  data: 0.0612  max mem: 2825
Epoch: [0]  [ 400/1250]  eta: 0:02:18  lr: 0.002008  loss: 0.3995 (0.4391)  loss_classifier: 0.1027 (0.1106)  loss_box_reg: 0.1746 (0.1814)  loss_objectne
ss: 0.0348 (0.0482)  loss_rpn_box_reg: 0.0799 (0.0988)  time: 0.1787  data: 0.0623  max mem: 2825
Epoch: [0]  [ 500/1250]  eta: 0:02:00  lr: 0.002507  loss: 0.3437 (0.4517)  loss_classifier: 0.0926 (0.1120)  loss_box_reg: 0.1378 (0.1817)  loss_objectne
ss: 0.0527 (0.0536)  loss_rpn_box_reg: 0.0319 (0.1043)  time: 0.1651  data: 0.0497  max mem: 2825
Epoch: [0]  [ 600/1250]  eta: 0:01:44  lr: 0.003007  loss: 0.5068 (0.4579)  loss_classifier: 0.1212 (0.1137)  loss_box_reg: 0.2066 (0.1856)  loss_objectne
ss: 0.0379 (0.0533)  loss_rpn_box_reg: 0.0820 (0.1053)  time: 0.1599  data: 0.0413  max mem: 2825
Epoch: [0]  [ 700/1250]  eta: 0:01:26  lr: 0.003506  loss: 0.3342 (0.4630)  loss_classifier: 0.0948 (0.1147)  loss_box_reg: 0.1423 (0.1877)  loss_objectne
ss: 0.0456 (0.0536)  loss_rpn_box_reg: 0.0282 (0.1070)  time: 0.1448  data: 0.0272  max mem: 2825
Epoch: [0]  [ 800/1250]  eta: 0:01:11  lr: 0.004006  loss: 0.4468 (0.4747)  loss_classifier: 0.1157 (0.1173)  loss_box_reg: 0.2222 (0.1912)  loss_objectne
ss: 0.0477 (0.0556)  loss_rpn_box_reg: 0.0847 (0.1106)  time: 0.1664  data: 0.0487  max mem: 2825
Epoch: [0]  [ 900/1250]  eta: 0:00:55  lr: 0.004505  loss: 0.4148 (0.4717)  loss_classifier: 0.1003 (0.1168)  loss_box_reg: 0.1598 (0.1914)  loss_objectne
ss: 0.0479 (0.0548)  loss_rpn_box_reg: 0.1045 (0.1086)  time: 0.1365  data: 0.0188  max mem: 2825
Epoch: [0]  [1000/1250]  eta: 0:00:39  lr: 0.005000  loss: 0.4814 (0.4734)  loss_classifier: 0.1300 (0.1175)  loss_box_reg: 0.2049 (0.1918)  loss_objectne
ss: 0.0483 (0.0554)  loss_rpn_box_reg: 0.0850 (0.1086)  time: 0.1430  data: 0.0252  max mem: 2825
Epoch: [0]  [1100/1250]  eta: 0:00:23  lr: 0.005000  loss: 0.3687 (0.4682)  loss_classifier: 0.0987 (0.1165)  loss_box_reg: 0.1621 (0.1900)  loss_objectne
ss: 0.0285 (0.0543)  loss_rpn_box_reg: 0.0576 (0.1074)  time: 0.1478  data: 0.0305  max mem: 2825
Epoch: [0]  [1200/1250]  eta: 0:00:07  lr: 0.005000  loss: 0.4245 (0.4694)  loss_classifier: 0.1031 (0.1171)  loss_box_reg: 0.1772 (0.1915)  loss_objectne
ss: 0.0387 (0.0540)  loss_rpn_box_reg: 0.0522 (0.1068)  time: 0.1430  data: 0.0251  max mem: 2825
Epoch: [0]  [1249/1250]  eta: 0:00:00  lr: 0.005000  loss: 0.5602 (0.4715)  loss_classifier: 0.1382 (0.1176)  loss_box_reg: 0.2604 (0.1925)  loss_objectne
ss: 0.0348 (0.0539)  loss_rpn_box_reg: 0.0483 (0.1073)  time: 0.1328  data: 0.0171  max mem: 2825
Epoch: [0] Total time: 0:03:16 (0.1571 s / it)
creating index...
index created!
Test:  [  0/441]  eta: 0:03:40  model_time: 0.0584 (0.0584)  evaluator_time: 0.0181 (0.0181)  time: 0.5010  data: 0.4188  max mem: 2825
Test:  [100/441]  eta: 0:00:29  model_time: 0.0442 (0.0446)  evaluator_time: 0.0125 (0.0267)  time: 0.0753  data: 0.0070  max mem: 2825
Test:  [200/441]  eta: 0:00:20  model_time: 0.0444 (0.0446)  evaluator_time: 0.0151 (0.0266)  time: 0.0799  data: 0.0073  max mem: 2825
Test:  [300/441]  eta: 0:00:11  model_time: 0.0442 (0.0445)  evaluator_time: 0.0112 (0.0262)  time: 0.0766  data: 0.0070  max mem: 2825
Test:  [400/441]  eta: 0:00:03  model_time: 0.0442 (0.0445)  evaluator_time: 0.0179 (0.0270)  time: 0.0930  data: 0.0070  max mem: 2825
Test:  [440/441]  eta: 0:00:00  model_time: 0.0444 (0.0445)  evaluator_time: 0.0113 (0.0267)  time: 0.0749  data: 0.0069  max mem: 2825
Test: Total time: 0:00:37 (0.0842 s / it)
Averaged stats: model_time: 0.0444 (0.0445)  evaluator_time: 0.0113 (0.0267)
Accumulating evaluation results...
```

```
DONE (t=0.32s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.360
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.635
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.365
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.084
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.370
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.591
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.073
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.324
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.452
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.228
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.473
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.640
Epoch: [1]  [   0/1250]  eta: 0:19:45  lr: 0.005000  loss: 0.3611 (0.3611)  loss_classifier: 0.1043 (0.1043)  loss_box_reg: 0.1523 (0.1523)  loss_objectne
ss: 0.0316 (0.0316)  loss_rpn_box_reg: 0.0729 (0.0729)  time: 0.9485  data: 0.8078  max mem: 2825
Epoch: [1]  [ 100/1250]  eta: 0:03:04  lr: 0.005000  loss: 0.3529 (0.4476)  loss_classifier: 0.1161 (0.1131)  loss_box_reg: 0.1651 (0.1822)  loss_objectne
ss: 0.0323 (0.0522)  loss_rpn_box_reg: 0.0359 (0.1001)  time: 0.1621  data: 0.0415  max mem: 2825
Epoch: [1]  [ 200/1250]  eta: 0:02:44  lr: 0.005000  loss: 0.3499 (0.4553)  loss_classifier: 0.0943 (0.1131)  loss_box_reg: 0.1619 (0.1849)  loss_objectne
ss: 0.0374 (0.0533)  loss_rpn_box_reg: 0.0431 (0.1040)  time: 0.1535  data: 0.0374  max mem: 2825
Epoch: [1]  [ 300/1250]  eta: 0:02:28  lr: 0.005000  loss: 0.3203 (0.4619)  loss_classifier: 0.0936 (0.1151)  loss_box_reg: 0.1507 (0.1904)  loss_objectne
ss: 0.0336 (0.0507)  loss_rpn_box_reg: 0.0601 (0.1056)  time: 0.1493  data: 0.0334  max mem: 2825
Epoch: [1]  [ 400/1250]  eta: 0:02:12  lr: 0.005000  loss: 0.5285 (0.4771)  loss_classifier: 0.1334 (0.1176)  loss_box_reg: 0.2123 (0.1940)  loss_objectne
ss: 0.0413 (0.0522)  loss_rpn_box_reg: 0.0771 (0.1132)  time: 0.1346  data: 0.0204  max mem: 2825
Epoch: [1]  [ 500/1250]  eta: 0:01:56  lr: 0.005000  loss: 0.3223 (0.4676)  loss_classifier: 0.0966 (0.1158)  loss_box_reg: 0.1394 (0.1909)  loss_objectne
ss: 0.0414 (0.0523)  loss_rpn_box_reg: 0.0317 (0.1086)  time: 0.1681  data: 0.0518  max mem: 2825
Epoch: [1]  [ 600/1250]  eta: 0:01:40  lr: 0.005000  loss: 0.3598 (0.4686)  loss_classifier: 0.0656 (0.1153)  loss_box_reg: 0.1169 (0.1886)  loss_objectne
ss: 0.0451 (0.0537)  loss_rpn_box_reg: 0.0576 (0.1110)  time: 0.1573  data: 0.0429  max mem: 2825
Epoch: [1]  [ 700/1250]  eta: 0:01:25  lr: 0.005000  loss: 0.4440 (0.4755)  loss_classifier: 0.1078 (0.1163)  loss_box_reg: 0.1873 (0.1907)  loss_objectne
ss: 0.0624 (0.0566)  loss_rpn_box_reg: 0.0457 (0.1120)  time: 0.1533  data: 0.0363  max mem: 2825
Epoch: [1]  [ 800/1250]  eta: 0:01:09  lr: 0.005000  loss: 0.3355 (0.4797)  loss_classifier: 0.0944 (0.1180)  loss_box_reg: 0.1943 (0.1928)  loss_objectne
ss: 0.0361 (0.0564)  loss_rpn_box_reg: 0.0217 (0.1125)  time: 0.1657  data: 0.0472  max mem: 2825
Epoch: [1]  [ 900/1250]  eta: 0:00:54  lr: 0.005000  loss: 0.3410 (0.4792)  loss_classifier: 0.0910 (0.1183)  loss_box_reg: 0.1777 (0.1935)  loss_objectne
ss: 0.0329 (0.0557)  loss_rpn_box_reg: 0.0557 (0.1117)  time: 0.1533  data: 0.0348  max mem: 2825
Epoch: [1]  [1000/1250]  eta: 0:00:38  lr: 0.005000  loss: 0.4587 (0.4811)  loss_classifier: 0.1110 (0.1181)  loss_box_reg: 0.2138 (0.1939)  loss_objectne
ss: 0.0465 (0.0561)  loss_rpn_box_reg: 0.0636 (0.1130)  time: 0.1571  data: 0.0427  max mem: 2825
Epoch: [1]  [1100/1250]  eta: 0:00:23  lr: 0.005000  loss: 0.4956 (0.4855)  loss_classifier: 0.1331 (0.1191)  loss_box_reg: 0.1901 (0.1956)  loss_objectne
ss: 0.0469 (0.0570)  loss_rpn_box_reg: 0.0418 (0.1138)  time: 0.1502  data: 0.0348  max mem: 2825
Epoch: [1]  [1200/1250]  eta: 0:00:07  lr: 0.005000  loss: 0.4017 (0.4838)  loss_classifier: 0.1058 (0.1188)  loss_box_reg: 0.2217 (0.1954)  loss_objectne
ss: 0.0407 (0.0569)  loss_rpn_box_reg: 0.0278 (0.1127)  time: 0.1449  data: 0.0286  max mem: 2825
Epoch: [1]  [1249/1250]  eta: 0:00:00  lr: 0.005000  loss: 0.3131 (0.4809)  loss_classifier: 0.0849 (0.1186)  loss_box_reg: 0.1584 (0.1948)  loss_objectne
ss: 0.0318 (0.0563)  loss_rpn_box_reg: 0.0394 (0.1112)  time: 0.1548  data: 0.0403  max mem: 2825
Epoch: [1] Total time: 0:03:12 (0.1538 s / it)
creating index...
index created!
Test:  [  0/441]  eta: 0:03:32  model_time: 0.0492 (0.0492)  evaluator_time: 0.0099 (0.0099)  time: 0.4808  data: 0.4165  max mem: 2825
Test:  [100/441]  eta: 0:00:30  model_time: 0.0442 (0.0448)  evaluator_time: 0.0131 (0.0283)  time: 0.0778  data: 0.0068  max mem: 2825
Test:  [200/441]  eta: 0:00:20  model_time: 0.0445 (0.0447)  evaluator_time: 0.0158 (0.0284)  time: 0.0790  data: 0.0069  max mem: 2825
Test:  [300/441]  eta: 0:00:12  model_time: 0.0445 (0.0446)  evaluator_time: 0.0123 (0.0282)  time: 0.0787  data: 0.0068  max mem: 2825
Test:  [400/441]  eta: 0:00:03  model_time: 0.0440 (0.0445)  evaluator_time: 0.0239 (0.0290)  time: 0.0933  data: 0.0067  max mem: 2825
Test:  [440/441]  eta: 0:00:00  model_time: 0.0441 (0.0445)  evaluator_time: 0.0151 (0.0288)  time: 0.0776  data: 0.0066  max mem: 2825
Test: Total time: 0:00:37 (0.0861 s / it)
Averaged stats: model_time: 0.0441 (0.0445)  evaluator_time: 0.0151 (0.0288)
Accumulating evaluation results...
DONE (t=0.38s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.370
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.638
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.375
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.085
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.379
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.604
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.074
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.331
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.462
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.247
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.480
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.649
Epoch: [2]  [   0/1250]  eta: 0:25:34  lr: 0.005000  loss: 0.1864 (0.1864)  loss_classifier: 0.0679 (0.0679)  loss_box_reg: 0.0860 (0.0860)  loss_objectne
ss: 0.0291 (0.0291)  loss_rpn_box_reg: 0.0033 (0.0033)  time: 1.2273  data: 1.0951  max mem: 2825
Epoch: [2]  [ 100/1250]  eta: 0:03:05  lr: 0.005000  loss: 0.4600 (0.4728)  loss_classifier: 0.1227 (0.1218)  loss_box_reg: 0.2052 (0.1959)  loss_objectne
ss: 0.0508 (0.0576)  loss_rpn_box_reg: 0.0632 (0.0975)  time: 0.1426  data: 0.0263  max mem: 2825
Epoch: [2]  [ 200/1250]  eta: 0:02:44  lr: 0.005000  loss: 0.5920 (0.4831)  loss_classifier: 0.1121 (0.1202)  loss_box_reg: 0.1777 (0.1955)  loss_objectne
ss: 0.0494 (0.0544)  loss_rpn_box_reg: 0.1155 (0.1130)  time: 0.1699  data: 0.0532  max mem: 2825
Epoch: [2]  [ 300/1250]  eta: 0:02:27  lr: 0.005000  loss: 0.4462 (0.4722)  loss_classifier: 0.1180 (0.1175)  loss_box_reg: 0.2047 (0.1916)  loss_objectne
ss: 0.0355 (0.0536)  loss_rpn_box_reg: 0.0742 (0.1095)  time: 0.1577  data: 0.0419  max mem: 2825
Epoch: [2]  [ 400/1250]  eta: 0:02:12  lr: 0.005000  loss: 0.3847 (0.4771)  loss_classifier: 0.0966 (0.1171)  loss_box_reg: 0.1390 (0.1921)  loss_objectne
ss: 0.0410 (0.0544)  loss_rpn_box_reg: 0.0347 (0.1136)  time: 0.1499  data: 0.0320  max mem: 2825
Epoch: [2]  [ 500/1250]  eta: 0:01:56  lr: 0.005000  loss: 0.3188 (0.4713)  loss_classifier: 0.0845 (0.1156)  loss_box_reg: 0.1432 (0.1874)  loss_objectne
ss: 0.0346 (0.0549)  loss_rpn_box_reg: 0.0329 (0.1134)  time: 0.1467  data: 0.0290  max mem: 2825
Epoch: [2]  [ 600/1250]  eta: 0:01:41  lr: 0.005000  loss: 0.5369 (0.4704)  loss_classifier: 0.0899 (0.1158)  loss_box_reg: 0.1839 (0.1882)  loss_objectne
ss: 0.0331 (0.0547)  loss_rpn_box_reg: 0.0403 (0.1116)  time: 0.1658  data: 0.0489  max mem: 2825
Epoch: [2]  [ 700/1250]  eta: 0:01:25  lr: 0.005000  loss: 0.4434 (0.4732)  loss_classifier: 0.1164 (0.1165)  loss_box_reg: 0.2042 (0.1893)  loss_objectne
ss: 0.0440 (0.0567)  loss_rpn_box_reg: 0.0576 (0.1107)  time: 0.1638  data: 0.0482  max mem: 2825
Epoch: [2]  [ 800/1250]  eta: 0:01:09  lr: 0.005000  loss: 0.4283 (0.4766)  loss_classifier: 0.0998 (0.1168)  loss_box_reg: 0.1749 (0.1902)  loss_objectne
ss: 0.0457 (0.0577)  loss_rpn_box_reg: 0.0680 (0.1118)  time: 0.1736  data: 0.0598  max mem: 2825
Epoch: [2]  [ 900/1250]  eta: 0:00:54  lr: 0.005000  loss: 0.4589 (0.4797)  loss_classifier: 0.1148 (0.1178)  loss_box_reg: 0.2055 (0.1918)  loss_objectne
ss: 0.0371 (0.0575)  loss_rpn_box_reg: 0.0724 (0.1125)  time: 0.1426  data: 0.0258  max mem: 2825
Epoch: [2]  [1000/1250]  eta: 0:00:38  lr: 0.005000  loss: 0.4102 (0.4797)  loss_classifier: 0.1056 (0.1182)  loss_box_reg: 0.1940 (0.1924)  loss_objectne
ss: 0.0330 (0.0569)  loss_rpn_box_reg: 0.0777 (0.1123)  time: 0.1405  data: 0.0266  max mem: 2825
Epoch: [2]  [1100/1250]  eta: 0:00:23  lr: 0.005000  loss: 0.4175 (0.4834)  loss_classifier: 0.1142 (0.1193)  loss_box_reg: 0.1874 (0.1943)  loss_objectne
ss: 0.0365 (0.0567)  loss_rpn_box_reg: 0.0688 (0.1131)  time: 0.1513  data: 0.0338  max mem: 2825
Epoch: [2]  [1200/1250]  eta: 0:00:07  lr: 0.005000  loss: 0.4406 (0.4799)  loss_classifier: 0.1016 (0.1184)  loss_box_reg: 0.1671 (0.1930)  loss_objectne
ss: 0.0568 (0.0563)  loss_rpn_box_reg: 0.1021 (0.1122)  time: 0.1550  data: 0.0397  max mem: 2825
Epoch: [2]  [1249/1250]  eta: 0:00:00  lr: 0.005000  loss: 0.4080 (0.4786)  loss_classifier: 0.1120 (0.1181)  loss_box_reg: 0.2328 (0.1926)  loss_objectne
ss: 0.0397 (0.0561)  loss_rpn_box_reg: 0.0781 (0.1118)  time: 0.1526  data: 0.0384  max mem: 2825
Epoch: [2] Total time: 0:03:13 (0.1544 s / it)
creating index...
index created!
Test:  [  0/441]  eta: 0:03:29  model_time: 0.0517 (0.0517)  evaluator_time: 0.0095 (0.0095)  time: 0.4755  data: 0.4086  max mem: 2825
Test:  [100/441]  eta: 0:00:28  model_time: 0.0442 (0.0445)  evaluator_time: 0.0091 (0.0235)  time: 0.0738  data: 0.0069  max mem: 2825
Test:  [200/441]  eta: 0:00:19  model_time: 0.0442 (0.0445)  evaluator_time: 0.0100 (0.0236)  time: 0.0740  data: 0.0068  max mem: 2825
Test:  [300/441]  eta: 0:00:11  model_time: 0.0442 (0.0445)  evaluator_time: 0.0076 (0.0235)  time: 0.0745  data: 0.0070  max mem: 2825
Test:  [400/441]  eta: 0:00:03  model_time: 0.0441 (0.0445)  evaluator_time: 0.0159 (0.0244)  time: 0.0897  data: 0.0070  max mem: 2825
Test:  [440/441]  eta: 0:00:00  model_time: 0.0438 (0.0444)  evaluator_time: 0.0084 (0.0240)  time: 0.0711  data: 0.0066  max mem: 2825
Test: Total time: 0:00:35 (0.0814 s / it)
Averaged stats: model_time: 0.0438 (0.0444)  evaluator_time: 0.0084 (0.0240)
Accumulating evaluation results...
DONE (t=0.28s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.364
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.640
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.369
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.086
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.370
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.598
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.073
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.325
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.454
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.224
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.476
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.646
Epoch: [3]  [   0/1250]  eta: 0:14:43  lr: 0.000500  loss: 0.4647 (0.4647)  loss_classifier: 0.1011 (0.1011)  loss_box_reg: 0.2222 (0.2222)  loss_objectne
ss: 0.0327 (0.0327)  loss_rpn_box_reg: 0.1087 (0.1087)  time: 0.7070  data: 0.5695  max mem: 2825
```

```
Epoch: [3]  [ 100/1250]  eta: 0:03:13  lr: 0.000500  loss: 0.3514 (0.4346)  loss_classifier: 0.0990 (0.1067)  loss_box_reg: 0.1594 (0.1798)  loss_objectne
ss: 0.0350 (0.0498)  loss_rpn_box_reg: 0.0633 (0.0983)  time: 0.1541  data: 0.0342  max mem: 2825
Epoch: [3]  [ 200/1250]  eta: 0:02:49  lr: 0.000500  loss: 0.2380 (0.4226)  loss_classifier: 0.0621 (0.1032)  loss_box_reg: 0.1191 (0.1717)  loss_objectne
ss: 0.0281 (0.0495)  loss_rpn_box_reg: 0.0228 (0.0983)  time: 0.1552  data: 0.0387  max mem: 2825
Epoch: [3]  [ 300/1250]  eta: 0:02:32  lr: 0.000500  loss: 0.3141 (0.4500)  loss_classifier: 0.0983 (0.1091)  loss_box_reg: 0.1602 (0.1802)  loss_objectne
ss: 0.0348 (0.0558)  loss_rpn_box_reg: 0.0581 (0.1049)  time: 0.1618  data: 0.0468  max mem: 2825
Epoch: [3]  [ 400/1250]  eta: 0:02:16  lr: 0.000500  loss: 0.3918 (0.4529)  loss_classifier: 0.0956 (0.1101)  loss_box_reg: 0.1576 (0.1816)  loss_objectne
ss: 0.0373 (0.0556)  loss_rpn_box_reg: 0.0808 (0.1057)  time: 0.1680  data: 0.0510  max mem: 2825
Epoch: [3]  [ 500/1250]  eta: 0:01:59  lr: 0.000500  loss: 0.4215 (0.4577)  loss_classifier: 0.1152 (0.1119)  loss_box_reg: 0.1924 (0.1835)  loss_objectne
ss: 0.0430 (0.0552)  loss_rpn_box_reg: 0.1154 (0.1071)  time: 0.1631  data: 0.0481  max mem: 2825
Epoch: [3]  [ 600/1250]  eta: 0:01:43  lr: 0.000500  loss: 0.2751 (0.4612)  loss_classifier: 0.0746 (0.1136)  loss_box_reg: 0.1128 (0.1848)  loss_objectne
ss: 0.0363 (0.0558)  loss_rpn_box_reg: 0.0111 (0.1070)  time: 0.1733  data: 0.0595  max mem: 2825
Epoch: [3]  [ 700/1250]  eta: 0:01:27  lr: 0.000500  loss: 0.2787 (0.4522)  loss_classifier: 0.0766 (0.1118)  loss_box_reg: 0.1673 (0.1826)  loss_objectne
ss: 0.0324 (0.0541)  loss_rpn_box_reg: 0.0349 (0.1037)  time: 0.1605  data: 0.0445  max mem: 2825
Epoch: [3]  [ 800/1250]  eta: 0:01:11  lr: 0.000500  loss: 0.3612 (0.4541)  loss_classifier: 0.0863 (0.1124)  loss_box_reg: 0.1569 (0.1840)  loss_objectne
ss: 0.0408 (0.0535)  loss_rpn_box_reg: 0.0486 (0.1042)  time: 0.1775  data: 0.0608  max mem: 2825
Epoch: [3]  [ 900/1250]  eta: 0:00:55  lr: 0.000500  loss: 0.2930 (0.4504)  loss_classifier: 0.0834 (0.1115)  loss_box_reg: 0.1076 (0.1823)  loss_objectne
ss: 0.0406 (0.0526)  loss_rpn_box_reg: 0.0480 (0.1040)  time: 0.1592  data: 0.0428  max mem: 2825
Epoch: [3]  [1000/1250]  eta: 0:00:39  lr: 0.000500  loss: 0.4020 (0.4518)  loss_classifier: 0.0925 (0.1116)  loss_box_reg: 0.1668 (0.1830)  loss_objectne
ss: 0.0330 (0.0527)  loss_rpn_box_reg: 0.0666 (0.1046)  time: 0.1772  data: 0.0611  max mem: 2825
Epoch: [3]  [1100/1250]  eta: 0:00:23  lr: 0.000500  loss: 0.3073 (0.4509)  loss_classifier: 0.0803 (0.1118)  loss_box_reg: 0.1206 (0.1829)  loss_objectne
ss: 0.0296 (0.0524)  loss_rpn_box_reg: 0.0368 (0.1038)  time: 0.1427  data: 0.0285  max mem: 2825
Epoch: [3]  [1200/1250]  eta: 0:00:07  lr: 0.000500  loss: 0.3937 (0.4495)  loss_classifier: 0.1290 (0.1121)  loss_box_reg: 0.1822 (0.1830)  loss_objectne
ss: 0.0427 (0.0521)  loss_rpn_box_reg: 0.0670 (0.1023)  time: 0.1705  data: 0.0526  max mem: 2825
Epoch: [3]  [1249/1250]  eta: 0:00:00  lr: 0.000500  loss: 0.3427 (0.4477)  loss_classifier: 0.0739 (0.1120)  loss_box_reg: 0.1538 (0.1827)  loss_objectne
ss: 0.0363 (0.0516)  loss_rpn_box_reg: 0.0538 (0.1014)  time: 0.1373  data: 0.0232  max mem: 2825
Epoch: [3] Total time: 0:03:18 (0.1584 s / it)
creating index...
index created!
Test:  [  0/441]  eta: 0:03:36  model_time: 0.0498 (0.0498)  evaluator_time: 0.0129 (0.0129)  time: 0.4907  data: 0.4226  max mem: 2825
Test:  [100/441]  eta: 0:00:29  model_time: 0.0442 (0.0446)  evaluator_time: 0.0089 (0.0247)  time: 0.0733  data: 0.0067  max mem: 2825
Test:  [200/441]  eta: 0:00:19  model_time: 0.0441 (0.0445)  evaluator_time: 0.0126 (0.0246)  time: 0.0772  data: 0.0071  max mem: 2825
Test:  [300/441]  eta: 0:00:11  model_time: 0.0445 (0.0446)  evaluator_time: 0.0098 (0.0246)  time: 0.0769  data: 0.0068  max mem: 2825
Test:  [400/441]  eta: 0:00:03  model_time: 0.0440 (0.0445)  evaluator_time: 0.0155 (0.0253)  time: 0.0882  data: 0.0067  max mem: 2825
Test:  [440/441]  eta: 0:00:00  model_time: 0.0439 (0.0445)  evaluator_time: 0.0099 (0.0250)  time: 0.0729  data: 0.0066  max mem: 2825
Test: Total time: 0:00:36 (0.0824 s / it)
Averaged stats: model_time: 0.0439 (0.0445)  evaluator_time: 0.0099 (0.0250)
Accumulating evaluation results...
DONE (t=0.29s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.381
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.650
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.390
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.092
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.390
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.620
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.075
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.335
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.471
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.239
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.494
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.665
Epoch: [4]  [   0/1250]  eta: 0:19:31  lr: 0.000500  loss: 0.3608 (0.3608)  loss_classifier: 0.0803 (0.0803)  loss_box_reg: 0.1769 (0.1769)  loss_objectne
ss: 0.0194 (0.0194)  loss_rpn_box_reg: 0.0842 (0.0842)  time: 0.9373  data: 0.8014  max mem: 2825
Epoch: [4]  [ 100/1250]  eta: 0:03:08  lr: 0.000500  loss: 0.3308 (0.4325)  loss_classifier: 0.1027 (0.1099)  loss_box_reg: 0.1610 (0.1816)  loss_objectne
ss: 0.0353 (0.0477)  loss_rpn_box_reg: 0.0512 (0.0933)  time: 0.1566  data: 0.0376  max mem: 2825
Epoch: [4]  [ 200/1250]  eta: 0:02:52  lr: 0.000500  loss: 0.3396 (0.4275)  loss_classifier: 0.0932 (0.1104)  loss_box_reg: 0.1494 (0.1810)  loss_objectne
ss: 0.0392 (0.0483)  loss_rpn_box_reg: 0.0214 (0.0878)  time: 0.1463  data: 0.0276  max mem: 2825
Epoch: [4]  [ 300/1250]  eta: 0:02:32  lr: 0.000500  loss: 0.4136 (0.4250)  loss_classifier: 0.1136 (0.1101)  loss_box_reg: 0.1813 (0.1789)  loss_objectne
ss: 0.0374 (0.0482)  loss_rpn_box_reg: 0.0901 (0.0877)  time: 0.1519  data: 0.0354  max mem: 2825
Epoch: [4]  [ 400/1250]  eta: 0:02:15  lr: 0.000500  loss: 0.3371 (0.4309)  loss_classifier: 0.0919 (0.1113)  loss_box_reg: 0.1593 (0.1812)  loss_objectne
ss: 0.0334 (0.0482)  loss_rpn_box_reg: 0.0506 (0.0902)  time: 0.1710  data: 0.0561  max mem: 2825
Epoch: [4]  [ 500/1250]  eta: 0:01:58  lr: 0.000500  loss: 0.2695 (0.4270)  loss_classifier: 0.0774 (0.1101)  loss_box_reg: 0.1389 (0.1813)  loss_objectne
ss: 0.0336 (0.0486)  loss_rpn_box_reg: 0.0164 (0.0871)  time: 0.1497  data: 0.0317  max mem: 2825
Epoch: [4]  [ 600/1250]  eta: 0:01:40  lr: 0.000500  loss: 0.4423 (0.4322)  loss_classifier: 0.0824 (0.1109)  loss_box_reg: 0.1402 (0.1816)  loss_objectne
ss: 0.0389 (0.0483)  loss_rpn_box_reg: 0.1066 (0.0915)  time: 0.1521  data: 0.0328  max mem: 2825
Epoch: [4]  [ 700/1250]  eta: 0:01:25  lr: 0.000500  loss: 0.4685 (0.4386)  loss_classifier: 0.1139 (0.1120)  loss_box_reg: 0.1640 (0.1830)  loss_objectne
ss: 0.0441 (0.0491)  loss_rpn_box_reg: 0.0817 (0.0945)  time: 0.1430  data: 0.0265  max mem: 2825
Epoch: [4]  [ 800/1250]  eta: 0:01:10  lr: 0.000500  loss: 0.4155 (0.4367)  loss_classifier: 0.1080 (0.1109)  loss_box_reg: 0.1707 (0.1811)  loss_objectne
ss: 0.0376 (0.0491)  loss_rpn_box_reg: 0.0835 (0.0956)  time: 0.1683  data: 0.0506  max mem: 2825
Epoch: [4]  [ 900/1250]  eta: 0:00:54  lr: 0.000500  loss: 0.5011 (0.4412)  loss_classifier: 0.1016 (0.1110)  loss_box_reg: 0.1889 (0.1812)  loss_objectne
ss: 0.0431 (0.0499)  loss_rpn_box_reg: 0.1085 (0.0991)  time: 0.1633  data: 0.0477  max mem: 2825
Epoch: [4]  [1000/1250]  eta: 0:00:38  lr: 0.000500  loss: 0.4841 (0.4420)  loss_classifier: 0.1003 (0.1108)  loss_box_reg: 0.1763 (0.1802)  loss_objectne
ss: 0.0467 (0.0515)  loss_rpn_box_reg: 0.0746 (0.0995)  time: 0.1756  data: 0.0611  max mem: 2825
Epoch: [4]  [1100/1250]  eta: 0:00:23  lr: 0.000500  loss: 0.3846 (0.4435)  loss_classifier: 0.0959 (0.1113)  loss_box_reg: 0.1614 (0.1807)  loss_objectne
ss: 0.0418 (0.0517)  loss_rpn_box_reg: 0.0480 (0.0998)  time: 0.1431  data: 0.0292  max mem: 2825
Epoch: [4]  [1200/1250]  eta: 0:00:07  lr: 0.000500  loss: 0.3236 (0.4418)  loss_classifier: 0.0832 (0.1108)  loss_box_reg: 0.1031 (0.1800)  loss_objectne
ss: 0.0302 (0.0515)  loss_rpn_box_reg: 0.0306 (0.0996)  time: 0.1466  data: 0.0336  max mem: 2825
Epoch: [4]  [1249/1250]  eta: 0:00:00  lr: 0.000500  loss: 0.4811 (0.4426)  loss_classifier: 0.1242 (0.1111)  loss_box_reg: 0.2013 (0.1803)  loss_objectne
ss: 0.0321 (0.0512)  loss_rpn_box_reg: 0.0536 (0.1000)  time: 0.1496  data: 0.0349  max mem: 2825
Epoch: [4] Total time: 0:03:15 (0.1560 s / it)
creating index...
index created!
Test:  [  0/441]  eta: 0:03:34  model_time: 0.0492 (0.0492)  evaluator_time: 0.0090 (0.0090)  time: 0.4863  data: 0.4231  max mem: 2825
Test:  [100/441]  eta: 0:00:28  model_time: 0.0441 (0.0445)  evaluator_time: 0.0081 (0.0240)  time: 0.0729  data: 0.0067  max mem: 2825
Test:  [200/441]  eta: 0:00:19  model_time: 0.0441 (0.0445)  evaluator_time: 0.0112 (0.0241)  time: 0.0732  data: 0.0067  max mem: 2825
Test:  [300/441]  eta: 0:00:11  model_time: 0.0440 (0.0444)  evaluator_time: 0.0108 (0.0240)  time: 0.0753  data: 0.0067  max mem: 2825
Test:  [400/441]  eta: 0:00:03  model_time: 0.0442 (0.0445)  evaluator_time: 0.0153 (0.0247)  time: 0.0889  data: 0.0068  max mem: 2825
Test:  [440/441]  eta: 0:00:00  model_time: 0.0439 (0.0445)  evaluator_time: 0.0105 (0.0244)  time: 0.0722  data: 0.0065  max mem: 2825
Test: Total time: 0:00:36 (0.0816 s / it)
Averaged stats: model_time: 0.0439 (0.0445)  evaluator_time: 0.0105 (0.0244)
Accumulating evaluation results...
DONE (t=0.28s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.378
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.646
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.390
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.093
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.385
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.619
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.076
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.337
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.468
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.239
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.487
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.666
Epoch: [5]  [   0/1250]  eta: 0:14:20  lr: 0.000500  loss: 0.2800 (0.2800)  loss_classifier: 0.0856 (0.0856)  loss_box_reg: 0.1155 (0.1155)  loss_objectne
ss: 0.0405 (0.0405)  loss_rpn_box_reg: 0.0384 (0.0384)  time: 0.6880  data: 0.5610  max mem: 2825
Epoch: [5]  [ 100/1250]  eta: 0:03:12  lr: 0.000500  loss: 0.3340 (0.4021)  loss_classifier: 0.0860 (0.1036)  loss_box_reg: 0.1466 (0.1725)  loss_objectne
ss: 0.0290 (0.0453)  loss_rpn_box_reg: 0.0362 (0.0807)  time: 0.1410  data: 0.0248  max mem: 2825
Epoch: [5]  [ 200/1250]  eta: 0:02:48  lr: 0.000500  loss: 0.4438 (0.4086)  loss_classifier: 0.1056 (0.1035)  loss_box_reg: 0.1552 (0.1719)  loss_objectne
ss: 0.0412 (0.0470)  loss_rpn_box_reg: 0.0862 (0.0863)  time: 0.1541  data: 0.0371  max mem: 2825
Epoch: [5]  [ 300/1250]  eta: 0:02:30  lr: 0.000500  loss: 0.4100 (0.4281)  loss_classifier: 0.1176 (0.1082)  loss_box_reg: 0.2111 (0.1799)  loss_objectne
ss: 0.0426 (0.0484)  loss_rpn_box_reg: 0.0507 (0.0915)  time: 0.1536  data: 0.0385  max mem: 2825
Epoch: [5]  [ 400/1250]  eta: 0:02:13  lr: 0.000500  loss: 0.3412 (0.4399)  loss_classifier: 0.0945 (0.1114)  loss_box_reg: 0.1591 (0.1831)  loss_objectne
ss: 0.0384 (0.0496)  loss_rpn_box_reg: 0.0725 (0.0958)  time: 0.1406  data: 0.0208  max mem: 2825
Epoch: [5]  [ 500/1250]  eta: 0:01:56  lr: 0.000500  loss: 0.2909 (0.4361)  loss_classifier: 0.0872 (0.1105)  loss_box_reg: 0.1363 (0.1813)  loss_objectne
ss: 0.0347 (0.0489)  loss_rpn_box_reg: 0.0211 (0.0955)  time: 0.1400  data: 0.0239  max mem: 2825
Epoch: [5]  [ 600/1250]  eta: 0:01:41  lr: 0.000500  loss: 0.4431 (0.4445)  loss_classifier: 0.0952 (0.1112)  loss_box_reg: 0.1813 (0.1823)  loss_objectne
ss: 0.0434 (0.0491)  loss_rpn_box_reg: 0.0846 (0.1019)  time: 0.1778  data: 0.0605  max mem: 2825
Epoch: [5]  [ 700/1250]  eta: 0:01:26  lr: 0.000500  loss: 0.3291 (0.4356)  loss_classifier: 0.0879 (0.1098)  loss_box_reg: 0.1476 (0.1800)  loss_objectne
ss: 0.0286 (0.0480)  loss_rpn_box_reg: 0.0316 (0.0979)  time: 0.1631  data: 0.0493  max mem: 2825
Epoch: [5]  [ 800/1250]  eta: 0:01:10  lr: 0.000500  loss: 0.4347 (0.4339)  loss_classifier: 0.1138 (0.1099)  loss_box_reg: 0.1949 (0.1804)  loss_objectne
ss: 0.0408 (0.0482)  loss_rpn_box_reg: 0.0474 (0.0954)  time: 0.1543  data: 0.0377  max mem: 2825
```

```
Epoch: [5]  [ 900/1250]  eta: 0:00:54  lr: 0.000500  loss: 0.5170 (0.4336)  loss_classifier: 0.1092 (0.1097)  loss_box_reg: 0.2305 (0.1810)  loss_objectne
ss: 0.0440 (0.0476)  loss_rpn_box_reg: 0.0984 (0.0953)  time: 0.1435  data: 0.0246  max mem: 2825
Epoch: [5]  [1000/1250]  eta: 0:00:39  lr: 0.000500  loss: 0.4367 (0.4345)  loss_classifier: 0.1198 (0.1103)  loss_box_reg: 0.2085 (0.1815)  loss_objectne
ss: 0.0436 (0.0478)  loss_rpn_box_reg: 0.0502 (0.0949)  time: 0.1808  data: 0.0646  max mem: 2825
Epoch: [5]  [1100/1250]  eta: 0:00:23  lr: 0.000500  loss: 0.2888 (0.4359)  loss_classifier: 0.0870 (0.1102)  loss_box_reg: 0.0894 (0.1807)  loss_objectne
ss: 0.0278 (0.0490)  loss_rpn_box_reg: 0.0283 (0.0960)  time: 0.1585  data: 0.0415  max mem: 2825
Epoch: [5]  [1200/1250]  eta: 0:00:07  lr: 0.000500  loss: 0.4012 (0.4373)  loss_classifier: 0.1011 (0.1100)  loss_box_reg: 0.1849 (0.1806)  loss_objectne
ss: 0.0380 (0.0495)  loss_rpn_box_reg: 0.0679 (0.0972)  time: 0.1580  data: 0.0409  max mem: 2825
Epoch: [5]  [1249/1250]  eta: 0:00:00  lr: 0.000500  loss: 0.4099 (0.4411)  loss_classifier: 0.1083 (0.1105)  loss_box_reg: 0.1903 (0.1811)  loss_objectne
ss: 0.0414 (0.0504)  loss_rpn_box_reg: 0.0689 (0.0992)  time: 0.1782  data: 0.0643  max mem: 2825
Epoch: [5] Total time: 0:03:15 (0.1565 s / it)
creating index...
index created!
Test:  [  0/441]  eta: 0:04:01  model_time: 0.0627 (0.0627)  evaluator_time: 0.0117 (0.0117)  time: 0.5485  data: 0.4683  max mem: 2825
Test:  [100/441]  eta: 0:00:29  model_time: 0.0446 (0.0449)  evaluator_time: 0.0079 (0.0238)  time: 0.0737  data: 0.0068  max mem: 2825
Test:  [200/441]  eta: 0:00:19  model_time: 0.0441 (0.0447)  evaluator_time: 0.0117 (0.0241)  time: 0.0746  data: 0.0071  max mem: 2825
Test:  [300/441]  eta: 0:00:11  model_time: 0.0445 (0.0446)  evaluator_time: 0.0086 (0.0239)  time: 0.0746  data: 0.0070  max mem: 2825
Test:  [400/441]  eta: 0:00:03  model_time: 0.0441 (0.0445)  evaluator_time: 0.0135 (0.0249)  time: 0.0877  data: 0.0071  max mem: 2825
Test:  [440/441]  eta: 0:00:00  model_time: 0.0440 (0.0445)  evaluator_time: 0.0110 (0.0246)  time: 0.0737  data: 0.0069  max mem: 2825
Test: Total time: 0:00:36 (0.0823 s / it)
Averaged stats: model_time: 0.0440 (0.0445)  evaluator_time: 0.0110 (0.0246)
Accumulating evaluation results...
DONE (t=0.28s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.381
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.648
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.388
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.094
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.386
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.623
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.076
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.337
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.469
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.245
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.487
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.667
Epoch: [6]  [   0/1250]  eta: 0:19:49  lr: 0.000050  loss: 0.2434 (0.2434)  loss_classifier: 0.0577 (0.0577)  loss_box_reg: 0.0821 (0.0821)  loss_objectne
ss: 0.0162 (0.0162)  loss_rpn_box_reg: 0.0874 (0.0874)  time: 0.9512  data: 0.8294  max mem: 2825
Epoch: [6]  [ 100/1250]  eta: 0:03:06  lr: 0.000050  loss: 0.3851 (0.4627)  loss_classifier: 0.0964 (0.1099)  loss_box_reg: 0.1630 (0.1814)  loss_objectne
ss: 0.0501 (0.0558)  loss_rpn_box_reg: 0.0854 (0.1156)  time: 0.1659  data: 0.0496  max mem: 2825
Epoch: [6]  [ 200/1250]  eta: 0:02:49  lr: 0.000050  loss: 0.3825 (0.4477)  loss_classifier: 0.1014 (0.1067)  loss_box_reg: 0.1437 (0.1728)  loss_objectne
ss: 0.0416 (0.0537)  loss_rpn_box_reg: 0.0444 (0.1146)  time: 0.1390  data: 0.0220  max mem: 2825
Epoch: [6]  [ 300/1250]  eta: 0:02:29  lr: 0.000050  loss: 0.3415 (0.4524)  loss_classifier: 0.0941 (0.1097)  loss_box_reg: 0.2061 (0.1792)  loss_objectne
ss: 0.0438 (0.0535)  loss_rpn_box_reg: 0.0579 (0.1099)  time: 0.1500  data: 0.0333  max mem: 2825
Epoch: [6]  [ 400/1250]  eta: 0:02:13  lr: 0.000050  loss: 0.3325 (0.4481)  loss_classifier: 0.1015 (0.1095)  loss_box_reg: 0.1180 (0.1795)  loss_objectne
ss: 0.0389 (0.0519)  loss_rpn_box_reg: 0.0233 (0.1072)  time: 0.1394  data: 0.0251  max mem: 2825
Epoch: [6]  [ 500/1250]  eta: 0:01:57  lr: 0.000050  loss: 0.3403 (0.4412)  loss_classifier: 0.0958 (0.1090)  loss_box_reg: 0.1439 (0.1777)  loss_objectne
ss: 0.0307 (0.0499)  loss_rpn_box_reg: 0.0261 (0.1045)  time: 0.1605  data: 0.0472  max mem: 2825
Epoch: [6]  [ 600/1250]  eta: 0:01:40  lr: 0.000050  loss: 0.4551 (0.4425)  loss_classifier: 0.1004 (0.1091)  loss_box_reg: 0.1592 (0.1786)  loss_objectne
ss: 0.0411 (0.0508)  loss_rpn_box_reg: 0.0727 (0.1040)  time: 0.1434  data: 0.0283  max mem: 2825
Epoch: [6]  [ 700/1250]  eta: 0:01:24  lr: 0.000050  loss: 0.2724 (0.4367)  loss_classifier: 0.0705 (0.1085)  loss_box_reg: 0.1272 (0.1775)  loss_objectne
ss: 0.0337 (0.0495)  loss_rpn_box_reg: 0.0236 (0.1011)  time: 0.1437  data: 0.0275  max mem: 2825
Epoch: [6]  [ 800/1250]  eta: 0:01:09  lr: 0.000050  loss: 0.4782 (0.4367)  loss_classifier: 0.1270 (0.1087)  loss_box_reg: 0.1898 (0.1776)  loss_objectne
ss: 0.0399 (0.0498)  loss_rpn_box_reg: 0.0994 (0.1007)  time: 0.1597  data: 0.0433  max mem: 2825
Epoch: [6]  [ 900/1250]  eta: 0:00:54  lr: 0.000050  loss: 0.3568 (0.4373)  loss_classifier: 0.1022 (0.1089)  loss_box_reg: 0.1761 (0.1779)  loss_objectne
ss: 0.0304 (0.0504)  loss_rpn_box_reg: 0.0357 (0.1001)  time: 0.1530  data: 0.0385  max mem: 2825
Epoch: [6]  [1000/1250]  eta: 0:00:38  lr: 0.000050  loss: 0.4296 (0.4388)  loss_classifier: 0.1121 (0.1093)  loss_box_reg: 0.2011 (0.1794)  loss_objectne
ss: 0.0388 (0.0500)  loss_rpn_box_reg: 0.0558 (0.1001)  time: 0.1504  data: 0.0337  max mem: 2825
Epoch: [6]  [1100/1250]  eta: 0:00:23  lr: 0.000050  loss: 0.4201 (0.4403)  loss_classifier: 0.1148 (0.1095)  loss_box_reg: 0.1730 (0.1795)  loss_objectne
ss: 0.0403 (0.0505)  loss_rpn_box_reg: 0.0516 (0.1008)  time: 0.1642  data: 0.0506  max mem: 2825
Epoch: [6]  [1200/1250]  eta: 0:00:07  lr: 0.000050  loss: 0.3240 (0.4374)  loss_classifier: 0.0830 (0.1091)  loss_box_reg: 0.1488 (0.1792)  loss_objectne
ss: 0.0393 (0.0504)  loss_rpn_box_reg: 0.0321 (0.0987)  time: 0.1757  data: 0.0594  max mem: 2825
Epoch: [6]  [1249/1250]  eta: 0:00:00  lr: 0.000050  loss: 0.3240 (0.4369)  loss_classifier: 0.0923 (0.1092)  loss_box_reg: 0.1563 (0.1789)  loss_objectne
ss: 0.0346 (0.0504)  loss_rpn_box_reg: 0.0447 (0.0985)  time: 0.1656  data: 0.0516  max mem: 2825
Epoch: [6] Total time: 0:03:13 (0.1551 s / it)
creating index...
index created!
Test:  [  0/441]  eta: 0:03:34  model_time: 0.0601 (0.0601)  evaluator_time: 0.0153 (0.0153)  time: 0.4855  data: 0.4041  max mem: 2825
Test:  [100/441]  eta: 0:00:28  model_time: 0.0445 (0.0452)  evaluator_time: 0.0117 (0.0237)  time: 0.0751  data: 0.0072  max mem: 2825
Test:  [200/441]  eta: 0:00:19  model_time: 0.0445 (0.0449)  evaluator_time: 0.0102 (0.0231)  time: 0.0743  data: 0.0073  max mem: 2825
Test:  [300/441]  eta: 0:00:11  model_time: 0.0444 (0.0448)  evaluator_time: 0.0094 (0.0230)  time: 0.0736  data: 0.0070  max mem: 2825
Test:  [400/441]  eta: 0:00:03  model_time: 0.0440 (0.0447)  evaluator_time: 0.0134 (0.0236)  time: 0.0876  data: 0.0067  max mem: 2825
Test:  [440/441]  eta: 0:00:00  model_time: 0.0439 (0.0446)  evaluator_time: 0.0091 (0.0233)  time: 0.0710  data: 0.0066  max mem: 2825
Test: Total time: 0:00:35 (0.0809 s / it)
Averaged stats: model_time: 0.0439 (0.0446)  evaluator_time: 0.0091 (0.0233)
Accumulating evaluation results...
DONE (t=0.27s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.382
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.647
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.391
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.093
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.388
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.626
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.076
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.336
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.470
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.242
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.488
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.669
Epoch: [7]  [   0/1250]  eta: 0:20:17  lr: 0.000050  loss: 0.9463 (0.9463)  loss_classifier: 0.2297 (0.2297)  loss_box_reg: 0.4457 (0.4457)  loss_objectne
ss: 0.0312 (0.0312)  loss_rpn_box_reg: 0.2396 (0.2396)  time: 0.9738  data: 0.8327  max mem: 2825
Epoch: [7]  [ 100/1250]  eta: 0:03:00  lr: 0.000050  loss: 0.3687 (0.4310)  loss_classifier: 0.0971 (0.1069)  loss_box_reg: 0.1696 (0.1788)  loss_objectne
ss: 0.0389 (0.0492)  loss_rpn_box_reg: 0.0265 (0.0960)  time: 0.1614  data: 0.0479  max mem: 2825
Epoch: [7]  [ 200/1250]  eta: 0:02:41  lr: 0.000050  loss: 0.3761 (0.4383)  loss_classifier: 0.1065 (0.1080)  loss_box_reg: 0.1612 (0.1752)  loss_objectne
ss: 0.0370 (0.0510)  loss_rpn_box_reg: 0.0274 (0.1041)  time: 0.1521  data: 0.0362  max mem: 2825
Epoch: [7]  [ 300/1250]  eta: 0:02:23  lr: 0.000050  loss: 0.2920 (0.4275)  loss_classifier: 0.0529 (0.1063)  loss_box_reg: 0.0960 (0.1732)  loss_objectne
ss: 0.0261 (0.0520)  loss_rpn_box_reg: 0.0329 (0.0960)  time: 0.1384  data: 0.0235  max mem: 2825
Epoch: [7]  [ 400/1250]  eta: 0:02:08  lr: 0.000050  loss: 0.5102 (0.4361)  loss_classifier: 0.0960 (0.1091)  loss_box_reg: 0.1716 (0.1778)  loss_objectne
ss: 0.0361 (0.0509)  loss_rpn_box_reg: 0.0890 (0.0982)  time: 0.1545  data: 0.0368  max mem: 2825
Epoch: [7]  [ 500/1250]  eta: 0:01:54  lr: 0.000050  loss: 0.3580 (0.4351)  loss_classifier: 0.0942 (0.1093)  loss_box_reg: 0.1756 (0.1783)  loss_objectne
ss: 0.0484 (0.0510)  loss_rpn_box_reg: 0.0508 (0.0965)  time: 0.1530  data: 0.0347  max mem: 2825
Epoch: [7]  [ 600/1250]  eta: 0:01:39  lr: 0.000050  loss: 0.4923 (0.4424)  loss_classifier: 0.1217 (0.1098)  loss_box_reg: 0.1909 (0.1796)  loss_objectne
ss: 0.0486 (0.0518)  loss_rpn_box_reg: 0.0932 (0.1011)  time: 0.1415  data: 0.0242  max mem: 2825
Epoch: [7]  [ 700/1250]  eta: 0:01:25  lr: 0.000050  loss: 0.3001 (0.4397)  loss_classifier: 0.0709 (0.1091)  loss_box_reg: 0.1191 (0.1780)  loss_objectne
ss: 0.0405 (0.0518)  loss_rpn_box_reg: 0.0256 (0.1007)  time: 0.1378  data: 0.0206  max mem: 2825
Epoch: [7]  [ 800/1250]  eta: 0:01:09  lr: 0.000050  loss: 0.3319 (0.4388)  loss_classifier: 0.0957 (0.1096)  loss_box_reg: 0.1555 (0.1792)  loss_objectne
ss: 0.0326 (0.0511)  loss_rpn_box_reg: 0.0314 (0.0988)  time: 0.1475  data: 0.0319  max mem: 2825
Epoch: [7]  [ 900/1250]  eta: 0:00:53  lr: 0.000050  loss: 0.3771 (0.4396)  loss_classifier: 0.0899 (0.1096)  loss_box_reg: 0.1682 (0.1792)  loss_objectne
ss: 0.0378 (0.0517)  loss_rpn_box_reg: 0.0955 (0.0992)  time: 0.1568  data: 0.0400  max mem: 2825
Epoch: [7]  [1000/1250]  eta: 0:00:38  lr: 0.000050  loss: 0.2449 (0.4367)  loss_classifier: 0.0687 (0.1090)  loss_box_reg: 0.0985 (0.1784)  loss_objectne
ss: 0.0408 (0.0509)  loss_rpn_box_reg: 0.0272 (0.0984)  time: 0.1539  data: 0.0347  max mem: 2825
Epoch: [7]  [1100/1250]  eta: 0:00:23  lr: 0.000050  loss: 0.3429 (0.4342)  loss_classifier: 0.0808 (0.1086)  loss_box_reg: 0.1508 (0.1779)  loss_objectne
ss: 0.0309 (0.0506)  loss_rpn_box_reg: 0.0351 (0.0972)  time: 0.1393  data: 0.0188  max mem: 2825
Epoch: [7]  [1200/1250]  eta: 0:00:07  lr: 0.000050  loss: 0.4769 (0.4376)  loss_classifier: 0.1033 (0.1090)  loss_box_reg: 0.1704 (0.1786)  loss_objectne
ss: 0.0656 (0.0511)  loss_rpn_box_reg: 0.1419 (0.0989)  time: 0.1329  data: 0.0185  max mem: 2825
Epoch: [7]  [1249/1250]  eta: 0:00:00  lr: 0.000050  loss: 0.4342 (0.4377)  loss_classifier: 0.1178 (0.1093)  loss_box_reg: 0.2182 (0.1791)  loss_objectne
ss: 0.0413 (0.0508)  loss_rpn_box_reg: 0.0361 (0.0984)  time: 0.1429  data: 0.0255  max mem: 2825
Epoch: [7] Total time: 0:03:12 (0.1536 s / it)
creating index...
index created!
Test:  [  0/441]  eta: 0:03:35  model_time: 0.0587 (0.0587)  evaluator_time: 0.0153 (0.0153)  time: 0.4893  data: 0.4095  max mem: 2825
Test:  [100/441]  eta: 0:00:28  model_time: 0.0443 (0.0449)  evaluator_time: 0.0075 (0.0237)  time: 0.0750  data: 0.0069  max mem: 2825
Test:  [200/441]  eta: 0:00:19  model_time: 0.0441 (0.0446)  evaluator_time: 0.0108 (0.0235)  time: 0.0732  data: 0.0067  max mem: 2825
```

```
Test:  [300/441]  eta: 0:00:11  model_time: 0.0440 (0.0445)  evaluator_time: 0.0078 (0.0233)  time: 0.0743  data: 0.0070  max mem: 2825
Test:  [400/441]  eta: 0:00:03  model_time: 0.0441 (0.0446)  evaluator_time: 0.0142 (0.0240)  time: 0.0873  data: 0.0070  max mem: 2825
Test:  [440/441]  eta: 0:00:00  model_time: 0.0439 (0.0445)  evaluator_time: 0.0096 (0.0237)  time: 0.0713  data: 0.0066  max mem: 2825
Test: Total time: 0:00:35 (0.0812 s / it)
Averaged stats: model_time: 0.0439 (0.0445)  evaluator_time: 0.0096 (0.0237)
Accumulating evaluation results...
DONE (t=0.27s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.383
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.648
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.393
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.091
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.391
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.625
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.076
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.337
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.471
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.240
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.491
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.669
Epoch: [8]  [   0/1250]  eta: 0:14:51  lr: 0.000050  loss: 0.1850 (0.1850)  loss_classifier: 0.0504 (0.0504)  loss_box_reg: 0.1089 (0.1089)  loss_objectne
ss: 0.0235 (0.0235)  loss_rpn_box_reg: 0.0021 (0.0021)  time: 0.7136  data: 0.5765  max mem: 2825
Epoch: [8]  [ 100/1250]  eta: 0:03:12  lr: 0.000050  loss: 0.3719 (0.4669)  loss_classifier: 0.1050 (0.1129)  loss_box_reg: 0.1898 (0.1865)  loss_objectne
ss: 0.0434 (0.0568)  loss_rpn_box_reg: 0.0335 (0.1108)  time: 0.1668  data: 0.0521  max mem: 2825
Epoch: [8]  [ 200/1250]  eta: 0:02:50  lr: 0.000050  loss: 0.4205 (0.4611)  loss_classifier: 0.1148 (0.1141)  loss_box_reg: 0.1961 (0.1866)  loss_objectne
ss: 0.0319 (0.0536)  loss_rpn_box_reg: 0.0734 (0.1068)  time: 0.1549  data: 0.0372  max mem: 2825
Epoch: [8]  [ 300/1250]  eta: 0:02:32  lr: 0.000050  loss: 0.4487 (0.4574)  loss_classifier: 0.1255 (0.1133)  loss_box_reg: 0.1789 (0.1861)  loss_objectne
ss: 0.0411 (0.0512)  loss_rpn_box_reg: 0.0812 (0.1068)  time: 0.1642  data: 0.0483  max mem: 2825
Epoch: [8]  [ 400/1250]  eta: 0:02:14  lr: 0.000050  loss: 0.3973 (0.4556)  loss_classifier: 0.1000 (0.1127)  loss_box_reg: 0.1496 (0.1858)  loss_objectne
ss: 0.0428 (0.0508)  loss_rpn_box_reg: 0.0829 (0.1063)  time: 0.1717  data: 0.0511  max mem: 2825
Epoch: [8]  [ 500/1250]  eta: 0:01:57  lr: 0.000050  loss: 0.4318 (0.4537)  loss_classifier: 0.1115 (0.1115)  loss_box_reg: 0.1663 (0.1831)  loss_objectne
ss: 0.0461 (0.0526)  loss_rpn_box_reg: 0.0784 (0.1065)  time: 0.1430  data: 0.0291  max mem: 2825
Epoch: [8]  [ 600/1250]  eta: 0:01:42  lr: 0.000050  loss: 0.3523 (0.4496)  loss_classifier: 0.1076 (0.1119)  loss_box_reg: 0.1697 (0.1825)  loss_objectne
ss: 0.0367 (0.0519)  loss_rpn_box_reg: 0.0388 (0.1034)  time: 0.1619  data: 0.0443  max mem: 2825
Epoch: [8]  [ 700/1250]  eta: 0:01:27  lr: 0.000050  loss: 0.3714 (0.4417)  loss_classifier: 0.1199 (0.1104)  loss_box_reg: 0.1789 (0.1805)  loss_objectne
ss: 0.0449 (0.0505)  loss_rpn_box_reg: 0.0474 (0.1002)  time: 0.1547  data: 0.0396  max mem: 2825
Epoch: [8]  [ 800/1250]  eta: 0:01:11  lr: 0.000050  loss: 0.2949 (0.4428)  loss_classifier: 0.0879 (0.1106)  loss_box_reg: 0.1610 (0.1804)  loss_objectne
ss: 0.0354 (0.0504)  loss_rpn_box_reg: 0.0453 (0.1013)  time: 0.1512  data: 0.0320  max mem: 2825
Epoch: [8]  [ 900/1250]  eta: 0:00:55  lr: 0.000050  loss: 0.2875 (0.4386)  loss_classifier: 0.0907 (0.1100)  loss_box_reg: 0.1237 (0.1796)  loss_objectne
ss: 0.0366 (0.0500)  loss_rpn_box_reg: 0.0193 (0.0990)  time: 0.1431  data: 0.0264  max mem: 2825
Epoch: [8]  [1000/1250]  eta: 0:00:39  lr: 0.000050  loss: 0.3172 (0.4405)  loss_classifier: 0.0839 (0.1102)  loss_box_reg: 0.1385 (0.1800)  loss_objectne
ss: 0.0404 (0.0502)  loss_rpn_box_reg: 0.0318 (0.1001)  time: 0.1694  data: 0.0545  max mem: 2825
Epoch: [8]  [1100/1250]  eta: 0:00:23  lr: 0.000050  loss: 0.3514 (0.4394)  loss_classifier: 0.0905 (0.1097)  loss_box_reg: 0.1347 (0.1791)  loss_objectne
ss: 0.0328 (0.0502)  loss_rpn_box_reg: 0.0749 (0.1004)  time: 0.1536  data: 0.0358  max mem: 2825
Epoch: [8]  [1200/1250]  eta: 0:00:07  lr: 0.000050  loss: 0.4038 (0.4383)  loss_classifier: 0.0917 (0.1094)  loss_box_reg: 0.1649 (0.1791)  loss_objectne
ss: 0.0317 (0.0503)  loss_rpn_box_reg: 0.0929 (0.0995)  time: 0.1596  data: 0.0454  max mem: 2825
Epoch: [8]  [1249/1250]  eta: 0:00:00  lr: 0.000050  loss: 0.2937 (0.4364)  loss_classifier: 0.0836 (0.1092)  loss_box_reg: 0.1242 (0.1786)  loss_objectne
ss: 0.0329 (0.0499)  loss_rpn_box_reg: 0.0424 (0.0987)  time: 0.1357  data: 0.0191  max mem: 2825
Epoch: [8] Total time: 0:03:15 (0.1565 s / it)
creating index...
index created!
Test:  [  0/441]  eta: 0:03:39  model_time: 0.0580 (0.0580)  evaluator_time: 0.0159 (0.0159)  time: 0.4988  data: 0.4191  max mem: 2825
Test:  [100/441]  eta: 0:00:28  model_time: 0.0444 (0.0449)  evaluator_time: 0.0082 (0.0231)  time: 0.0745  data: 0.0070  max mem: 2825
Test:  [200/441]  eta: 0:00:19  model_time: 0.0444 (0.0447)  evaluator_time: 0.0104 (0.0229)  time: 0.0754  data: 0.0075  max mem: 2825
Test:  [300/441]  eta: 0:00:11  model_time: 0.0441 (0.0446)  evaluator_time: 0.0076 (0.0228)  time: 0.0734  data: 0.0071  max mem: 2825
Test:  [400/441]  eta: 0:00:03  model_time: 0.0441 (0.0445)  evaluator_time: 0.0139 (0.0237)  time: 0.0857  data: 0.0068  max mem: 2825
Test:  [440/441]  eta: 0:00:00  model_time: 0.0441 (0.0445)  evaluator_time: 0.0086 (0.0234)  time: 0.0718  data: 0.0068  max mem: 2825
Test: Total time: 0:00:35 (0.0810 s / it)
Averaged stats: model_time: 0.0441 (0.0445)  evaluator_time: 0.0086 (0.0234)
Accumulating evaluation results...
DONE (t=0.27s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.382
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.648
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.393
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.091
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.390
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.624
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.076
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.337
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.470
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.238
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.490
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.667
Epoch: [9]  [   0/1250]  eta: 0:14:31  lr: 0.000005  loss: 0.3199 (0.3199)  loss_classifier: 0.0968 (0.0968)  loss_box_reg: 0.1442 (0.1442)  loss_objectne
ss: 0.0413 (0.0413)  loss_rpn_box_reg: 0.0376 (0.0376)  time: 0.6970  data: 0.5644  max mem: 2825
Epoch: [9]  [ 100/1250]  eta: 0:03:14  lr: 0.000005  loss: 0.3430 (0.4559)  loss_classifier: 0.0840 (0.1143)  loss_box_reg: 0.1188 (0.1827)  loss_objectne
ss: 0.0410 (0.0507)  loss_rpn_box_reg: 0.0580 (0.1083)  time: 0.1982  data: 0.0836  max mem: 2825
Epoch: [9]  [ 200/1250]  eta: 0:02:52  lr: 0.000005  loss: 0.4671 (0.4401)  loss_classifier: 0.1076 (0.1089)  loss_box_reg: 0.1662 (0.1811)  loss_objectne
ss: 0.0441 (0.0507)  loss_rpn_box_reg: 0.0483 (0.0994)  time: 0.1370  data: 0.0190  max mem: 2825
Epoch: [9]  [ 300/1250]  eta: 0:02:31  lr: 0.000005  loss: 0.3248 (0.4327)  loss_classifier: 0.0890 (0.1074)  loss_box_reg: 0.1304 (0.1793)  loss_objectne
ss: 0.0344 (0.0487)  loss_rpn_box_reg: 0.0176 (0.0973)  time: 0.1627  data: 0.0471  max mem: 2825
Epoch: [9]  [ 400/1250]  eta: 0:02:14  lr: 0.000005  loss: 0.3078 (0.4334)  loss_classifier: 0.0915 (0.1081)  loss_box_reg: 0.1471 (0.1797)  loss_objectne
ss: 0.0356 (0.0494)  loss_rpn_box_reg: 0.0279 (0.0962)  time: 0.1514  data: 0.0346  max mem: 2825
Epoch: [9]  [ 500/1250]  eta: 0:01:59  lr: 0.000005  loss: 0.3322 (0.4368)  loss_classifier: 0.0818 (0.1088)  loss_box_reg: 0.1353 (0.1804)  loss_objectne
ss: 0.0353 (0.0494)  loss_rpn_box_reg: 0.0415 (0.0983)  time: 0.1539  data: 0.0387  max mem: 2825
Epoch: [9]  [ 600/1250]  eta: 0:01:42  lr: 0.000005  loss: 0.2853 (0.4346)  loss_classifier: 0.0730 (0.1083)  loss_box_reg: 0.1156 (0.1788)  loss_objectne
ss: 0.0440 (0.0494)  loss_rpn_box_reg: 0.0323 (0.0982)  time: 0.1498  data: 0.0283  max mem: 2825
Epoch: [9]  [ 700/1250]  eta: 0:01:26  lr: 0.000005  loss: 0.3424 (0.4377)  loss_classifier: 0.0990 (0.1090)  loss_box_reg: 0.1800 (0.1795)  loss_objectne
ss: 0.0425 (0.0511)  loss_rpn_box_reg: 0.0309 (0.0980)  time: 0.1487  data: 0.0331  max mem: 2825
Epoch: [9]  [ 800/1250]  eta: 0:01:10  lr: 0.000005  loss: 0.3670 (0.4403)  loss_classifier: 0.0997 (0.1097)  loss_box_reg: 0.1696 (0.1799)  loss_objectne
ss: 0.0371 (0.0511)  loss_rpn_box_reg: 0.0548 (0.0996)  time: 0.1527  data: 0.0376  max mem: 2825
Epoch: [9]  [ 900/1250]  eta: 0:00:54  lr: 0.000005  loss: 0.4120 (0.4410)  loss_classifier: 0.1075 (0.1101)  loss_box_reg: 0.1631 (0.1808)  loss_objectne
ss: 0.0352 (0.0508)  loss_rpn_box_reg: 0.0707 (0.0993)  time: 0.1729  data: 0.0557  max mem: 2825
Epoch: [9]  [1000/1250]  eta: 0:00:39  lr: 0.000005  loss: 0.3912 (0.4407)  loss_classifier: 0.1088 (0.1098)  loss_box_reg: 0.1462 (0.1805)  loss_objectne
ss: 0.0399 (0.0507)  loss_rpn_box_reg: 0.0721 (0.0997)  time: 0.1522  data: 0.0329  max mem: 2825
Epoch: [9]  [1100/1250]  eta: 0:00:23  lr: 0.000005  loss: 0.3970 (0.4377)  loss_classifier: 0.0957 (0.1092)  loss_box_reg: 0.1652 (0.1790)  loss_objectne
ss: 0.0456 (0.0506)  loss_rpn_box_reg: 0.0529 (0.0989)  time: 0.1373  data: 0.0221  max mem: 2825
Epoch: [9]  [1200/1250]  eta: 0:00:07  lr: 0.000005  loss: 0.2983 (0.4355)  loss_classifier: 0.0801 (0.1090)  loss_box_reg: 0.1344 (0.1785)  loss_objectne
ss: 0.0349 (0.0501)  loss_rpn_box_reg: 0.0328 (0.0979)  time: 0.1480  data: 0.0301  max mem: 2825
Epoch: [9]  [1249/1250]  eta: 0:00:00  lr: 0.000005  loss: 0.4199 (0.4362)  loss_classifier: 0.1032 (0.1089)  loss_box_reg: 0.1682 (0.1787)  loss_objectne
ss: 0.0302 (0.0500)  loss_rpn_box_reg: 0.0684 (0.0985)  time: 0.1450  data: 0.0317  max mem: 2825
Epoch: [9] Total time: 0:03:13 (0.1550 s / it)
creating index...
index created!
Test:  [  0/441]  eta: 0:03:34  model_time: 0.0582 (0.0582)  evaluator_time: 0.0153 (0.0153)  time: 0.4875  data: 0.4081  max mem: 2825
Test:  [100/441]  eta: 0:00:28  model_time: 0.0442 (0.0446)  evaluator_time: 0.0066 (0.0228)  time: 0.0722  data: 0.0070  max mem: 2825
Test:  [200/441]  eta: 0:00:19  model_time: 0.0442 (0.0446)  evaluator_time: 0.0101 (0.0232)  time: 0.0737  data: 0.0073  max mem: 2825
Test:  [300/441]  eta: 0:00:11  model_time: 0.0443 (0.0447)  evaluator_time: 0.0077 (0.0231)  time: 0.0731  data: 0.0070  max mem: 2825
Test:  [400/441]  eta: 0:00:03  model_time: 0.0441 (0.0446)  evaluator_time: 0.0137 (0.0237)  time: 0.0882  data: 0.0069  max mem: 2825
Test:  [440/441]  eta: 0:00:00  model_time: 0.0441 (0.0446)  evaluator_time: 0.0086 (0.0235)  time: 0.0720  data: 0.0067  max mem: 2825
Test: Total time: 0:00:35 (0.0811 s / it)
Averaged stats: model_time: 0.0441 (0.0446)  evaluator_time: 0.0086 (0.0235)
Accumulating evaluation results...
DONE (t=0.27s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.382
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.649
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.393
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.091
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.390
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.625
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.076
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.337
```

```
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.471
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.239
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.491
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.668
```

Training time on Nvidia Tesla V100 with 16 GB memory

In [289…

```
print('Time elapsed = %.2f min' % ((tend - tstart)/60))
```

```
Time elapsed = 46.60 min
```

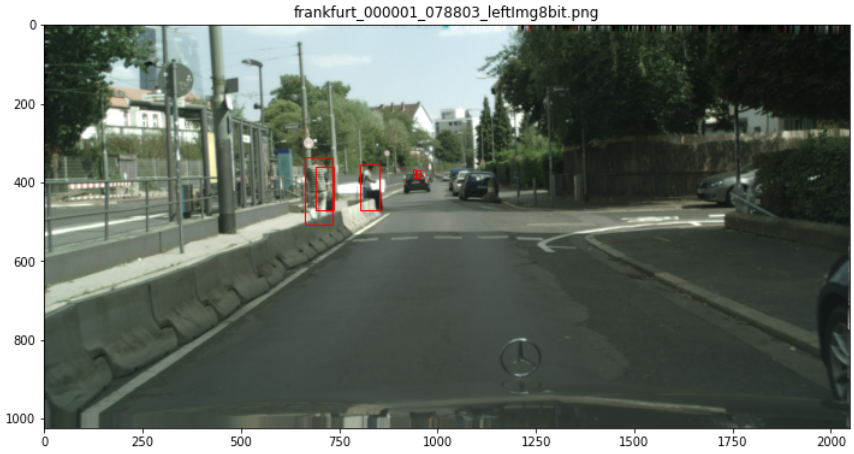
4. Error Analysis

Ground truth

Select one image from test set with occlusions:

In [272…

```
i = 0
img_name = imgs_test[i]
bboxes = anno_test[img_name]

img_path = imgs_path_test + img_name
img = Image.open(img_path)
np_image = np.array(img)

img_raw = img
plt.rcParams['figure.figsize'] = [12, 8]
fig, ax = plt.subplots()
plt.title('Pedestrians detected')
ax.imshow(img)
for bbox in bboxes:
    rect = patches.Rectangle(
        (bbox[0], bbox[1]), bbox[2], bbox[3],
        linewidth=1, edgecolor='r', facecolor='none')

    ax.add_patch(rect)

plt.title(img_name)
plt.show()
```



In [273…

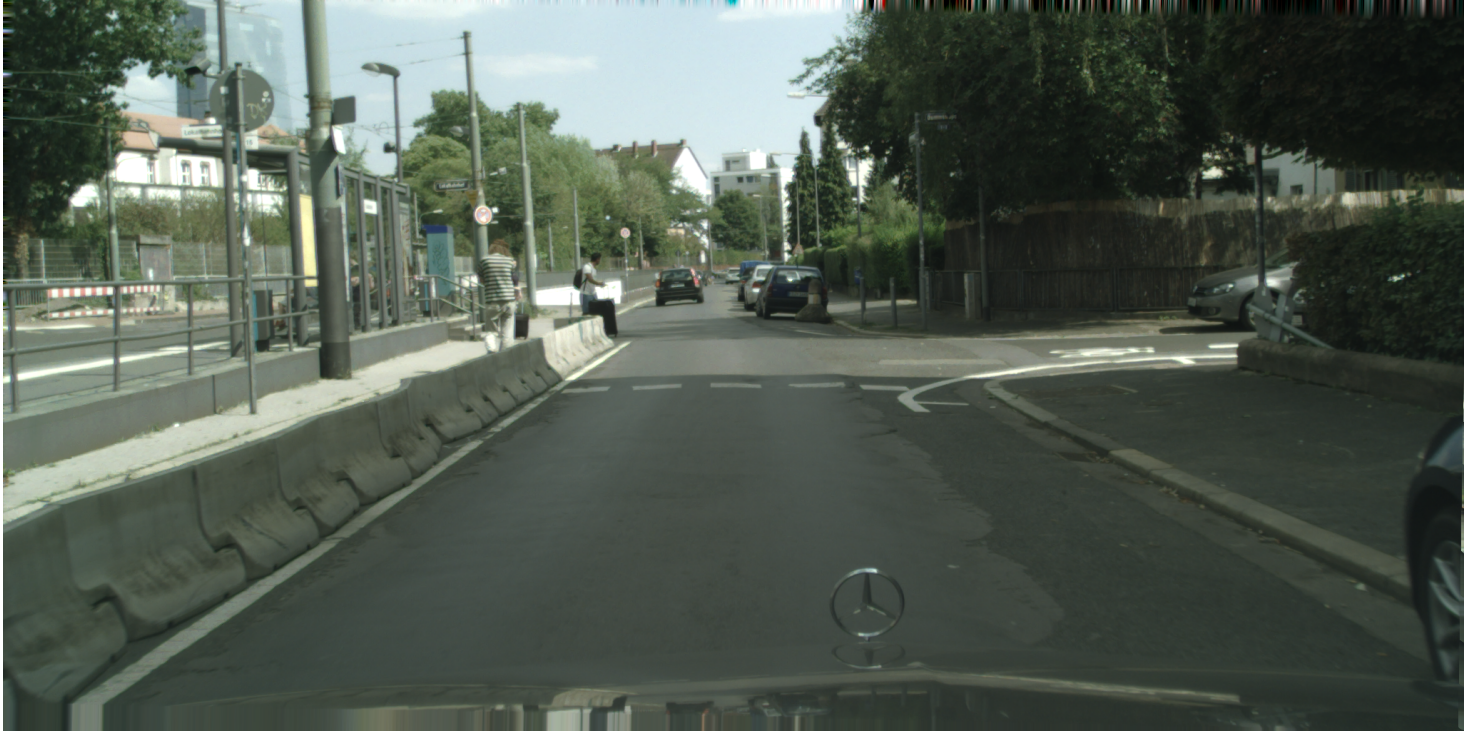
```
def convert_back(img, img_raw):
    img = Image.fromarray(img.mul(255).permute(1, 2, 0).byte().numpy())
    img = np.array(img)
    print('img shape: %d x %d x %d' % img.shape)
    img = Image.fromarray(np.uint8(img)).convert('RGB')

    img_flipped = np.array(img.transpose(Image.FLIP_LEFT_RIGHT))
    img_raw = np.array(img_raw)
    img_was_flipped = np.sum(img_flipped.flatten() == img_raw.flatten()) == img_flipped.shape[0] * img_flipped.shape[1] * img_flipped.shape[2]
    print('Image was flipped: %r' % img_was_flipped)

    return img
```

In [274…

```
## check again
i = 0
img_name = imgs_test[i]
img_path = imgs_path_test + img_name
img_raw = Image.open(img_path)
img, anno = dataset_test[i]
convert_back(img, img_raw)
```

```
img shape: 1024 x 2048 x 3
Image was flipped: False
```

Out[274…



Predicted

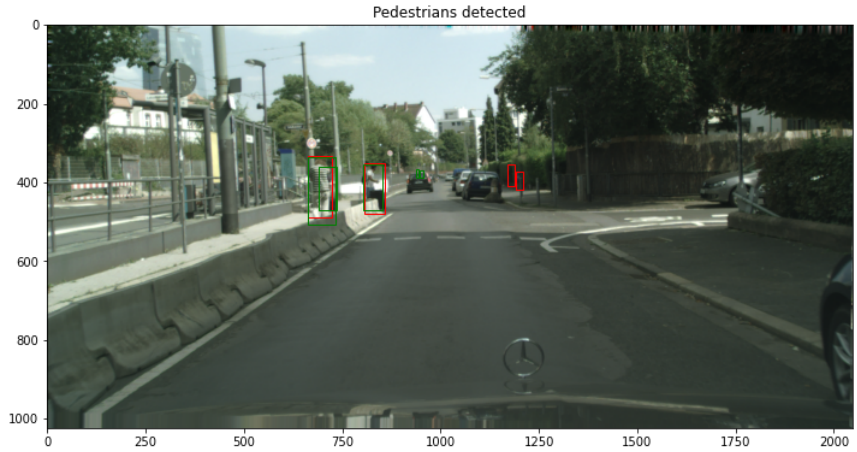
In [275…

```
def show(i, imgs_test, threshold=.3):
    img_name = imgs_test[i]
    bboxes_gt = anno_test[img_name]

    img_path = imgs_path_test + img_name
    img = Image.open(img_path)
    np_image = np.array(img)
    img_raw = img

    ## idx = indices.index(i) # if on train set to unpermute
    img, anno = dataset_test[i]

    # put the model in evaluation mode
    model.eval()
    with torch.no_grad():
        prediction = model([img.to(device)])[0]

    preds = prediction['boxes'] # predicted bboxes
    preds = preds.cpu().data.numpy() # to numpy array

    scores = prediction['scores'] # scores of predicted bboxes
    scores = scores.cpu().data.numpy()

    ## keep only bboxes where score > threshold:
    highs = list(np.where(scores > threshold)[0])

    img = convert_back(img, img_raw)

    bboxes_x0x1y0y1 = []
    for high in highs:
        bboxes_x0x1y0y1.append(list(preds[high]))

    ## transform the bboxes from tensor to list and back to [x, y, w, h] format
    ## transform back to [x, y, w, h] format
    bboxes = []
    for bbox in bboxes_x0x1y0y1:
        bbox = list(bbox)
        x0, y0 = bbox[0], bbox[1]
        x1, y1 = bbox[2], bbox[3]

        bboxes.append([x0, y0, x1 - x0, y1 - y0])

    ## draw the predicted bounding boxes
    plt.rcParams['figure.figsize'] = [12, 8]

    fig, ax = plt.subplots()

    ax.imshow(img);

    # bbox = [x, y, w, h]
    for bbox in bboxes:
        rect = patches.Rectangle(
            (bbox[0], bbox[1]), bbox[2], bbox[3],
            linewidth=1, edgecolor='r', facecolor='none')

        ax.add_patch(rect)

    plt.title('Pedestrians detected')

    # bbox = [x, y, w, h]
    for bbox in bboxes_gt:
        rect = patches.Rectangle(
            (bbox[0], bbox[1]), bbox[2], bbox[3],
            linewidth=1, edgecolor='g', facecolor='none')

        ax.add_patch(rect)

    plt.show()
```

In [276…

```
show(i, imgs_test, threshold=.3)
```

```
img shape: 1024 x 2048 x 3
Image was flipped: False
```



Results:

- Fit1:

```
Averaged stats: model_time: 0.0440 (0.0444)  evaluator_time: 0.0100 (0.0249)
Accumulating evaluation results...
DONE (t=0.30s).
IoU metric: bbox
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.367
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.647
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.378
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.097
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.379
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.586
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.072
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.330
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.464
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.236
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.490
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.644
```

- Fit2:

```
Averaged stats: model_time: 0.0443 (0.0445)  evaluator_time: 0.0128 (0.0276)
Accumulating evaluation results...
DONE (t=0.34s).
IoU metric: bbox
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.374
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.646
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.384
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.097
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.384
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.607
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.074
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.335
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.469
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.242
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.492
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.656
```

- Final fit (5 epochs):

```
Averaged stats: model_time: 0.0439 (0.0445)  evaluator_time: 0.0069 (0.0218)
Accumulating evaluation results...
DONE (t=0.25s).
IoU metric: bbox
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.373
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.639
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.383
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.092
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.379
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.608
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.075
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.334
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.457
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.221
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.481
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.651
```

- Final fit (10 epochs):

```
Averaged stats: model_time: 0.0441 (0.0446)  evaluator_time: 0.0086 (0.0235)
Accumulating evaluation results...
DONE (t=0.27s).
IoU metric: bbox
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.382
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.649
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.393
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.091
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.390
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.625
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.076
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.337
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.471
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.239
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.491
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.668
```

5. Export

In [277…

```python
def save_model(model, path="./models/entire_model.pt"):
    torch.save(model, path)
    print('Model saved to ' + path)
```

In [278…
```
save_model(model, path="../data/models/model-fit3.pt")
```

```
Model saved to ../data/models/model-fit3.pt
```

Detections first just to test, I made using model

For first export of predictions I used the model I build on only 152 images from Aachen, which is a small subset of training set of Citypersons dataset. I had to test the training with complete test set of 500 images. I will train it on complete training set export predictions again soon. You can test your scripts on this.

For testing we are using Citypersons validation set of 500 images and corresponding annotations in particular 441 images where there are persons on the images. From cities: Munster, Frankfurt, Lindau. Let me know if you need image names

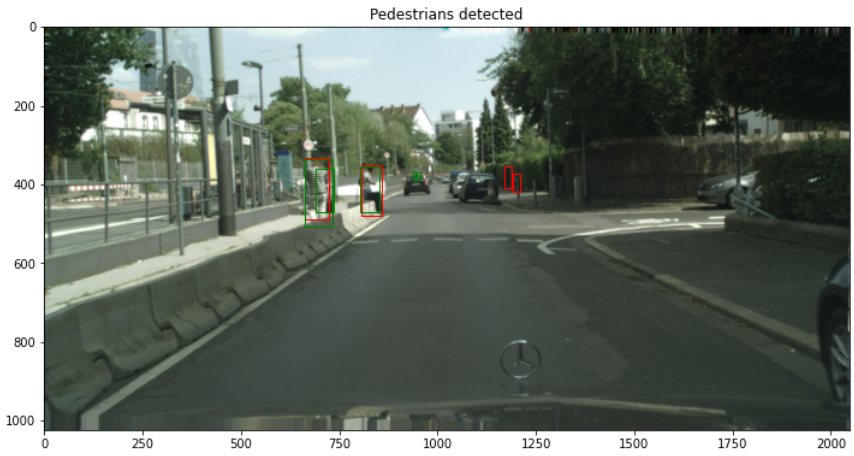
In [279…
```
len(dataset_test), len(imgs_test)
```

Out[279… (441, 441)

In [280…
```
tstart = time.time()
# put the model in evaluation mode
model.eval()

predictions = {}
for i in range(len(dataset_test)):
    img, anno = dataset_test[i]

    with torch.no_grad():
        prediction = model([img.to(device)])[0]
        predictions[imgs_test[i]] = prediction

    print('.', end = '')

tend = time.time()
print('\nExported %d predictions' % (len(predictions)))
print('\nTime elapsed = %.2f min' % ((tend - tstart)/60))
```

```
.....................................................................................................................................................
Exported 441 predictions

Time elapsed = 1.15 min
```

In [281…
```
## final check
threshold = .3
i = 0

prediction = predictions[imgs_test[i]]
# preds
# pred['boxes'].cpu().data.numpy()

img_name = imgs_test[i]
bboxes_gt = anno_test[img_name]

img_path = imgs_path_test + img_name
img = Image.open(img_path)
np_image = np.array(img)
img_raw = img

preds = prediction['boxes'] # predicted bboxes
preds = preds.cpu().data.numpy() # to numpy array

scores = prediction['scores'] # scores of predicted bboxes
scores = scores.cpu().data.numpy()

## keep only bboxes where score > threshold:
highs = list(np.where(scores > threshold)[0])

bboxes_x0x1y0y1 = []
for high in highs:
    bboxes_x0x1y0y1.append(list(preds[high]))

## transform the bboxes from tensor to list and back to [x, y, w, h] format
## transform back to [x, y, w, h] format
bboxes = []
for bbox in bboxes_x0x1y0y1:
    bbox = list(bbox)
    x0, y0 = bbox[0], bbox[1]
    x1, y1 = bbox[2], bbox[3]

    bboxes.append([x0, y0, x1 - x0, y1 - y0])

## draw the predicted bounding boxes
plt.rcParams['figure.figsize'] = [12, 8]

fig, ax = plt.subplots()

ax.imshow(img);

## detections in red
for bbox in bboxes:
    rect = patches.Rectangle(
        (bbox[0], bbox[1]), bbox[2], bbox[3],
        linewidth=1, edgecolor='r', facecolor='none')

    ax.add_patch(rect)

plt.title('Pedestrians detected')

## ground truths in green
for bbox in bboxes_gt:
    rect = patches.Rectangle(
        (bbox[0], bbox[1]), bbox[2], bbox[3],
        linewidth=1, edgecolor='g', facecolor='none')

    ax.add_patch(rect)

plt.show()
```



In [282…
```
## upack transform
predictions2 = {}
for img in imgs_test:
    pred = predictions[img]
    boxes, labels, scores = pred['boxes'], pred['labels'], pred['scores']
    predictions2[img] = {
        'boxes': boxes.cpu().detach().numpy(),
        'labels': labels.cpu().detach().numpy(),
        'scores': scores.cpu().detach().numpy(),
    }
```

In [283…
```
import pickle
with open(r"../data/predictions-fit3.pickle", "wb") as output_file:
    pickle.dump(predictions2, output_file)
```

In [284…
```
predictions2[imgs_test[0]]
```

Out[284…
```
{'boxes': array([[ 662.8422 ,  333.23648,  724.67004,  489.5734 ],
       [ 804.9902 ,  351.8607 ,  857.18896,  478.32767],
       [1190.9387 ,  372.5352 , 1208.9575 ,  417.42972],
       [1168.5942 ,  354.90222, 1189.2792 ,  410.26035],
       [ 506.14056,  381.96063,  529.40564,  441.35327],
       [ 482.36337,  385.08633,  507.2187 ,  452.23398],
       [ 974.3028 ,  372.84573,  986.3778 ,  403.5369 ],
       [1666.2249 ,  244.87538, 1683.0969 ,  285.50778],
       [ 980.4713 ,  380.0943 ,  991.3689 ,  401.73987],
       [ 797.926  ,  359.9849 ,  840.6897 ,  459.0884 ],
       [1093.2878 ,  349.7019 , 1113.2712 ,  400.63788],
       [1106.1025 ,  339.00433, 1123.4646 ,  380.41708],
       [1100.2612 ,  338.5715 , 1117.2216 ,  374.12656],
       [ 961.02277,  372.21182,  975.7608 ,  409.11932],
       [1100.98   ,  349.47116, 1121.0948 ,  401.87943],
       [ 969.98114,  373.68027,  982.4322 ,  404.15237],
       [1093.0094 ,  340.18192, 1110.3083 ,  370.44778],
       [ 203.38579,  411.0308 ,  218.29079,  439.7943 ],
       [ 985.6838 ,  373.08505,  996.0272 ,  398.62445],
       [ 480.6779 ,  365.9381 ,  520.2521 ,  464.04248],
       [1638.8489 ,  244.40083, 1660.9598 ,  297.7853 ],
       [ 531.8914 ,  390.41302,  551.45416,  447.91772],
       [ 487.67883,  382.57794,  543.7731 ,  459.7677 ],
       [1030.9539 ,  362.50424, 1050.144  ,  414.33475],
       [ 565.5408 ,  336.61697,  611.68286,  444.60886],
       [ 894.5299 ,  356.28265,  908.48584,  378.33298],
       [ 487.79684,  378.1975 ,  509.60413,  431.21844]], dtype=float32),
 'labels': array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1]),
 'scores': array([0.9954417 , 0.98729044, 0.615358  , 0.49267325, 0.26960692,
       0.23225957, 0.16275023, 0.15749817, 0.13838229, 0.13712686,
       0.12830791, 0.12077933, 0.10867473, 0.10105093, 0.10023607,
       0.09701079, 0.08561058, 0.08243335, 0.08148868, 0.07317794,
       0.0699328 , 0.06809862, 0.06504314, 0.06342299, 0.05445972,
       0.05393126, 0.05173681], dtype=float32)}
```

In [285…
```
predictions2[imgs_test[1]]
```

Out[285…
```
{'boxes': array([[ 807.6329 ,  343.27094,  890.54095,  541.85565],
       [ 890.9647 ,  356.02332,  969.51465,  556.47705],
       [ 474.9681 ,  386.52133,  507.82343,  469.51187],
       [ 827.70087,  346.0721 ,  930.8689 ,  561.2693 ],
       [ 485.3781 ,  386.30475,  511.11487,  445.50436],
       [ 484.03607,  340.01706,  505.17908,  391.65332],
       [ 424.89105,  379.37683,  444.8973 ,  429.67422],
       [ 464.28568,  389.54422,  518.2076 ,  505.72626],
       [1025.3784 ,  389.18253, 1042.268  ,  429.142  ],
       [ 741.1415 ,  326.30078,  795.6432 ,  454.52313],
       [ 892.6991 ,  381.67633,  940.414  ,  509.3583 ],
       [1015.54474,  388.00772, 1036.6201 ,  438.29373],
       [1011.90265,  386.93658, 1030.1705 ,  427.48602],
       [ 936.6248 ,  397.043  ,  973.863  ,  492.2272 ],
       [1032.6469 ,  386.47656, 1053.1163 ,  437.26135],
       [1633.1798 ,  376.05768, 1682.7677 ,  507.09323],
       [ 406.62244,  418.8006 ,  430.04385,  476.8751 ],
       [ 805.1208 ,  351.11926,  865.1101 ,  495.97406]], dtype=float32),
 'labels': array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]),
 'scores': array([0.99913067, 0.998912  , 0.7067206 , 0.27925304, 0.21685177,
       0.17854594, 0.17204796, 0.1240773 , 0.11756765, 0.10845991,
       0.08898115, 0.07367137, 0.07041547, 0.06784511, 0.06294197,
       0.0562113 , 0.05353821, 0.05334361], dtype=float32)}
```

In [ ]:
```
```