Final Model

In [1]:
```
%load_ext autoreload
%autoreload 2
```

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

Setup

Dependencies

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

In [2]:
```
import torch
print(torch.__version__)
```

```
1.8.0
```

```
$ pip install --upgrade pip
$ pip install torchvision
$ pip install pycocotools
```

Helper functions

We modifed some helper functions to simplify training and evaluating detection model. We are using `/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 [3]:
```
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 [4]:
```
import scipy.io
import torch
import torch.utils.data
import torchvision
```

In [5]:
```
# ../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
```

In [6]:
```
module_path = os.path.abspath(os.path.join('../src/main-model/'))
if module_path not in sys.path:
    sys.path.append(module_path)

import fasterutils
import fasterrcnn
```

Data

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 [7]:
```
imgs_path_train = '../datasets/cityscapes/train/'
imgs_path_test = '../datasets/cityscapes/test/'
```

In [8]:
```
anno_dict_train = 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_train, **anno_dict_val}
```

In [9]:
```
## fix the dictionary
cutoff = 100
anno_dict2 = {}
for img in list(anno_dict.keys()):
    bboxes = anno_dict[img]

    bboxes2 = []
    for bbox in bboxes:
        if np.prod(bbox[2:]) > cutoff:
            bboxes2.append(bbox)
    if len(bboxes2) > 0:
        anno_dict2[img] = bboxes2
assert len(anno_dict) == len(anno_dict2), 'images removed'
```

In [10]:
```
anno_dict = anno_dict2
```

In [11]:
```
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 [12]:
```
imgs1 = os.listdir(imgs_path_test)
imgs2 = list(anno_dict.keys())
imgs3 = list(set(imgs1) & set(imgs2))

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

In [13]:
```
print('In train: %d, in test: %d.' % (len(imgs_train), len(imgs_test)))
```

```
In train: 2315, in test: 418.
```

In [14]:
```
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 [15]:
```
plt.rcParams['figure.figsize'] = [12, 8]
```

In [16]:
```
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 [17]:
```
img_np = np.array(img)
H, W = img_np.shape[:2]
H, W
```

Out[17]: (1024, 2048)

Prepare Torch.utils.data.Dataset class for this dataset

In [18]:
```
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 [20]:
```
dataset = fasterrcnn.Dataset(imgs_paths_train, imgs_train, anno_train)
dataset_test = fasterrcnn.Dataset(imgs_paths_test, imgs_test, anno_test)
```

In [21]:
```
## test it
data = []
for i in range(10):
    data.append(dataset[i])
```

In [22]:
```
data[0][0]
```

Out[22]:

Train the model

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 [23]:
```
# 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 [24]:

```
## check it
print(len(indices), len(dataset_test))
# print(dataset[0],  dataset_test[0])
```

```
2315 418
```

In [25]:

```
## 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=0, # changed num_workers to 0
    collate_fn=utils.collate_fn)

data_loader_test = torch.utils.data.DataLoader(
    dataset_test, batch_size=1, shuffle=False, num_workers=0,  # changed num_workers to 0
    collate_fn=utils.collate_fn)
```

In [26]:

```
# check the results
# print('Batch size = %d' % (data_loader.batch_size))
# data_loader.dataset[0], data_loader_test.dataset[0]
```

In [27]:

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

Out[27]:

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

In [28]:

```
# model = fasterutils.get_model()
model = fasterutils.get_model_anchors()
model.to(device);
```

In [29]:

```
# construct an optimizer
params = [p for p in model.parameters() if p.requires_grad]
## lr, momentum, weight_decay taken from Pedestron:
# https://github.com/hasanirtiza/Pedestron/blob/master/configs/faster_rcnn_r50_fpn_1x.py
optimizer = torch.optim.SGD(params, lr=0.02, momentum=0.9, weight_decay=0.0001)
```

In [30]:

```
# 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 [31]:

```
num_epochs = 10 # total_epochs in Pedestron = 12

tstart = time.time()
all_mean_losses = []
for epoch in range(num_epochs):
    ## train for one epoch, printing every |train|/batch_size iterations
    mean_losses = train_one_epoch(
        model, optimizer, data_loader, device, epoch,
        print_freq=int(len(dataset)/2))

    all_mean_losses.append(mean_losses)

    ## update the learning rate
    lr_scheduler.step()

    ## evaluate on the test dataset
    evaluator = evaluate(model, data_loader_test, device=device)

tend = time.time()
```

```
Epoch: [0]  [   0/1158]  eta: 0:09:17  lr: 0.000040  loss: 0.8489 (0.8489)  loss_classifier: 0.5236 (0.5236)  loss_box_re
g: 0.2997 (0.2997)  loss_objectness: 0.0173 (0.0173)  loss_rpn_box_reg: 0.0084 (0.0084)  time: 0.4816  data: 0.3643  max m
em: 994
Epoch: [0]  [1157/1158]  eta: 0:00:00  lr: 0.020000  loss: 0.3510 (0.4115)  loss_classifier: 0.1107 (0.1262)  loss_box_re
g: 0.2107 (0.2288)  loss_objectness: 0.0159 (0.0346)  loss_rpn_box_reg: 0.0116 (0.0218)  time: 0.3551  data: 0.2488  max m
em: 1405
Epoch: [0] Total time: 0:07:17 (0.3775 s / it)
creating index...
index created!
Test:  [  0/418]  eta: 0:01:15  model_time: 0.0443 (0.0443)  evaluator_time: 0.0031 (0.0031)  time: 0.1805  data: 0.1294
max mem: 1405
Test:  [100/418]  eta: 0:00:56  model_time: 0.0442 (0.0443)  evaluator_time: 0.0084 (0.0175)  time: 0.1779  data: 0.1119
max mem: 1405
Test:  [200/418]  eta: 0:00:38  model_time: 0.0441 (0.0442)  evaluator_time: 0.0119 (0.0176)  time: 0.1781  data: 0.1104
```

```
max mem: 1405
Test:  [300/418]  eta: 0:00:20  model_time: 0.0438 (0.0441)  evaluator_time: 0.0044 (0.0168)  time: 0.1662  data: 0.1104
max mem: 1405
Test:  [400/418]  eta: 0:00:03  model_time: 0.0444 (0.0441)  evaluator_time: 0.0056 (0.0171)  time: 0.1712  data: 0.1132
max mem: 1405
Test:  [417/418]  eta: 0:00:00  model_time: 0.0441 (0.0441)  evaluator_time: 0.0051 (0.0174)  time: 0.1811  data: 0.1114
max mem: 1405
Test: Total time: 0:01:13 (0.1757 s / it)
Averaged stats: model_time: 0.0441 (0.0441)  evaluator_time: 0.0051 (0.0174)
Accumulating evaluation results...
DONE (t=0.23s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.417
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.737
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.431
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.066
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.350
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.569
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.089
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.382
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.512
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.210
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.470
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.623
Epoch: [1]  [   0/1158]  eta: 0:07:33  lr: 0.020000  loss: 0.6416 (0.6416)  loss_classifier: 0.1681 (0.1681)  loss_box_re
g: 0.3878 (0.3878)  loss_objectness: 0.0607 (0.0607)  loss_rpn_box_reg: 0.0251 (0.0251)  time: 0.3919  data: 0.2823  max m
em: 1405
Epoch: [1]  [1157/1158]  eta: 0:00:00  lr: 0.020000  loss: 0.3310 (0.3646)  loss_classifier: 0.1088 (0.1155)  loss_box_re
g: 0.1844 (0.1998)  loss_objectness: 0.0243 (0.0279)  loss_rpn_box_reg: 0.0145 (0.0213)  time: 0.3776  data: 0.2715  max m
em: 1412
Epoch: [1] Total time: 0:07:13 (0.3741 s / it)
creating index...
index created!
Test:  [  0/418]  eta: 0:01:06  model_time: 0.0440 (0.0440)  evaluator_time: 0.0024 (0.0024)  time: 0.1590  data: 0.1095
max mem: 1412
Test:  [100/418]  eta: 0:00:54  model_time: 0.0438 (0.0439)  evaluator_time: 0.0082 (0.0156)  time: 0.1727  data: 0.1106
max mem: 1412
Test:  [200/418]  eta: 0:00:37  model_time: 0.0439 (0.0439)  evaluator_time: 0.0095 (0.0160)  time: 0.1760  data: 0.1099
max mem: 1412
Test:  [300/418]  eta: 0:00:20  model_time: 0.0442 (0.0439)  evaluator_time: 0.0036 (0.0154)  time: 0.1680  data: 0.1121
max mem: 1412
Test:  [400/418]  eta: 0:00:03  model_time: 0.0439 (0.0439)  evaluator_time: 0.0050 (0.0157)  time: 0.1667  data: 0.1103
max mem: 1412
Test:  [417/418]  eta: 0:00:00  model_time: 0.0438 (0.0439)  evaluator_time: 0.0044 (0.0160)  time: 0.1790  data: 0.1103
max mem: 1412
Test: Total time: 0:01:12 (0.1729 s / it)
Averaged stats: model_time: 0.0438 (0.0439)  evaluator_time: 0.0044 (0.0160)
Accumulating evaluation results...
DONE (t=0.21s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.422
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.741
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.438
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.087
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.361
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.564
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.089
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.389
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.522
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.290
 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 ] = 0.627
Epoch: [2]  [   0/1158]  eta: 0:06:23  lr: 0.020000  loss: 0.2074 (0.2074)  loss_classifier: 0.0597 (0.0597)  loss_box_re
g: 0.1291 (0.1291)  loss_objectness: 0.0075 (0.0075)  loss_rpn_box_reg: 0.0112 (0.0112)  time: 0.3316  data: 0.2247  max m
em: 1412
Epoch: [2]  [1157/1158]  eta: 0:00:00  lr: 0.020000  loss: 0.2531 (0.3436)  loss_classifier: 0.0737 (0.1097)  loss_box_re
g: 0.1746 (0.1880)  loss_objectness: 0.0126 (0.0256)  loss_rpn_box_reg: 0.0068 (0.0203)  time: 0.3539  data: 0.2479  max m
em: 1413
Epoch: [2] Total time: 0:07:13 (0.3740 s / it)
creating index...
index created!
Test:  [  0/418]  eta: 0:01:08  model_time: 0.0449 (0.0449)  evaluator_time: 0.0022 (0.0022)  time: 0.1634  data: 0.1111
max mem: 1413
Test:  [100/418]  eta: 0:00:57  model_time: 0.0446 (0.0447)  evaluator_time: 0.0069 (0.0153)  time: 0.1762  data: 0.1116
max mem: 1413
Test:  [200/418]  eta: 0:00:38  model_time: 0.0447 (0.0447)  evaluator_time: 0.0093 (0.0158)  time: 0.1803  data: 0.1103
max mem: 1413
Test:  [300/418]  eta: 0:00:20  model_time: 0.0446 (0.0447)  evaluator_time: 0.0040 (0.0153)  time: 0.1685  data: 0.1108
max mem: 1413
Test:  [400/418]  eta: 0:00:03  model_time: 0.0446 (0.0447)  evaluator_time: 0.0045 (0.0155)  time: 0.1693  data: 0.1105
max mem: 1413
Test:  [417/418]  eta: 0:00:00  model_time: 0.0446 (0.0447)  evaluator_time: 0.0044 (0.0158)  time: 0.1815  data: 0.1109
max mem: 1413
Test: Total time: 0:01:14 (0.1771 s / it)
Averaged stats: model_time: 0.0446 (0.0447)  evaluator_time: 0.0044 (0.0158)
Accumulating evaluation results...
DONE (t=0.21s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.425
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.736
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.452
 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 ] = 0.367
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.558
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.089
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.389
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.522
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.310
 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 ] = 0.615
Epoch: [3]  [   0/1158]  eta: 0:07:19  lr: 0.002000  loss: 0.1167 (0.1167)  loss_classifier: 0.0385 (0.0385)  loss_box_re
g: 0.0667 (0.0667)  loss_objectness: 0.0110 (0.0110)  loss_rpn_box_reg: 0.0004 (0.0004)  time: 0.3798  data: 0.2659  max m
em: 1413
Epoch: [3]  [1157/1158]  eta: 0:00:00  lr: 0.002000  loss: 0.2488 (0.3078)  loss_classifier: 0.0783 (0.1011)  loss_box_re
```

```
g: 0.1325 (0.1703)  loss_objectness: 0.0109 (0.0212)  loss_rpn_box_reg: 0.0097 (0.0152)  time: 0.3624  data: 0.2558  max m
em: 1413
Epoch: [3] Total time: 0:07:13 (0.3746 s / it)
creating index...
index created!
Test:  [  0/418]  eta: 0:01:06  model_time: 0.0445 (0.0445)  evaluator_time: 0.0020 (0.0020)  time: 0.1591  data: 0.1095
max mem: 1413
Test:  [100/418]  eta: 0:00:55  model_time: 0.0439 (0.0440)  evaluator_time: 0.0080 (0.0153)  time: 0.1739  data: 0.1120
max mem: 1413
Test:  [200/418]  eta: 0:00:37  model_time: 0.0438 (0.0440)  evaluator_time: 0.0084 (0.0155)  time: 0.1754  data: 0.1096
max mem: 1413
Test:  [300/418]  eta: 0:00:20  model_time: 0.0438 (0.0439)  evaluator_time: 0.0038 (0.0149)  time: 0.1651  data: 0.1104
max mem: 1413
Test:  [400/418]  eta: 0:00:03  model_time: 0.0438 (0.0439)  evaluator_time: 0.0044 (0.0152)  time: 0.1653  data: 0.1095
max mem: 1413
Test:  [417/418]  eta: 0:00:00  model_time: 0.0438 (0.0439)  evaluator_time: 0.0040 (0.0156)  time: 0.1785  data: 0.1105
max mem: 1413
Test: Total time: 0:01:12 (0.1727 s / it)
Averaged stats: model_time: 0.0438 (0.0439)  evaluator_time: 0.0040 (0.0156)
Accumulating evaluation results...
DONE (t=0.21s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.457
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.751
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.488
 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 ] = 0.387
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.609
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.095
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.413
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.547
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.310
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.505
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.655
Epoch: [4]  [   0/1158]  eta: 0:08:32  lr: 0.002000  loss: 0.1354 (0.1354)  loss_classifier: 0.0447 (0.0447)  loss_box_re
g: 0.0782 (0.0782)  loss_objectness: 0.0106 (0.0106)  loss_rpn_box_reg: 0.0020 (0.0020)  time: 0.4426  data: 0.3362  max m
em: 1413
Epoch: [4]  [1157/1158]  eta: 0:00:00  lr: 0.002000  loss: 0.1836 (0.3020)  loss_classifier: 0.0658 (0.0992)  loss_box_re
g: 0.0927 (0.1672)  loss_objectness: 0.0160 (0.0210)  loss_rpn_box_reg: 0.0067 (0.0147)  time: 0.3460  data: 0.2401  max m
em: 1413
Epoch: [4] Total time: 0:07:09 (0.3705 s / it)
creating index...
index created!
Test:  [  0/418]  eta: 0:01:07  model_time: 0.0451 (0.0451)  evaluator_time: 0.0021 (0.0021)  time: 0.1618  data: 0.1115
max mem: 1413
Test:  [100/418]  eta: 0:00:54  model_time: 0.0438 (0.0439)  evaluator_time: 0.0071 (0.0144)  time: 0.1723  data: 0.1115
max mem: 1413
Test:  [200/418]  eta: 0:00:37  model_time: 0.0438 (0.0439)  evaluator_time: 0.0079 (0.0146)  time: 0.1745  data: 0.1105
max mem: 1413
Test:  [300/418]  eta: 0:00:20  model_time: 0.0438 (0.0439)  evaluator_time: 0.0033 (0.0139)  time: 0.1647  data: 0.1109
max mem: 1413
Test:  [400/418]  eta: 0:00:03  model_time: 0.0438 (0.0439)  evaluator_time: 0.0042 (0.0142)  time: 0.1652  data: 0.1104
max mem: 1413
Test:  [417/418]  eta: 0:00:00  model_time: 0.0438 (0.0439)  evaluator_time: 0.0039 (0.0146)  time: 0.1780  data: 0.1109
max mem: 1413
Test: Total time: 0:01:11 (0.1719 s / it)
Averaged stats: model_time: 0.0438 (0.0439)  evaluator_time: 0.0039 (0.0146)
Accumulating evaluation results...
DONE (t=0.20s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.457
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.751
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.486
 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 ] = 0.390
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.606
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.095
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.414
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.548
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.360
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.507
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.654
Epoch: [5]  [   0/1158]  eta: 0:07:45  lr: 0.002000  loss: 0.3820 (0.3820)  loss_classifier: 0.1489 (0.1489)  loss_box_re
g: 0.2040 (0.2040)  loss_objectness: 0.0175 (0.0175)  loss_rpn_box_reg: 0.0115 (0.0115)  time: 0.4017  data: 0.2949  max m
em: 1413
Epoch: [5]  [1157/1158]  eta: 0:00:00  lr: 0.002000  loss: 0.2548 (0.2974)  loss_classifier: 0.0914 (0.0979)  loss_box_re
g: 0.1368 (0.1650)  loss_objectness: 0.0123 (0.0200)  loss_rpn_box_reg: 0.0061 (0.0146)  time: 0.3538  data: 0.2481  max m
em: 1413
Epoch: [5] Total time: 0:07:10 (0.3722 s / it)
creating index...
index created!
Test:  [  0/418]  eta: 0:01:07  model_time: 0.0455 (0.0455)  evaluator_time: 0.0023 (0.0023)  time: 0.1614  data: 0.1104
max mem: 1413
Test:  [100/418]  eta: 0:00:54  model_time: 0.0437 (0.0440)  evaluator_time: 0.0075 (0.0148)  time: 0.1716  data: 0.1107
max mem: 1413
Test:  [200/418]  eta: 0:00:37  model_time: 0.0438 (0.0439)  evaluator_time: 0.0084 (0.0149)  time: 0.1738  data: 0.1095
max mem: 1413
Test:  [300/418]  eta: 0:00:20  model_time: 0.0438 (0.0439)  evaluator_time: 0.0037 (0.0143)  time: 0.1642  data: 0.1098
max mem: 1413
Test:  [400/418]  eta: 0:00:03  model_time: 0.0437 (0.0439)  evaluator_time: 0.0045 (0.0146)  time: 0.1648  data: 0.1096
max mem: 1413
Test:  [417/418]  eta: 0:00:00  model_time: 0.0437 (0.0439)  evaluator_time: 0.0042 (0.0150)  time: 0.1773  data: 0.1101
max mem: 1413
Test: Total time: 0:01:11 (0.1717 s / it)
Averaged stats: model_time: 0.0437 (0.0439)  evaluator_time: 0.0042 (0.0150)
Accumulating evaluation results...
DONE (t=0.20s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.457
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.756
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.487
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.087
 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 ] = 0.610
```

```
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.095
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.413
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.546
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.350
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.502
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.661
Epoch: [6]  [   0/1158]  eta: 0:07:03  lr: 0.000200  loss: 0.2568 (0.2568)  loss_classifier: 0.0789 (0.0789)  loss_box_re
g: 0.1652 (0.1652)  loss_objectness: 0.0056 (0.0056)  loss_rpn_box_reg: 0.0072 (0.0072)  time: 0.3661  data: 0.2595  max m
em: 1413
Epoch: [6]  [1157/1158]  eta: 0:00:00  lr: 0.000200  loss: 0.3355 (0.2932)  loss_classifier: 0.0987 (0.0969)  loss_box_re
g: 0.1549 (0.1629)  loss_objectness: 0.0205 (0.0194)  loss_rpn_box_reg: 0.0112 (0.0140)  time: 0.3674  data: 0.2624  max m
em: 1419
Epoch: [6] Total time: 0:07:15 (0.3764 s / it)
creating index...
index created!
Test:  [  0/418]  eta: 0:01:06  model_time: 0.0446 (0.0446)  evaluator_time: 0.0022 (0.0022)  time: 0.1591  data: 0.1093
max mem: 1419
Test:  [100/418]  eta: 0:00:54  model_time: 0.0438 (0.0439)  evaluator_time: 0.0083 (0.0148)  time: 0.1715  data: 0.1106
max mem: 1419
Test:  [200/418]  eta: 0:00:37  model_time: 0.0438 (0.0439)  evaluator_time: 0.0083 (0.0150)  time: 0.1744  data: 0.1096
max mem: 1419
Test:  [300/418]  eta: 0:00:20  model_time: 0.0438 (0.0439)  evaluator_time: 0.0037 (0.0144)  time: 0.1648  data: 0.1103
max mem: 1419
Test:  [400/418]  eta: 0:00:03  model_time: 0.0439 (0.0439)  evaluator_time: 0.0048 (0.0147)  time: 0.1661  data: 0.1105
max mem: 1419
Test:  [417/418]  eta: 0:00:00  model_time: 0.0438 (0.0439)  evaluator_time: 0.0044 (0.0151)  time: 0.1775  data: 0.1103
max mem: 1419
Test: Total time: 0:01:11 (0.1720 s / it)
Averaged stats: model_time: 0.0438 (0.0439)  evaluator_time: 0.0044 (0.0151)
Accumulating evaluation results...
DONE (t=0.21s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.460
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.754
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.490
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.087
 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 ] = 0.613
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.095
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.414
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.550
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.340
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.507
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.660
Epoch: [7]  [   0/1158]  eta: 0:07:13  lr: 0.000200  loss: 0.2257 (0.2257)  loss_classifier: 0.0709 (0.0709)  loss_box_re
g: 0.1234 (0.1234)  loss_objectness: 0.0181 (0.0181)  loss_rpn_box_reg: 0.0133 (0.0133)  time: 0.3748  data: 0.2674  max m
em: 1419
Epoch: [7]  [1157/1158]  eta: 0:00:00  lr: 0.000200  loss: 0.2742 (0.2929)  loss_classifier: 0.0847 (0.0966)  loss_box_re
g: 0.1719 (0.1628)  loss_objectness: 0.0185 (0.0196)  loss_rpn_box_reg: 0.0103 (0.0140)  time: 0.3692  data: 0.2639  max m
em: 1419
Epoch: [7] Total time: 0:07:13 (0.3740 s / it)
creating index...
index created!
Test:  [  0/418]  eta: 0:01:10  model_time: 0.0440 (0.0440)  evaluator_time: 0.0023 (0.0023)  time: 0.1697  data: 0.1201
max mem: 1419
Test:  [100/418]  eta: 0:00:55  model_time: 0.0438 (0.0439)  evaluator_time: 0.0080 (0.0147)  time: 0.1716  data: 0.1110
max mem: 1419
Test:  [200/418]  eta: 0:00:38  model_time: 0.0443 (0.0441)  evaluator_time: 0.0084 (0.0149)  time: 0.1777  data: 0.1105
max mem: 1419
Test:  [300/418]  eta: 0:00:20  model_time: 0.0444 (0.0442)  evaluator_time: 0.0035 (0.0142)  time: 0.1684  data: 0.1115
max mem: 1419
Test:  [400/418]  eta: 0:00:03  model_time: 0.0443 (0.0443)  evaluator_time: 0.0044 (0.0145)  time: 0.1681  data: 0.1103
max mem: 1419
Test:  [417/418]  eta: 0:00:00  model_time: 0.0444 (0.0443)  evaluator_time: 0.0043 (0.0149)  time: 0.1806  data: 0.1108
max mem: 1419
Test: Total time: 0:01:13 (0.1747 s / it)
Averaged stats: model_time: 0.0444 (0.0443)  evaluator_time: 0.0043 (0.0149)
Accumulating evaluation results...
DONE (t=0.21s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.462
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.753
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.493
 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 ] = 0.393
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.615
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.095
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.415
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.551
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.340
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.509
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.662
Epoch: [8]  [   0/1158]  eta: 0:06:16  lr: 0.000200  loss: 0.3788 (0.3788)  loss_classifier: 0.1467 (0.1467)  loss_box_re
g: 0.2151 (0.2151)  loss_objectness: 0.0075 (0.0075)  loss_rpn_box_reg: 0.0096 (0.0096)  time: 0.3249  data: 0.2115  max m
em: 1419
Epoch: [8]  [1157/1158]  eta: 0:00:00  lr: 0.000200  loss: 0.2915 (0.2913)  loss_classifier: 0.0993 (0.0962)  loss_box_re
g: 0.1625 (0.1622)  loss_objectness: 0.0205 (0.0190)  loss_rpn_box_reg: 0.0087 (0.0140)  time: 0.3737  data: 0.2677  max m
em: 1419
Epoch: [8] Total time: 0:07:17 (0.3774 s / it)
creating index...
index created!
Test:  [  0/418]  eta: 0:01:06  model_time: 0.0450 (0.0450)  evaluator_time: 0.0023 (0.0023)  time: 0.1601  data: 0.1098
max mem: 1419
Test:  [100/418]  eta: 0:00:55  model_time: 0.0441 (0.0442)  evaluator_time: 0.0081 (0.0149)  time: 0.1722  data: 0.1110
max mem: 1419
Test:  [200/418]  eta: 0:00:37  model_time: 0.0442 (0.0442)  evaluator_time: 0.0085 (0.0150)  time: 0.1767  data: 0.1116
max mem: 1419
Test:  [300/418]  eta: 0:00:20  model_time: 0.0443 (0.0442)  evaluator_time: 0.0035 (0.0144)  time: 0.1664  data: 0.1112
max mem: 1419
Test:  [400/418]  eta: 0:00:03  model_time: 0.0441 (0.0442)  evaluator_time: 0.0048 (0.0147)  time: 0.1663  data: 0.1107
max mem: 1419
Test:  [417/418]  eta: 0:00:00  model_time: 0.0441 (0.0442)  evaluator_time: 0.0043 (0.0150)  time: 0.1782  data: 0.1107
max mem: 1419
Test: Total time: 0:01:12 (0.1730 s / it)
```

```
Averaged stats: model_time: 0.0441 (0.0442)  evaluator_time: 0.0043 (0.0150)
Accumulating evaluation results...
DONE (t=0.21s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.462
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.754
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.495
 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 ] = 0.393
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.615
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.096
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.417
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.550
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.340
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.507
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.662
Epoch: [9]  [   0/1158]  eta: 0:07:05  lr: 0.000020  loss: 0.4406 (0.4406)  loss_classifier: 0.1160 (0.1160)  loss_box_re
g: 0.3037 (0.3037)  loss_objectness: 0.0085 (0.0085)  loss_rpn_box_reg: 0.0123 (0.0123)  time: 0.3679  data: 0.2601  max m
em: 1419
Epoch: [9]  [1157/1158]  eta: 0:00:00  lr: 0.000020  loss: 0.2899 (0.2925)  loss_classifier: 0.0889 (0.0965)  loss_box_re
g: 0.1723 (0.1626)  loss_objectness: 0.0087 (0.0194)  loss_rpn_box_reg: 0.0073 (0.0140)  time: 0.3540  data: 0.2481  max m
em: 1419
Epoch: [9] Total time: 0:07:13 (0.3739 s / it)
creating index...
index created!
Test:  [  0/418]  eta: 0:01:08  model_time: 0.0442 (0.0442)  evaluator_time: 0.0023 (0.0023)  time: 0.1650  data: 0.1151
max mem: 1419
Test:  [100/418]  eta: 0:00:54  model_time: 0.0438 (0.0440)  evaluator_time: 0.0077 (0.0148)  time: 0.1739  data: 0.1131
max mem: 1419
Test:  [200/418]  eta: 0:00:37  model_time: 0.0440 (0.0440)  evaluator_time: 0.0087 (0.0149)  time: 0.1749  data: 0.1100
max mem: 1419
Test:  [300/418]  eta: 0:00:20  model_time: 0.0442 (0.0440)  evaluator_time: 0.0035 (0.0143)  time: 0.1653  data: 0.1105
max mem: 1419
Test:  [400/418]  eta: 0:00:03  model_time: 0.0442 (0.0441)  evaluator_time: 0.0045 (0.0146)  time: 0.1704  data: 0.1141
max mem: 1419
Test:  [417/418]  eta: 0:00:00  model_time: 0.0438 (0.0441)  evaluator_time: 0.0042 (0.0150)  time: 0.1774  data: 0.1104
max mem: 1419
Test: Total time: 0:01:12 (0.1724 s / it)
Averaged stats: model_time: 0.0438 (0.0441)  evaluator_time: 0.0042 (0.0150)
Accumulating evaluation results...
DONE (t=0.20s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.461
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.754
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.492
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.087
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.392
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.616
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.095
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.417
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.550
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.340
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.506
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.663
```

Training time on Nvidia Tesla V100 with 16 GB memory

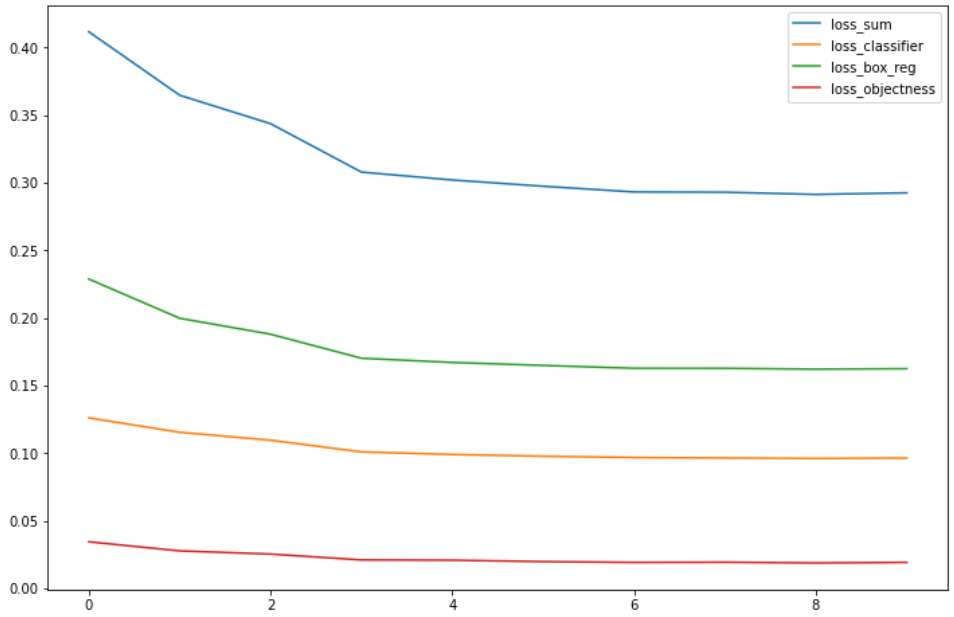
In [32]:

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

```
Time elapsed = 92.22 min
```

In [33]:

```python
fasterutils.show_losses(all_mean_losses)
```



Results:

```
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.461
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.754
```

```
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.492
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.087
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.392
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.616
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.095
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.417
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.550
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.340
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.506
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.663
```

Error Analysis

Ground truth

In [34]:

```
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 [35]:

```
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] *
    print('Image was flipped: %r' % img_was_flipped)

    return img
```

In [36]:

```
## 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[36]:

Predicted

In [37]:

```
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 [38]:

```
show(0, imgs_test)
```

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

Export for testing

In [39]:

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

In [40]:

```
save_model(model, path="../data/models/final-model-fit.pt")
```

```
Model saved to ../data/models/final-model-fit.pt
```

In [41]:

```
# def load_model(path="./models/entire_model.pt"):
#     if torch.cuda.is_available():
#         return torch.load(path)
#     else:
#         return torch.load(path, map_location=torch.device('cpu'))
# model = load_model(path="../data/models/final-model-fit.pt")
```

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.

For the 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. For final test, I trained it on the complete training set and used the complete test set of 500 images.

In [42]:

```
len(dataset_test), len(imgs_test)
```

Out[42]:

```
(418, 418)
```

In [43]:

```
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 418 predictions

Time elapsed = 1.04 min
```

In [44]:
```
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 [48]:
```
import pickle
with open(r"../data/predictions-final.pickle", "wb") as output_file:
    pickle.dump(predictions2, output_file)
```

In [ ]: