

HW4_MOT_SORT_Person_Detection

May 5, 2023

0.1 CV HW4: Multi-object Tracking (MOT) with Detection

Detection: YOLOv5, Tracking: Simple Online Realtime Tracking (SORT)

0.2 1. Unzip data folder

```
[ ]: from google.colab import drive  
drive.mount("/content/drive")
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).

```
[ ]: # Change the path according to your setup  
!unzip '/content/drive/MyDrive/ECE344: CV (Computer Vision)/Assignments/  
↳Assignment-4/hw4/sort-master.zip'  
!unzip '/content/drive/MyDrive/ECE344: CV (Computer Vision)/Assignments/  
↳Assignment-4/hw4/KITTI_17_images.zip'
```

```
Archive: /content/drive/MyDrive/ECE344: CV (Computer  
Vision)/Assignments/Assignment-4/hw4/sort-master.zip  
replace sort-master/.gitignore? [y]es, [n]o, [A]ll, [N]one, [r]ename: A  
extracting: sort-master/.gitignore  
inflating: sort-master/data/train/ADL-Rundle-6/det/det.txt  
inflating: sort-master/data/train/ADL-Rundle-8/det/det.txt  
inflating: sort-master/data/train/ETH-Bahnhof/det/det.txt  
inflating: sort-master/data/train/ETH-Pedcross2/det/det.txt  
inflating: sort-master/data/train/ETH-Sunnyday/det/det.txt  
inflating: sort-master/data/train/KITTI-13/det/det.txt  
inflating: sort-master/data/train/KITTI-17/det/det.txt  
inflating: sort-master/data/train/PETS09-S2L1/det/det.txt  
inflating: sort-master/data/train/TUD-Campus/det/det.txt  
inflating: sort-master/data/train/TUD-Stadtmitte/det/det.txt  
inflating: sort-master/data/train/Venice-2/det/det.txt  
inflating: sort-master/LICENSE  
inflating: sort-master/README.md  
inflating: sort-master/requirements.txt  
inflating: sort-master/sort.py  
Archive: /content/drive/MyDrive/ECE344: CV (Computer
```

```
Vision)/Assignments/Assignment-4/hw4/KITTI_17_images.zip  
replace KITTI_17_images/000001.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: A  
    inflating: KITTI_17_images/000001.jpg  
    inflating: KITTI_17_images/000002.jpg  
    inflating: KITTI_17_images/000003.jpg  
    inflating: KITTI_17_images/000004.jpg  
    inflating: KITTI_17_images/000005.jpg  
    inflating: KITTI_17_images/000006.jpg  
    inflating: KITTI_17_images/000007.jpg  
    inflating: KITTI_17_images/000008.jpg  
    inflating: KITTI_17_images/000009.jpg  
    inflating: KITTI_17_images/000010.jpg  
    inflating: KITTI_17_images/000011.jpg  
    inflating: KITTI_17_images/000012.jpg  
    inflating: KITTI_17_images/000013.jpg  
    inflating: KITTI_17_images/000014.jpg  
    inflating: KITTI_17_images/000015.jpg  
    inflating: KITTI_17_images/000016.jpg  
    inflating: KITTI_17_images/000017.jpg  
    inflating: KITTI_17_images/000018.jpg  
    inflating: KITTI_17_images/000019.jpg  
    inflating: KITTI_17_images/000020.jpg  
    inflating: KITTI_17_images/000021.jpg  
    inflating: KITTI_17_images/000022.jpg  
    inflating: KITTI_17_images/000023.jpg  
    inflating: KITTI_17_images/000024.jpg  
    inflating: KITTI_17_images/000025.jpg  
    inflating: KITTI_17_images/000026.jpg  
    inflating: KITTI_17_images/000027.jpg  
    inflating: KITTI_17_images/000028.jpg  
    inflating: KITTI_17_images/000029.jpg  
    inflating: KITTI_17_images/000030.jpg  
    inflating: KITTI_17_images/000031.jpg  
    inflating: KITTI_17_images/000032.jpg  
    inflating: KITTI_17_images/000033.jpg  
    inflating: KITTI_17_images/000034.jpg  
    inflating: KITTI_17_images/000035.jpg  
    inflating: KITTI_17_images/000036.jpg  
    inflating: KITTI_17_images/000037.jpg  
    inflating: KITTI_17_images/000038.jpg  
    inflating: KITTI_17_images/000039.jpg  
    inflating: KITTI_17_images/000040.jpg  
    inflating: KITTI_17_images/000041.jpg  
    inflating: KITTI_17_images/000042.jpg  
    inflating: KITTI_17_images/000043.jpg  
    inflating: KITTI_17_images/000044.jpg  
    inflating: KITTI_17_images/000045.jpg  
    inflating: KITTI_17_images/000046.jpg
```

inflating: KITTI_17_images/000047.jpg
inflating: KITTI_17_images/000048.jpg
inflating: KITTI_17_images/000049.jpg
inflating: KITTI_17_images/000050.jpg
inflating: KITTI_17_images/000051.jpg
inflating: KITTI_17_images/000052.jpg
inflating: KITTI_17_images/000053.jpg
inflating: KITTI_17_images/000054.jpg
inflating: KITTI_17_images/000055.jpg
inflating: KITTI_17_images/000056.jpg
inflating: KITTI_17_images/000057.jpg
inflating: KITTI_17_images/000058.jpg
inflating: KITTI_17_images/000059.jpg
inflating: KITTI_17_images/000060.jpg
inflating: KITTI_17_images/000061.jpg
inflating: KITTI_17_images/000062.jpg
inflating: KITTI_17_images/000063.jpg
inflating: KITTI_17_images/000064.jpg
inflating: KITTI_17_images/000065.jpg
inflating: KITTI_17_images/000066.jpg
inflating: KITTI_17_images/000067.jpg
inflating: KITTI_17_images/000068.jpg
inflating: KITTI_17_images/000069.jpg
inflating: KITTI_17_images/000070.jpg
inflating: KITTI_17_images/000071.jpg
inflating: KITTI_17_images/000072.jpg
inflating: KITTI_17_images/000073.jpg
inflating: KITTI_17_images/000074.jpg
inflating: KITTI_17_images/000075.jpg
inflating: KITTI_17_images/000076.jpg
inflating: KITTI_17_images/000077.jpg
inflating: KITTI_17_images/000078.jpg
inflating: KITTI_17_images/000079.jpg
inflating: KITTI_17_images/000080.jpg
inflating: KITTI_17_images/000081.jpg
inflating: KITTI_17_images/000082.jpg
inflating: KITTI_17_images/000083.jpg
inflating: KITTI_17_images/000084.jpg
inflating: KITTI_17_images/000085.jpg
inflating: KITTI_17_images/000086.jpg
inflating: KITTI_17_images/000087.jpg
inflating: KITTI_17_images/000088.jpg
inflating: KITTI_17_images/000089.jpg
inflating: KITTI_17_images/000090.jpg
inflating: KITTI_17_images/000091.jpg
inflating: KITTI_17_images/000092.jpg
inflating: KITTI_17_images/000093.jpg
inflating: KITTI_17_images/000094.jpg

inflating: KITTI_17_images/000095.jpg
inflating: KITTI_17_images/000096.jpg
inflating: KITTI_17_images/000097.jpg
inflating: KITTI_17_images/000098.jpg
inflating: KITTI_17_images/000099.jpg
inflating: KITTI_17_images/000100.jpg
inflating: KITTI_17_images/000101.jpg
inflating: KITTI_17_images/000102.jpg
inflating: KITTI_17_images/000103.jpg
inflating: KITTI_17_images/000104.jpg
inflating: KITTI_17_images/000105.jpg
inflating: KITTI_17_images/000106.jpg
inflating: KITTI_17_images/000107.jpg
inflating: KITTI_17_images/000108.jpg
inflating: KITTI_17_images/000109.jpg
inflating: KITTI_17_images/000110.jpg
inflating: KITTI_17_images/000111.jpg
inflating: KITTI_17_images/000112.jpg
inflating: KITTI_17_images/000113.jpg
inflating: KITTI_17_images/000114.jpg
inflating: KITTI_17_images/000115.jpg
inflating: KITTI_17_images/000116.jpg
inflating: KITTI_17_images/000117.jpg
inflating: KITTI_17_images/000118.jpg
inflating: KITTI_17_images/000119.jpg
inflating: KITTI_17_images/000120.jpg
inflating: KITTI_17_images/000121.jpg
inflating: KITTI_17_images/000122.jpg
inflating: KITTI_17_images/000123.jpg
inflating: KITTI_17_images/000124.jpg
inflating: KITTI_17_images/000125.jpg
inflating: KITTI_17_images/000126.jpg
inflating: KITTI_17_images/000127.jpg
inflating: KITTI_17_images/000128.jpg
inflating: KITTI_17_images/000129.jpg
inflating: KITTI_17_images/000130.jpg
inflating: KITTI_17_images/000131.jpg
inflating: KITTI_17_images/000132.jpg
inflating: KITTI_17_images/000133.jpg
inflating: KITTI_17_images/000134.jpg
inflating: KITTI_17_images/000135.jpg
inflating: KITTI_17_images/000136.jpg
inflating: KITTI_17_images/000137.jpg
inflating: KITTI_17_images/000138.jpg
inflating: KITTI_17_images/000139.jpg
inflating: KITTI_17_images/000140.jpg
inflating: KITTI_17_images/000141.jpg
inflating: KITTI_17_images/000142.jpg

```
inflating: KITTI_17_images/000143.jpg
inflating: KITTI_17_images/000144.jpg
inflating: KITTI_17_images/000145.jpg
```

1 2. Install requirements

```
[ ]: # !pip install -r sort-master/requirements.txt
# !pip install cv
```

```
[ ]: !pip install filterpy
!pip install scikit-image
!pip install lap
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: filterpy in /usr/local/lib/python3.10/dist-packages (1.4.5)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from filterpy) (1.22.4)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from filterpy) (3.7.1)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from filterpy) (1.10.1)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->filterpy) (23.1)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->filterpy) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->filterpy) (8.4.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->filterpy) (1.0.7)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->filterpy) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->filterpy) (1.4.4)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->filterpy) (3.0.9)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->filterpy) (4.39.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib->filterpy) (1.16.0)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: scikit-image in /usr/local/lib/python3.10/dist-packages (0.19.3)
Requirement already satisfied: PyWavelets>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from scikit-image) (1.4.1)
```

```
Requirement already satisfied: pillow!=7.1.0,!=7.1.1,!=8.3.0,>=6.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (8.4.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (23.1)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (1.10.1)
Requirement already satisfied: numpy>=1.17.0 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (1.22.4)
Requirement already satisfied: imageio>=2.4.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (2.25.1)
Requirement already satisfied: networkx>=2.2 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (3.1)
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (2023.4.12)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: lap in /usr/local/lib/python3.10/dist-packages (0.4.0)
```

2 3. Import libraries

```
[ ]: import torch
import torchvision
import cv2
import sys
sys.path.insert(0,'./sort-master/')
import matplotlib
from google.colab.patches import cv2_imshow
from collections import namedtuple, OrderedDict
```

3 4. Load YOLOv5 detector from torch hub

```
[ ]: yolov5_detector = torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained = True)
yolov5_detector.float()
yolov5_detector.eval()
```

```
Using cache found in /root/.cache/torch/hub.ultralytics_yolov5_master
YOLOv5 2023-5-4 Python-3.10.11 torch-2.0.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
```

```
requirements: /root/.cache/torch/hub/requirements.txt not found,
check failed.
```

```
Fusing layers...
YOLOv5s summary: 213 layers, 7225885 parameters, 0 gradients
Adding AutoShape...
```

```

[ ]: AutoShape(
    (model): DetectMultiBackend(
        (model): DetectionModel(
            (model): Sequential(
                (0): Conv(
                    (conv): Conv2d(3, 32, kernel_size=(6, 6), stride=(2, 2), padding=(2,
2))
                    (act): SiLU(inplace=True)
                )
                (1): Conv(
                    (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1))
                    (act): SiLU(inplace=True)
                )
                (2): C3(
                    (cv1): Conv(
                        (conv): Conv2d(64, 32, kernel_size=(1, 1), stride=(1, 1))
                        (act): SiLU(inplace=True)
                    )
                    (cv2): Conv(
                        (conv): Conv2d(64, 32, kernel_size=(1, 1), stride=(1, 1))
                        (act): SiLU(inplace=True)
                    )
                    (cv3): Conv(
                        (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))
                        (act): SiLU(inplace=True)
                    )
                )
                (m): Sequential(
                    (0): Bottleneck(
                        (cv1): Conv(
                            (conv): Conv2d(32, 32, kernel_size=(1, 1), stride=(1, 1))
                            (act): SiLU(inplace=True)
                        )
                        (cv2): Conv(
                            (conv): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                            (act): SiLU(inplace=True)
                        )
                    )
                )
                (3): Conv(
                    (conv): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1))
                    (act): SiLU(inplace=True)
                )
                (4): C3(

```

```

(cv1): Conv(
    (conv): Conv2d(128, 64, kernel_size=(1, 1), stride=(1, 1))
    (act): SiLU(inplace=True)
)
(cv2): Conv(
    (conv): Conv2d(128, 64, kernel_size=(1, 1), stride=(1, 1))
    (act): SiLU(inplace=True)
)
(cv3): Conv(
    (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
    (act): SiLU(inplace=True)
)
(m): Sequential(
    (0): Bottleneck(
        (cv1): Conv(
            (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
        )
        (cv2): Conv(
            (conv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
            (act): SiLU(inplace=True)
        )
    )
    (1): Bottleneck(
        (cv1): Conv(
            (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
        )
        (cv2): Conv(
            (conv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
            (act): SiLU(inplace=True)
        )
    )
)
(5): Conv(
    (conv): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))
    (act): SiLU(inplace=True)
)
(6): C3(
    (cv1): Conv(
        (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
)

```

```

(cv2): Conv(
    (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
    (act): SiLU(inplace=True)
)
(cv3): Conv(
    (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
    (act): SiLU(inplace=True)
)
(m): Sequential(
    (0): Bottleneck(
        (cv1): Conv(
            (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
        )
        (cv2): Conv(
            (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
            (act): SiLU(inplace=True)
        )
    )
    (1): Bottleneck(
        (cv1): Conv(
            (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
        )
        (cv2): Conv(
            (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
            (act): SiLU(inplace=True)
        )
    )
    (2): Bottleneck(
        (cv1): Conv(
            (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
        )
        (cv2): Conv(
            (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
            (act): SiLU(inplace=True)
        )
    )
)
(7): Conv(
    (conv): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))

```

```

        (act): SiLU(inplace=True)
    )
(8): C3(
    (cv1): Conv(
        (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (cv2): Conv(
        (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (cv3): Conv(
        (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
)
(m): Sequential(
    (0): Bottleneck(
        (cv1): Conv(
            (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
        )
        (cv2): Conv(
            (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
            (act): SiLU(inplace=True)
        )
    )
)
)
)
)
(9): SPPF(
    (cv1): Conv(
        (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (cv2): Conv(
        (conv): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
)
(m): MaxPool2d(kernel_size=5, stride=1, padding=2, dilation=1,
ceil_mode=False)
)
(10): Conv(
    (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
    (act): SiLU(inplace=True)
)
(11): Upsample(scale_factor=2.0, mode='nearest')
(12): Concat()

```

```

(13): C3(
    (cv1): Conv(
        (conv): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (cv2): Conv(
        (conv): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (cv3): Conv(
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (m): Sequential(
        (0): Bottleneck(
            (cv1): Conv(
                (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
            )
            (cv2): Conv(
                (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
            )
        )
    )
)
(14): Conv(
    (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
    (act): SiLU(inplace=True)
)
(15): Upsample(scale_factor=2.0, mode='nearest')
(16): Concat()
(17): C3(
    (cv1): Conv(
        (conv): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (cv2): Conv(
        (conv): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (cv3): Conv(
        (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
    )
    (m): Sequential(

```



```
(conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))
      (act): SiLU(inplace=True)
    )
  (22): Concat()
  (23): C3(
    (cv1): Conv(
      (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
      (act): SiLU(inplace=True)
    )
    (cv2): Conv(
      (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
      (act): SiLU(inplace=True)
    )
    (cv3): Conv(
      (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1))
      (act): SiLU(inplace=True)
    )
  )
  (m): Sequential(
    (0): Bottleneck(
      (cv1): Conv(
        (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
      )
      (cv2): Conv(
        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
        (act): SiLU(inplace=True)
      )
    )
  )
  (24): Detect(
    (m): ModuleList(
      (0): Conv2d(128, 255, kernel_size=(1, 1), stride=(1, 1))
      (1): Conv2d(256, 255, kernel_size=(1, 1), stride=(1, 1))
      (2): Conv2d(512, 255, kernel_size=(1, 1), stride=(1, 1))
    )
  )
)
```

4 5. Import SORT library

```
[ ]: from sort import *
```

#6. Perform tracking with detection

```
[ ]: # Write your code here to perform tracking with detection using the provided
    ↪YOLOv5 model and the SORT implementation
```

```
import os
import cv2
import numpy as np
import torch
import torchvision
from google.colab.patches import cv2_imshow
import glob
```

```
[ ]: # Define the paths to the input images and output files
output_video_file_path = '/content/drive/MyDrive/ECE344: CV (Computer Vision)/
    ↪Assignments/Assignment-4/output_video.mp4'
output_ground_truth_file_path = '/content/drive/MyDrive/ECE344: CV (Computer
    ↪Vision)/Assignments/Assignment-4/output_boxes_file.txt'
```

```
# Load the input images
all_image_paths = sorted(glob.glob('/content/KITTI_17_images/*.jpg'))
```

```
# Create a SORT object
mot_tracker = Sort()
```

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
yolov5_detector.to(device)
```

```
# Video resolution and frame rate for writing into the .mp4 file
frame_rate = 11
frame_size = (1224, 370)
```

```
# Create a Video writer
video_writer = cv2.VideoWriter(output_video_file_path, cv2.
    ↪VideoWriter_fourcc(*'DIVX'), frame_rate, frame_size)
```

```
# Detect persons in each image of the directory
image_number = 0
```

```
# Store the colors used for bounding box for each unique person
all_colors = {}
```

```
with open(output_ground_truth_file_path, 'w') as f:
    pass
```

```

for image_path in all_image_paths:
    image = cv2.imread(image_path)
    # print(image.shape)
    image = cv2.resize(image, (1224, 370))
    # print(image.shape)

    results = yolov5_detector(image)

    # Extract the bounding box coordinates, class labels and confidence scores
    # of the detected objects
    boundaries_scores = results.xyxy[0].cpu().numpy()[:, :4]
    class_labels = []
    for label in results.xyxy[0].cpu().numpy()[:, 5]:
        class_labels.append(results.names[label])
    confidence_scores = results.xyxy[0].cpu().numpy()[:, 4]

    # Filter the detected objects to only include pedestrians
    detected_persons = []
    for i in range(len(boundaries_scores)):
        if class_labels[i] == 'person' and confidence_scores[i] > 0:
            detected_persons.append(boundaries_scores[i])
    detected_persons = np.array(detected_persons)

    # Using Inbuilt SORT Algorithm to detect/track the persons
    person_trackers = mot_tracker.update(detected_persons)

    # Plot the track IDs and draw the bounding boxes for every person in the
    # image
    for i, person_tracker in enumerate(person_trackers):
        x1, y1, x2, y2, track_id = person_tracker.astype(np.int32)
        if track_id in all_colors.keys():
            color = all_colors[track_id]
        else:
            color = list(np.random.random(size=3) * 256)
            all_colors[track_id] = color
        cv2.rectangle(image, (x1, y1), (x2, y2), color, 2)
        cv2.putText(image, 'ID: '+str(track_id), (x1, y1 - 10), cv2.
        FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

    # In the ground truth output file, save all the bounding box
    # coordinates
    with open(output_ground_truth_file_path, 'a') as f:
        f.write('{},{},{},{},{},{},-1,-1,-1\n'.format(image_number+1,
        track_id, x1, y1, x2 - x1, y2 - y1))

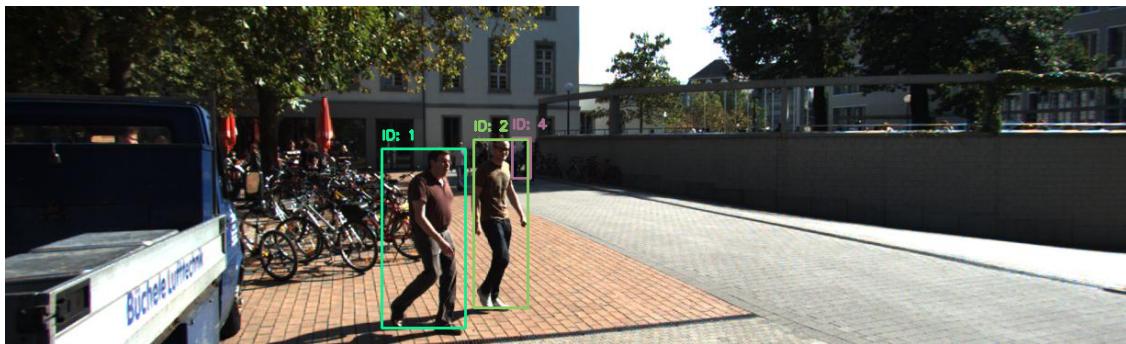
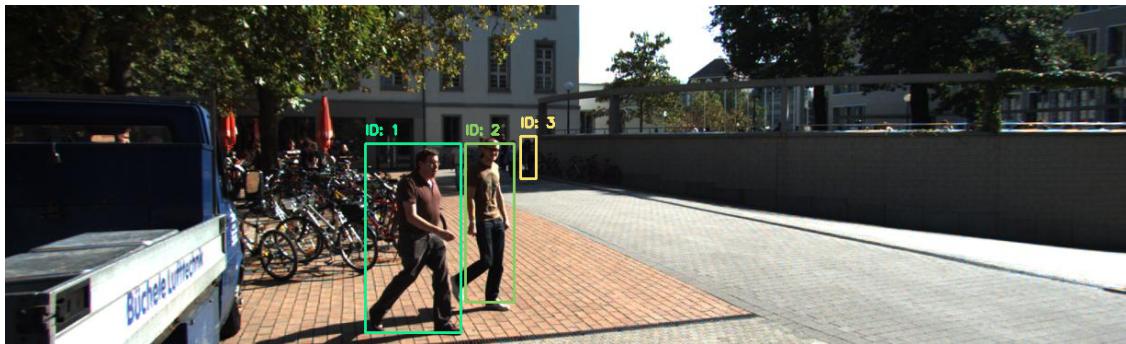
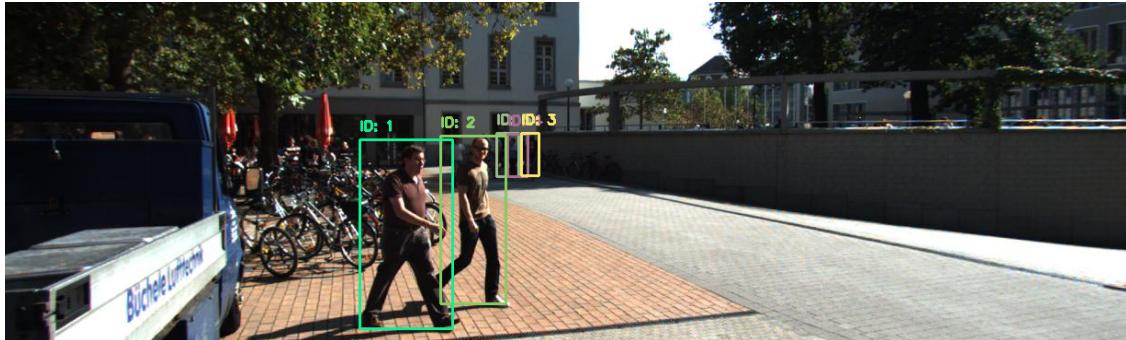
    # Plot the image and write the image-frame to the output video file

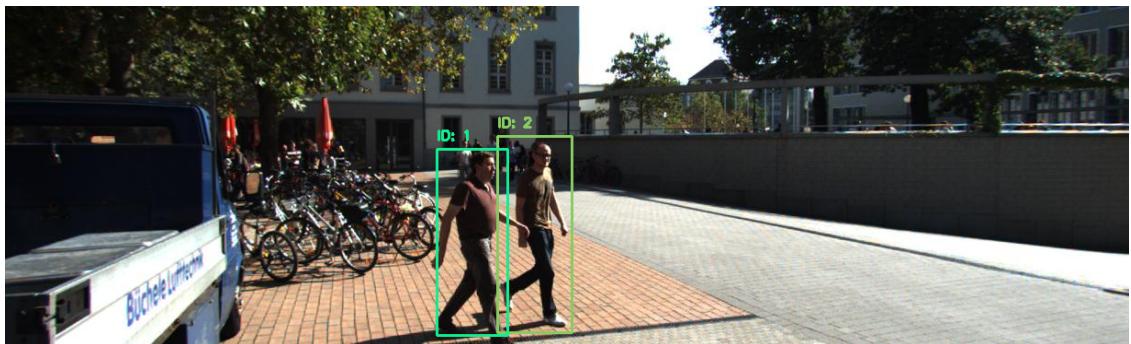
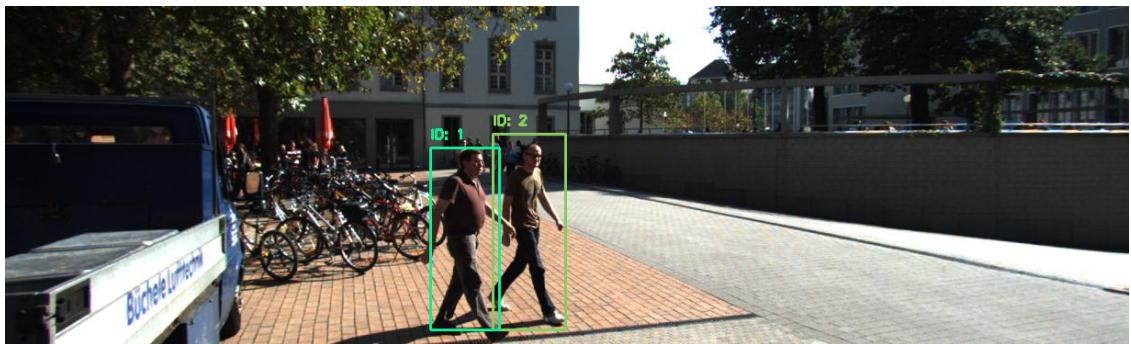
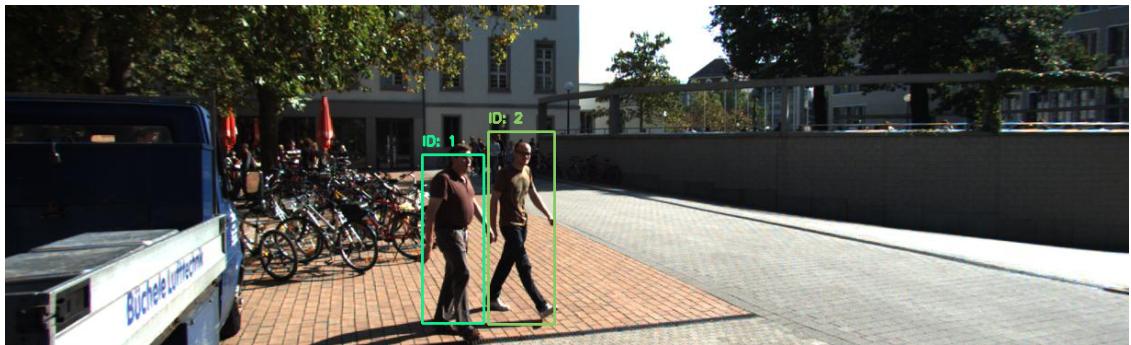
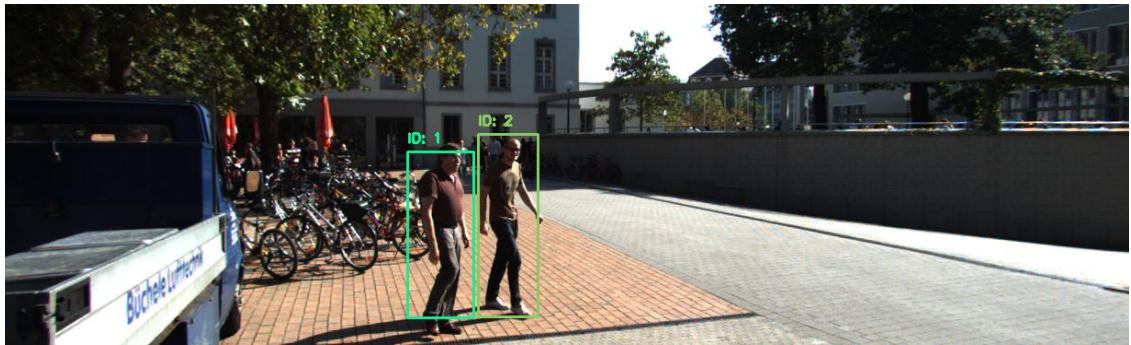
```

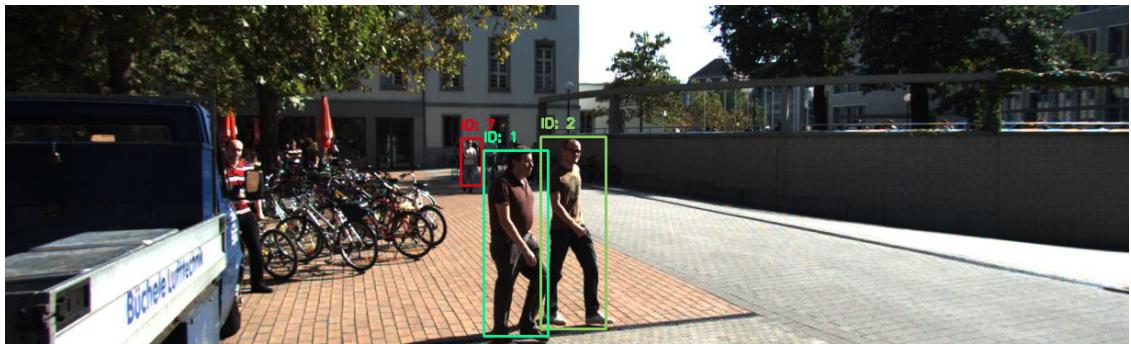
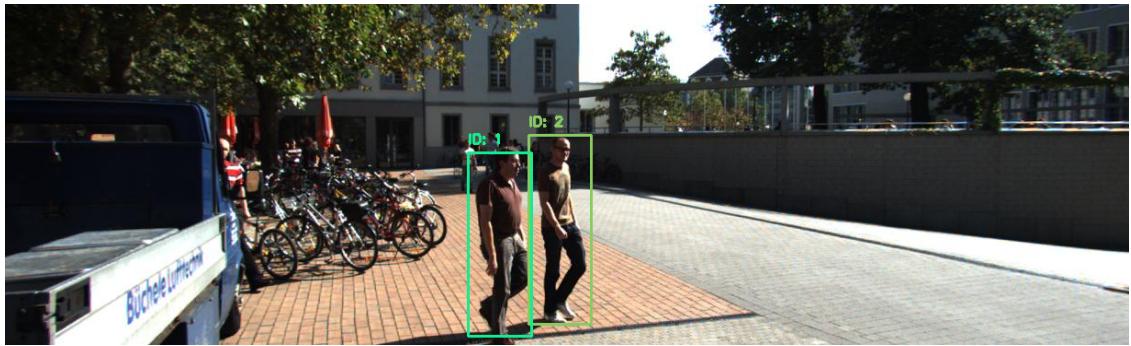
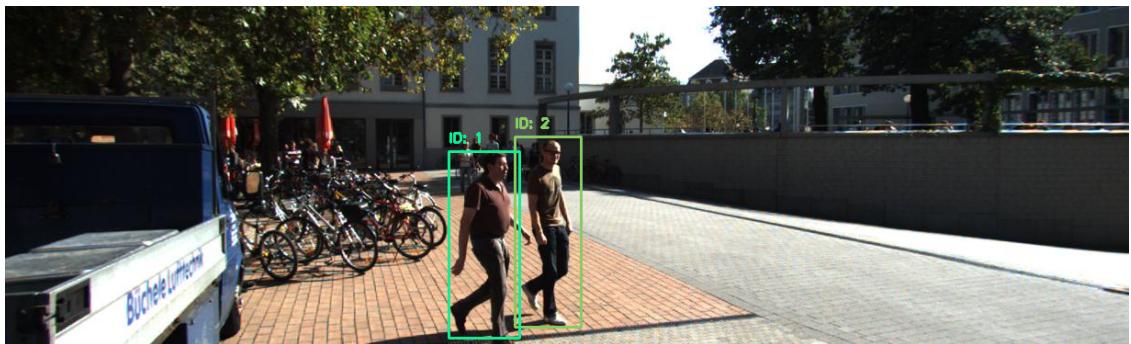
```
video_writer.write(image)
cv2_imshow(image)

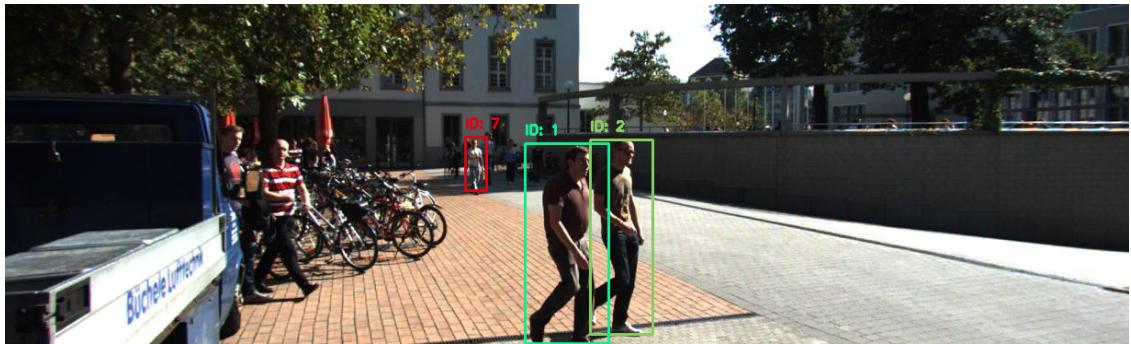
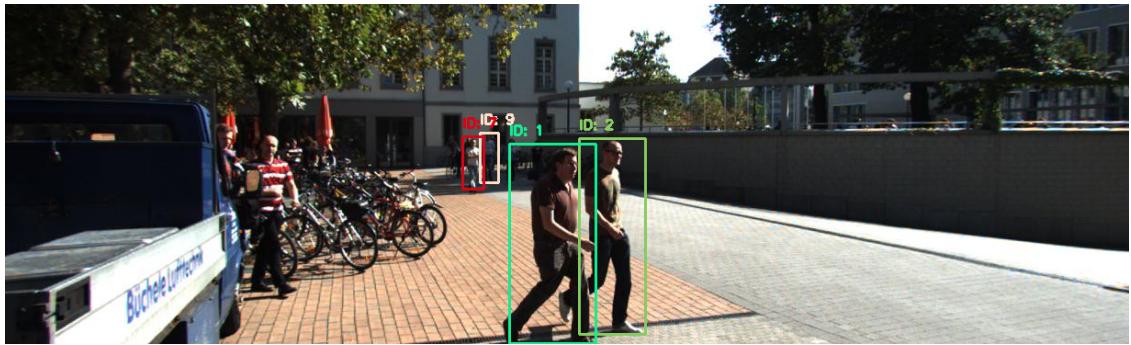
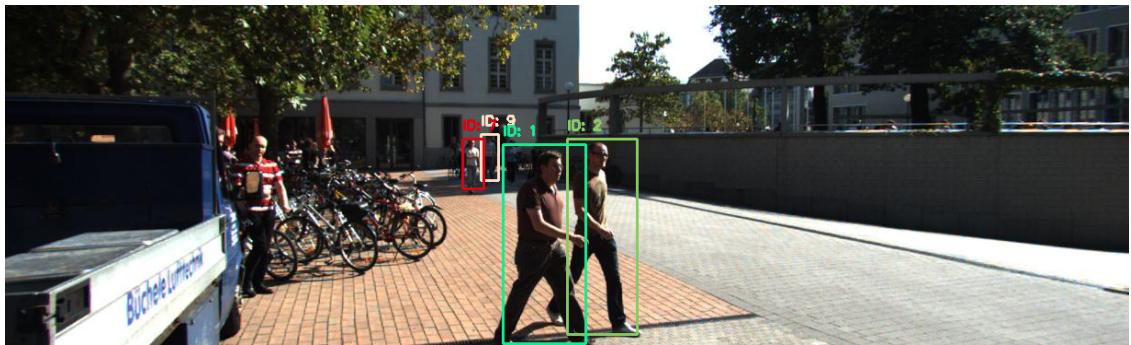
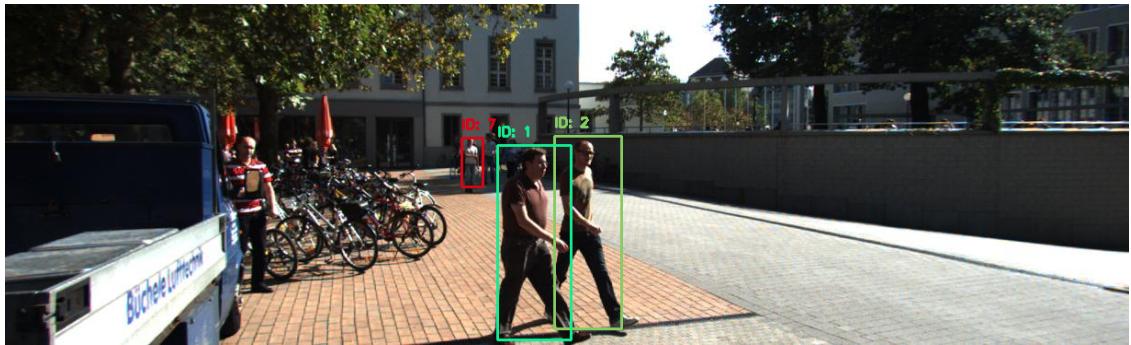
# Increment frame count for every image
image_number += 1

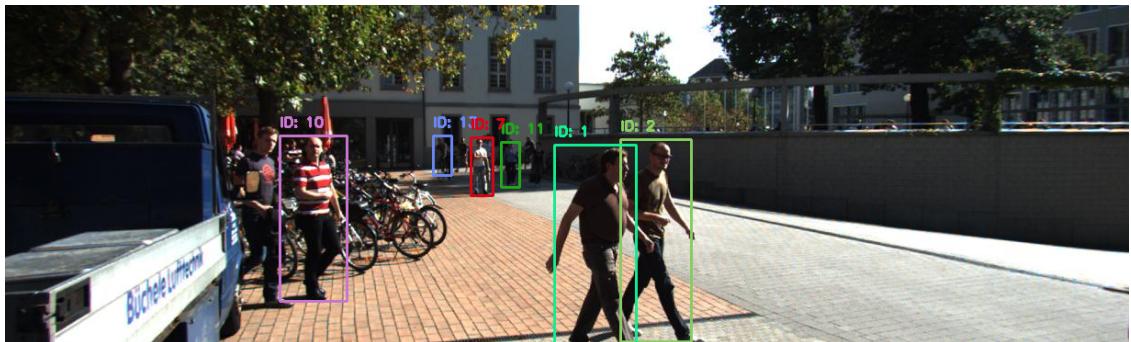
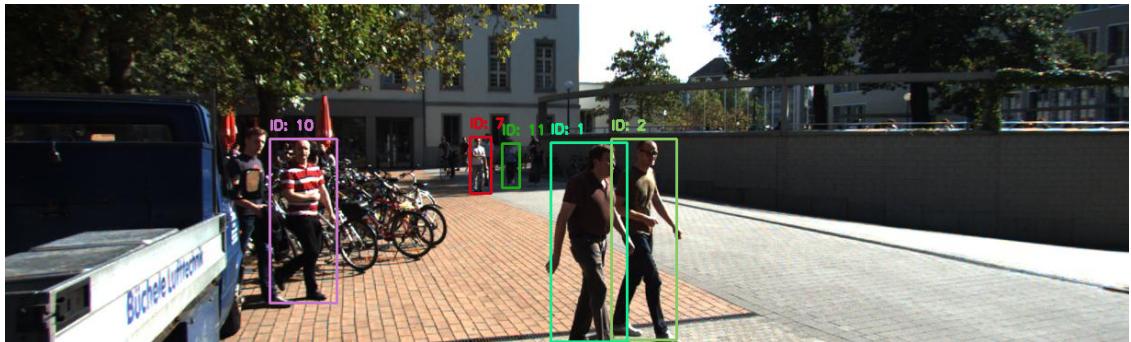
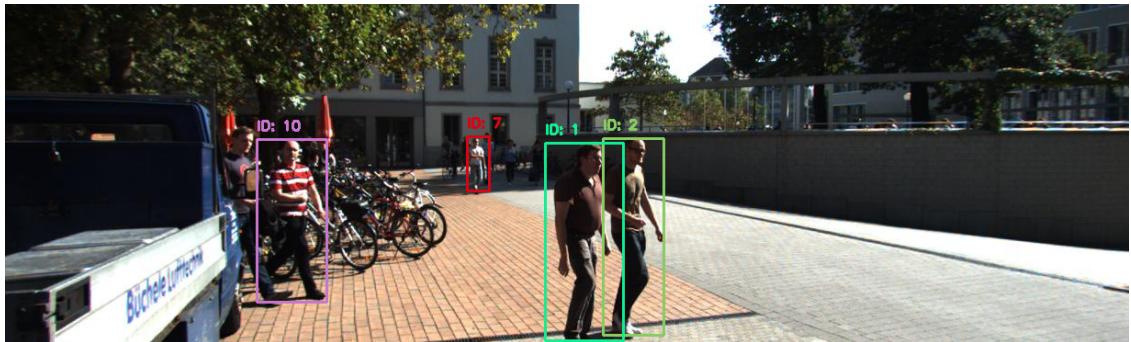
# Save the video
video_writer.release()
```

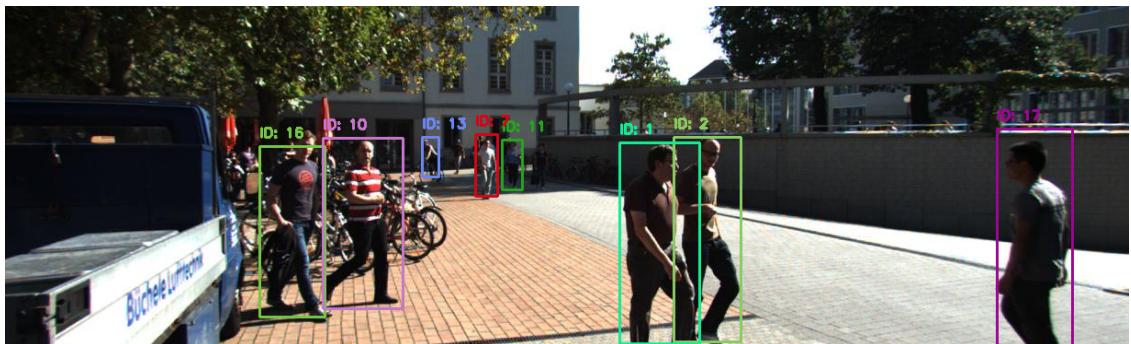
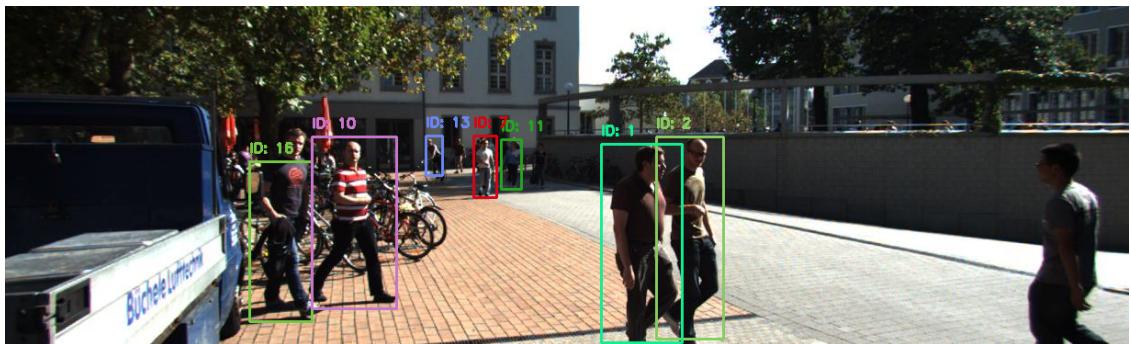
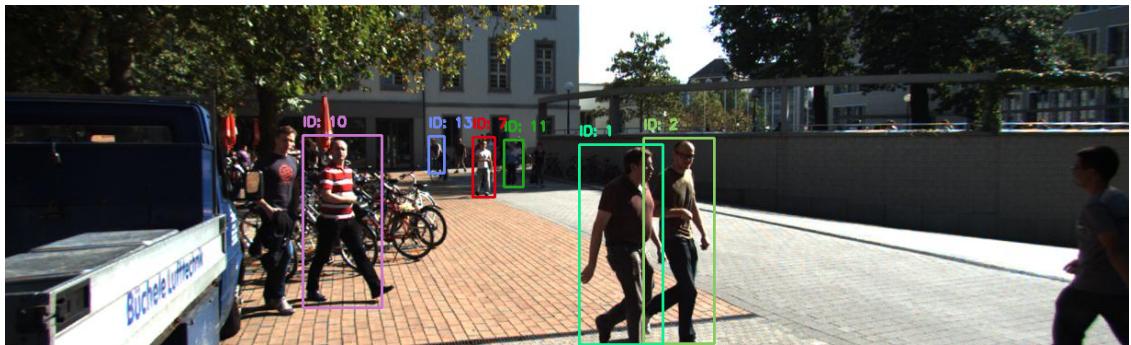
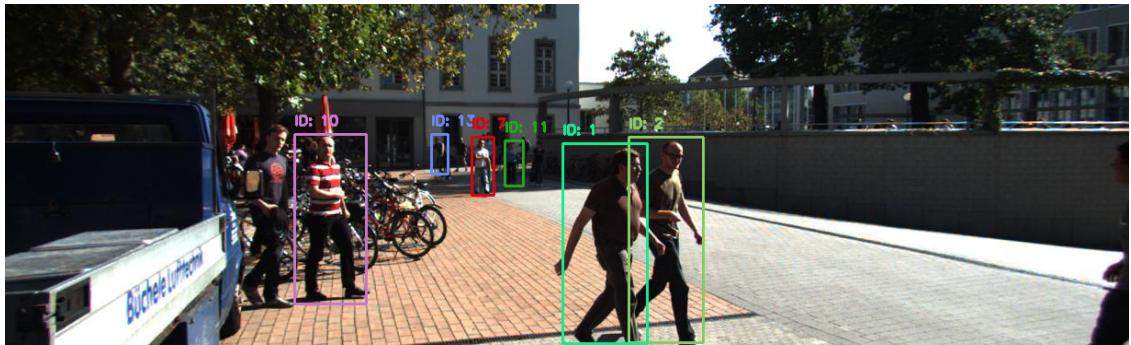


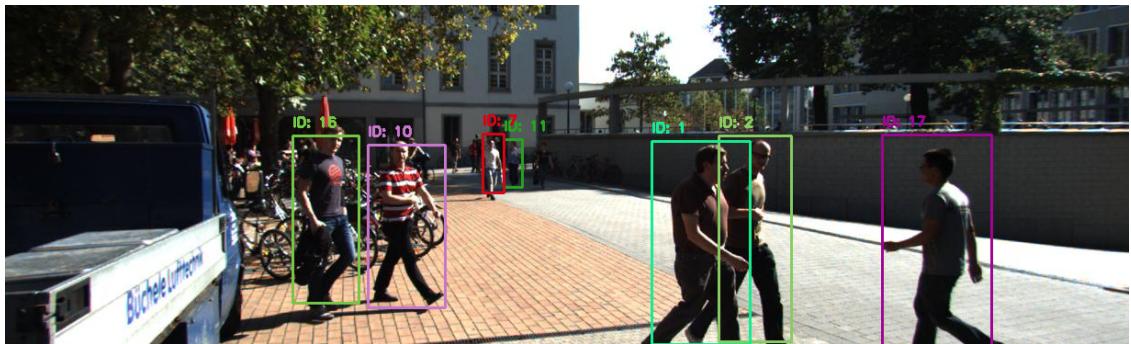
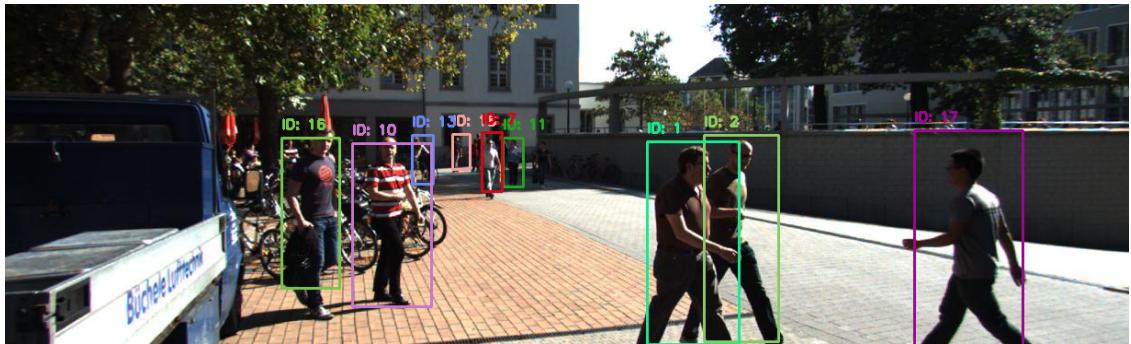
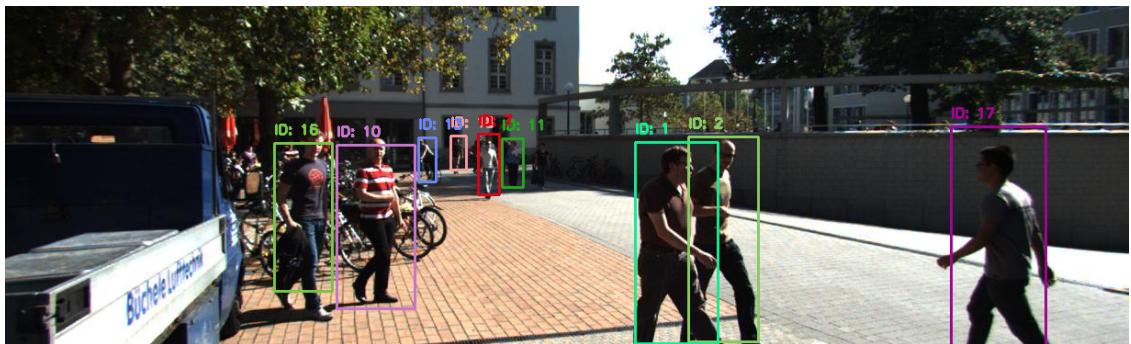


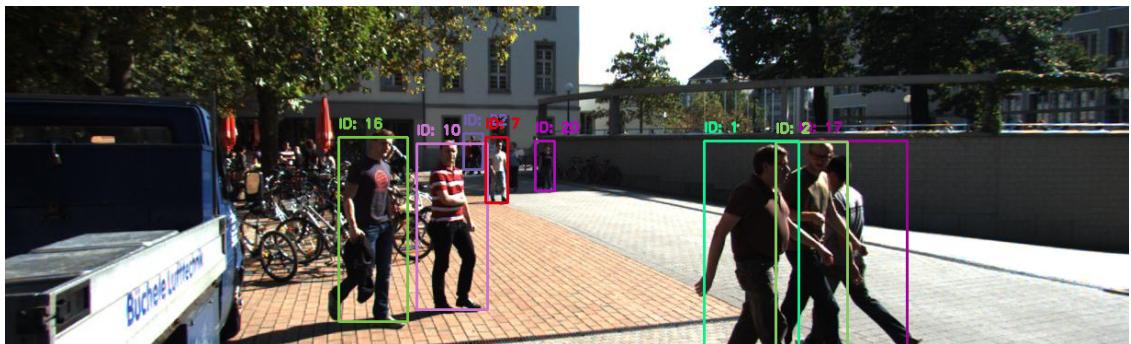
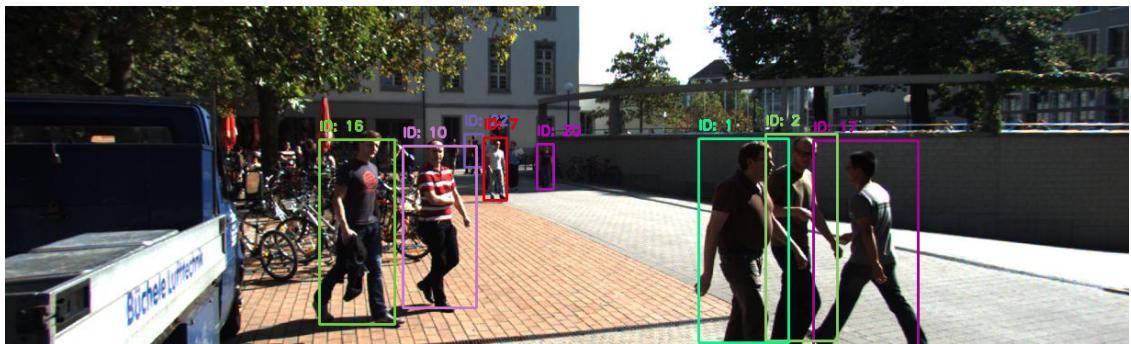
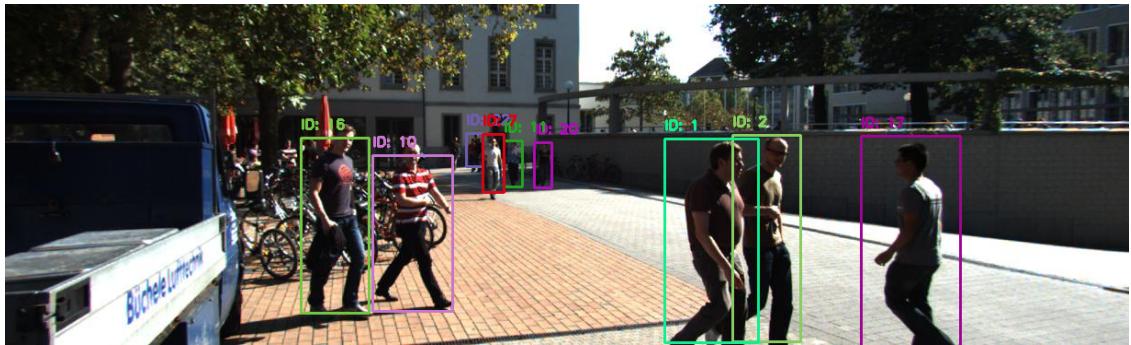


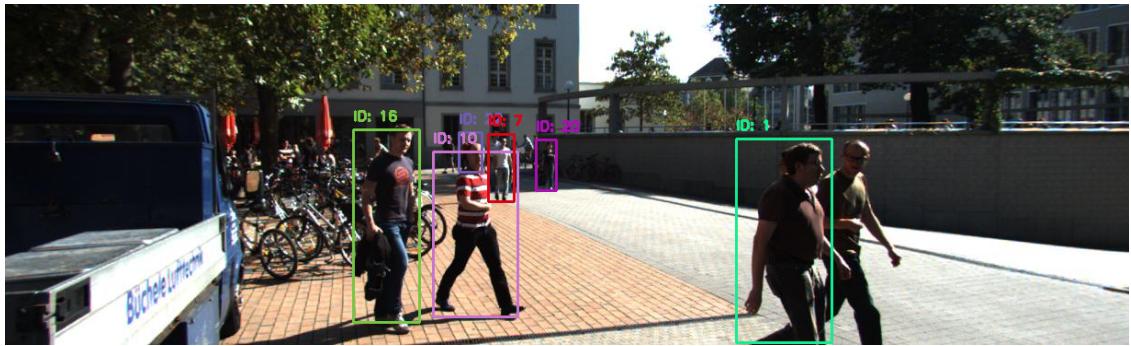
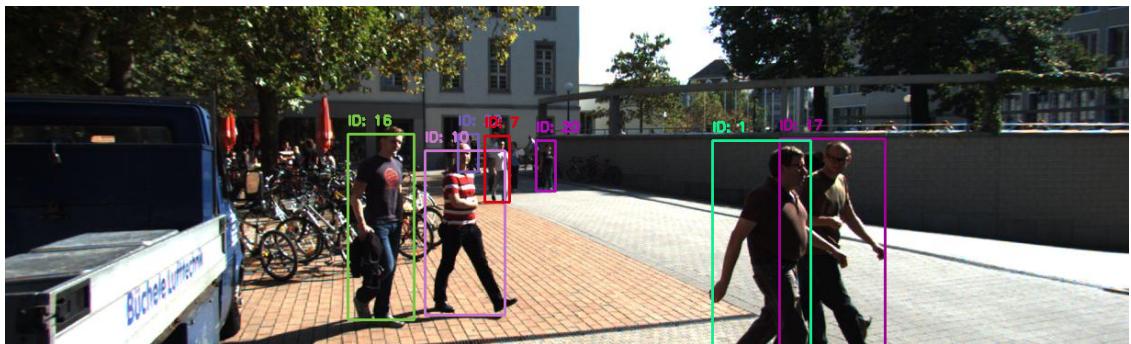


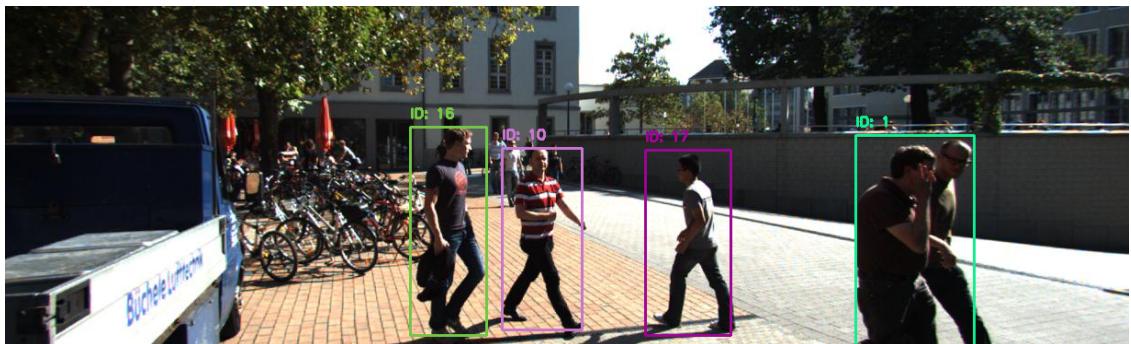
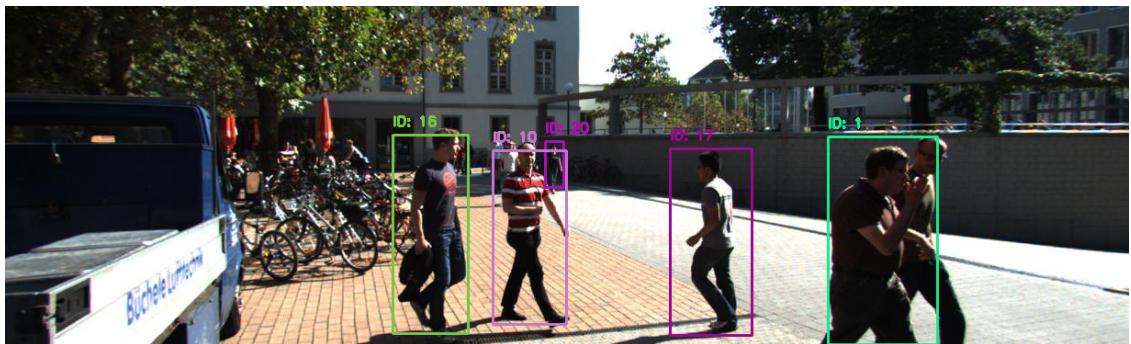
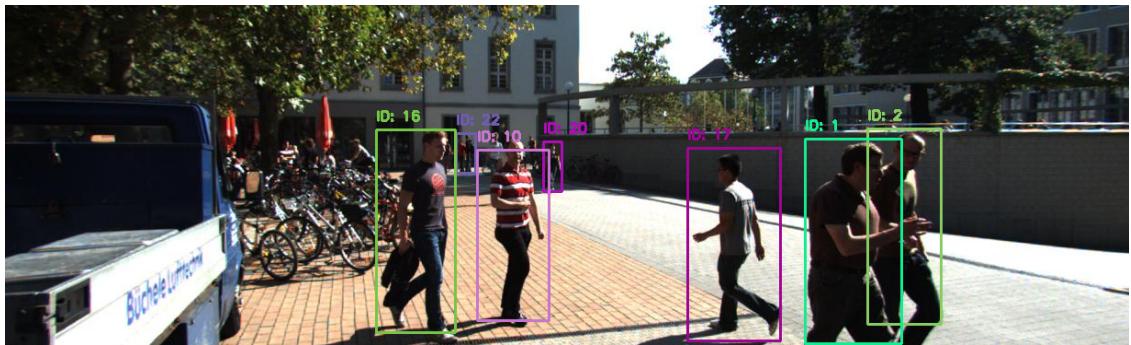
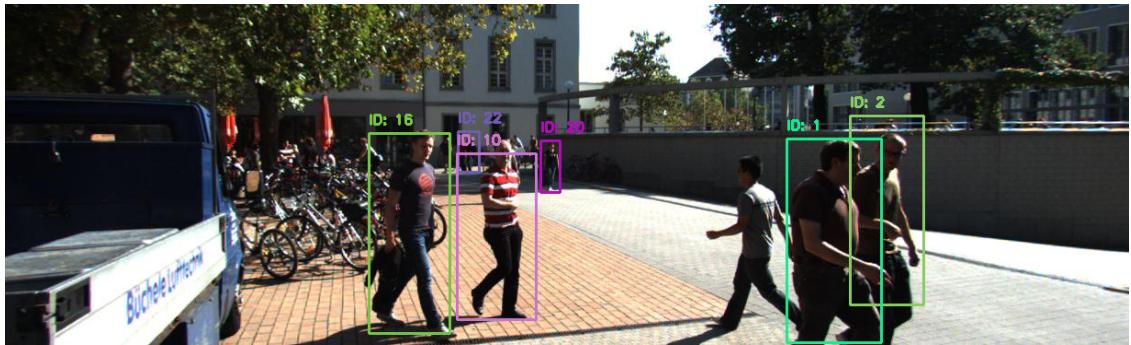


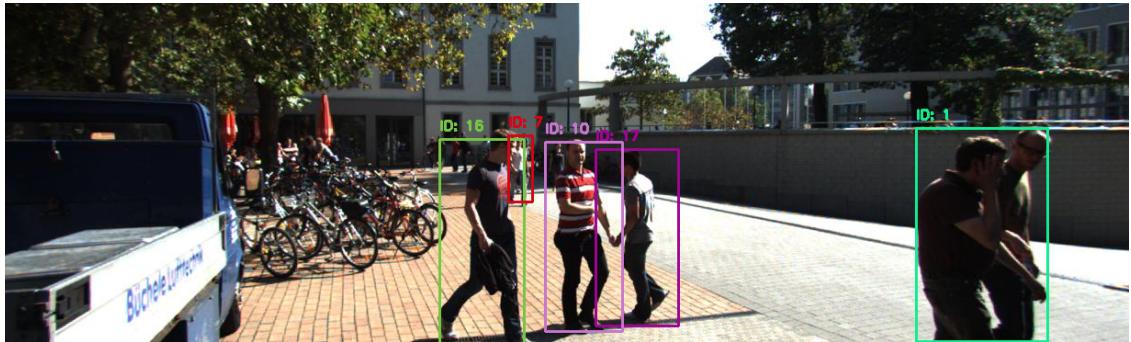
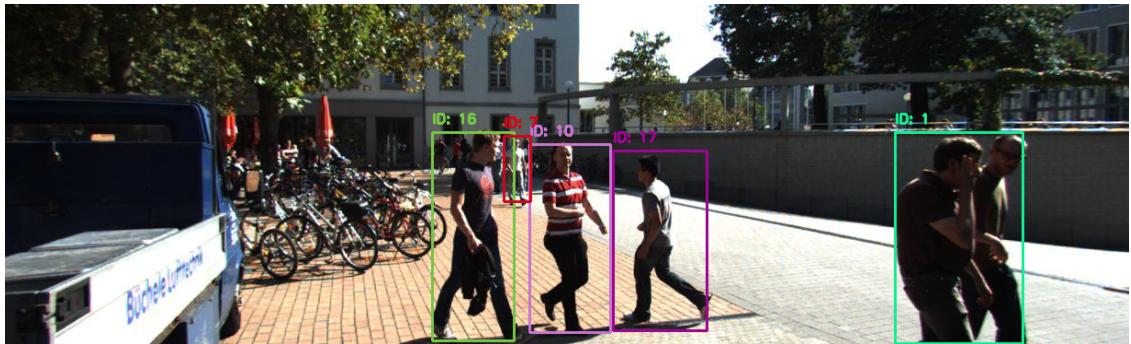
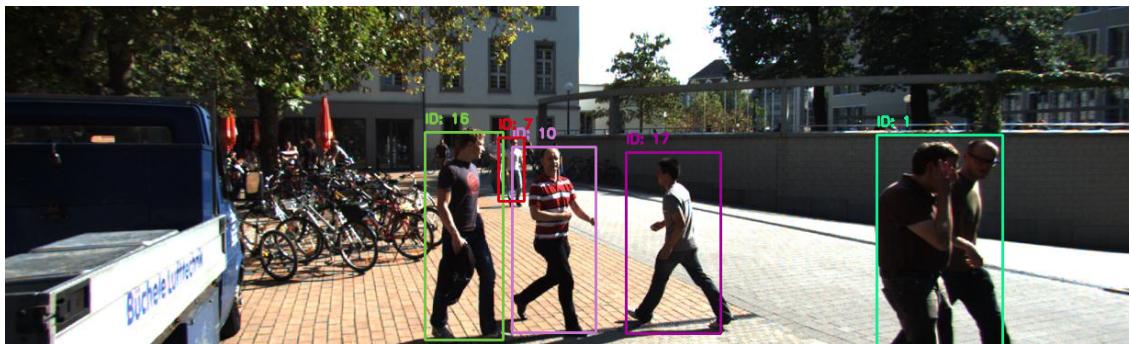


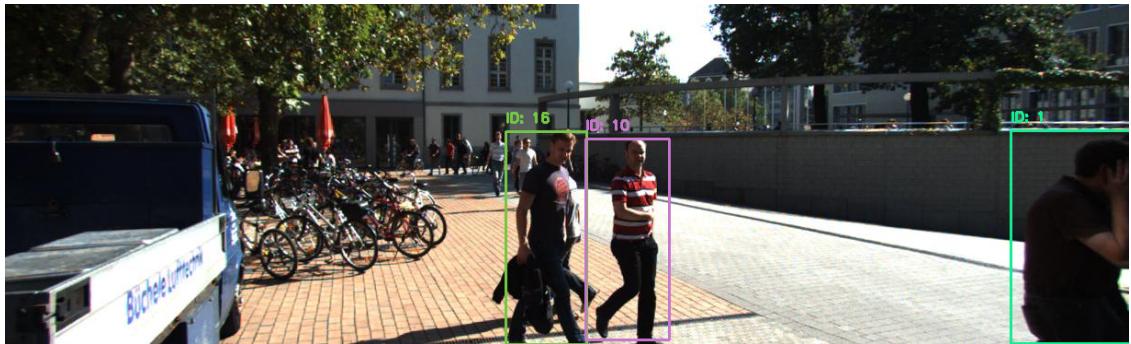
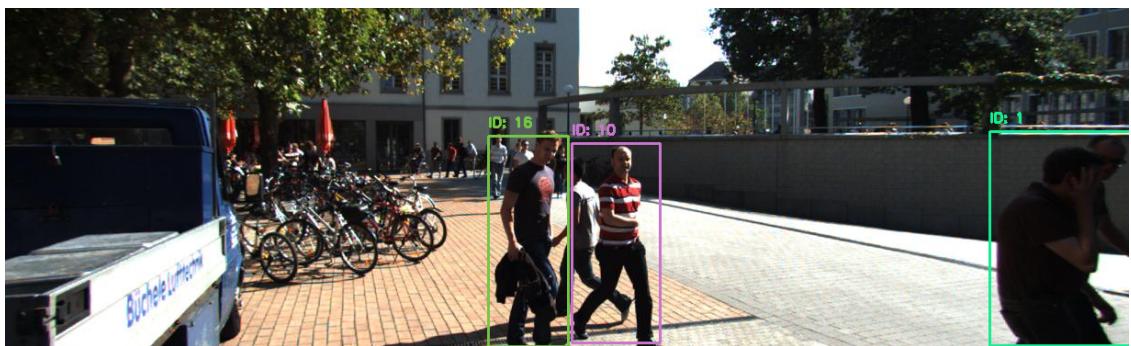
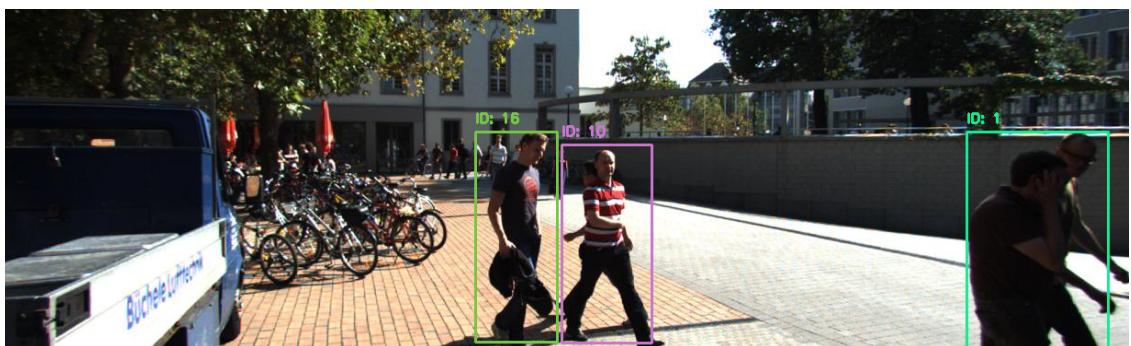
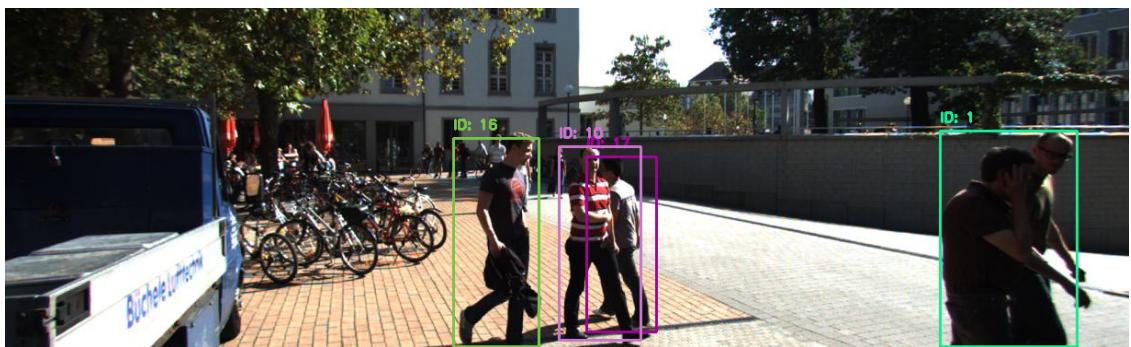


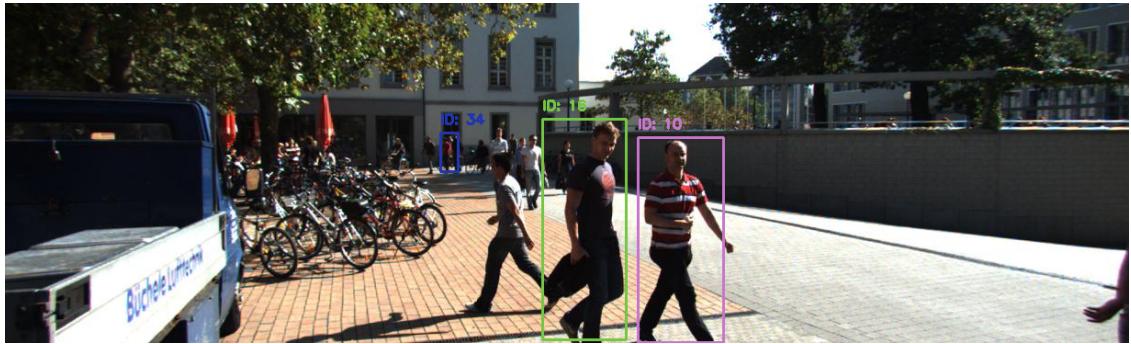
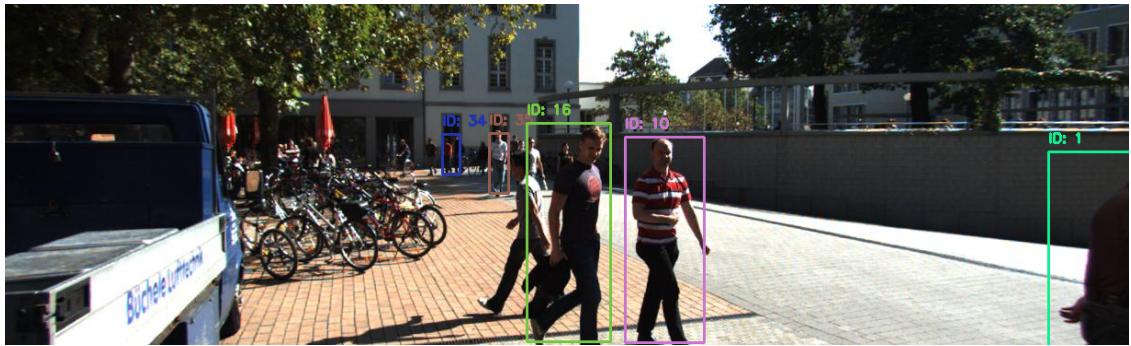
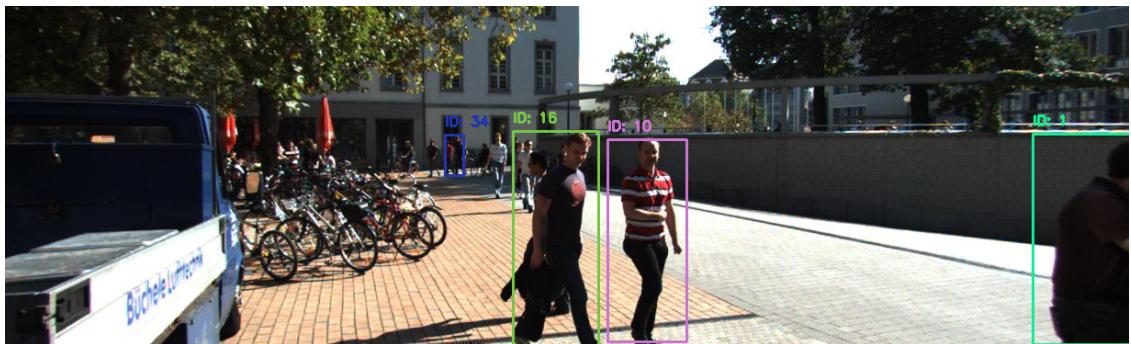


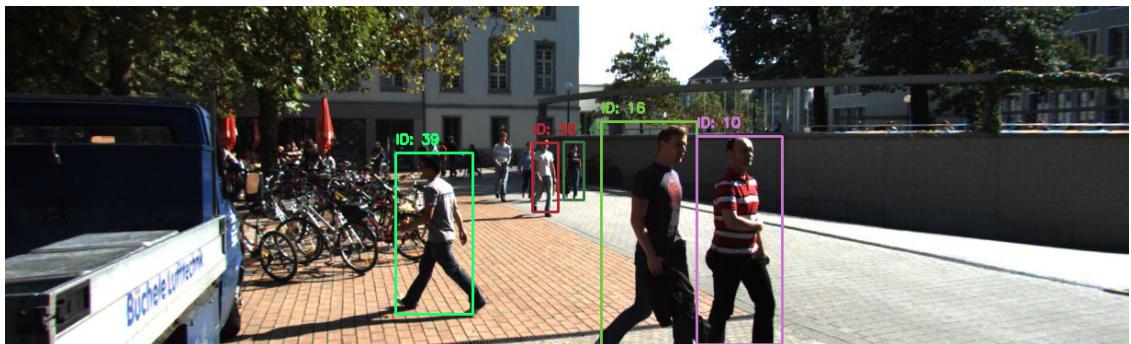
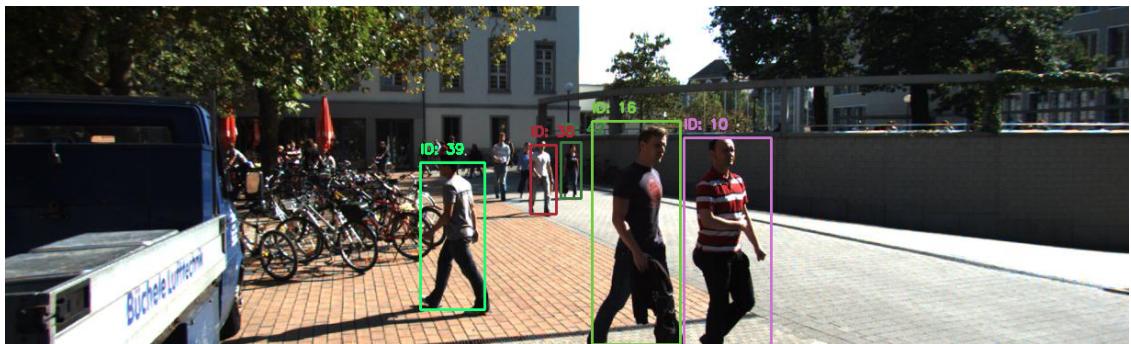
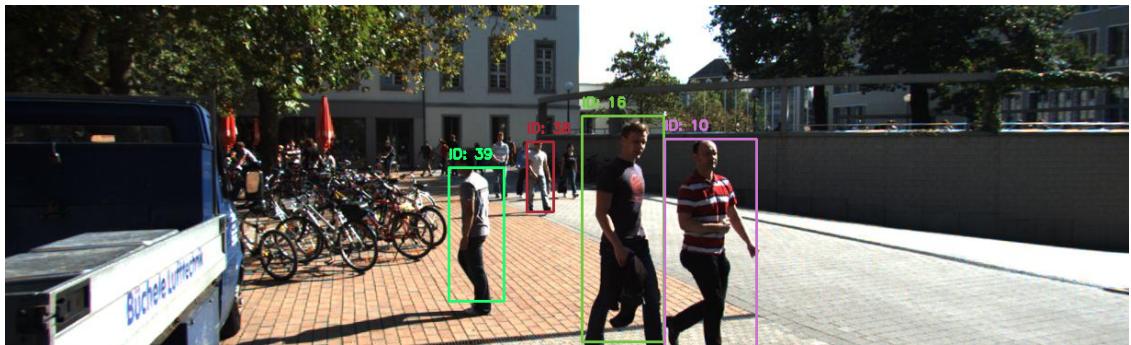


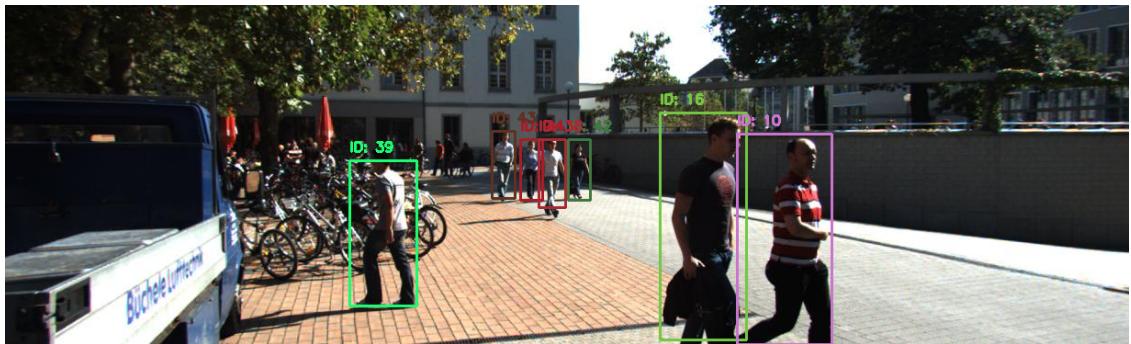


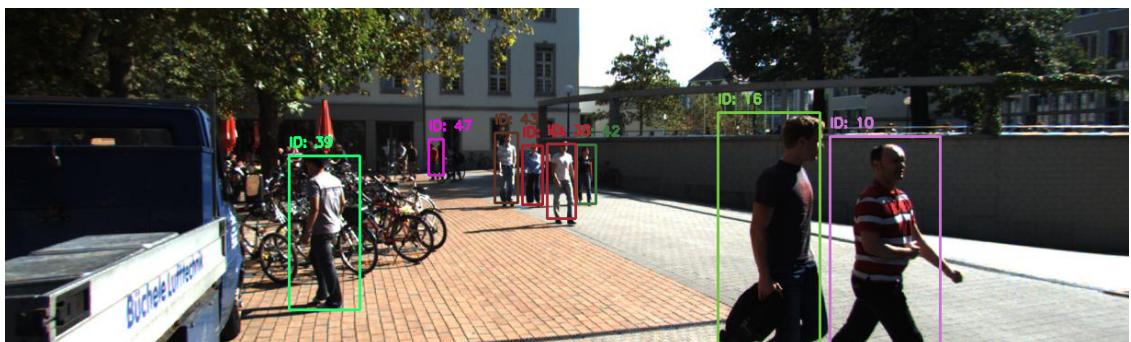
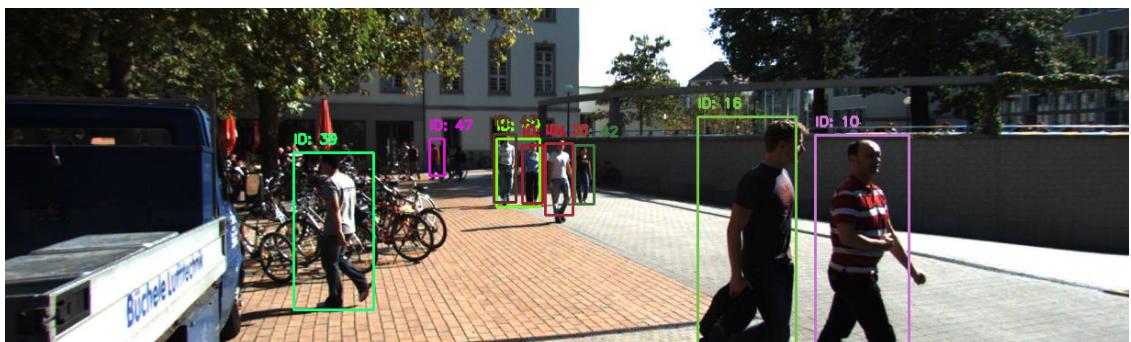
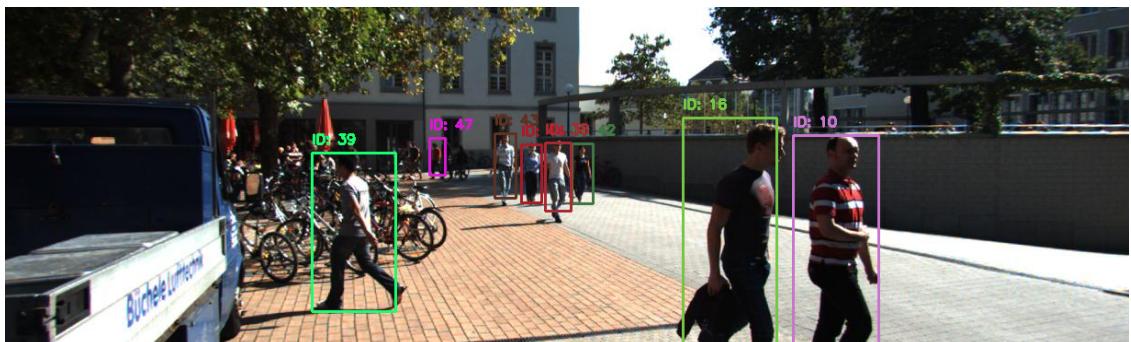
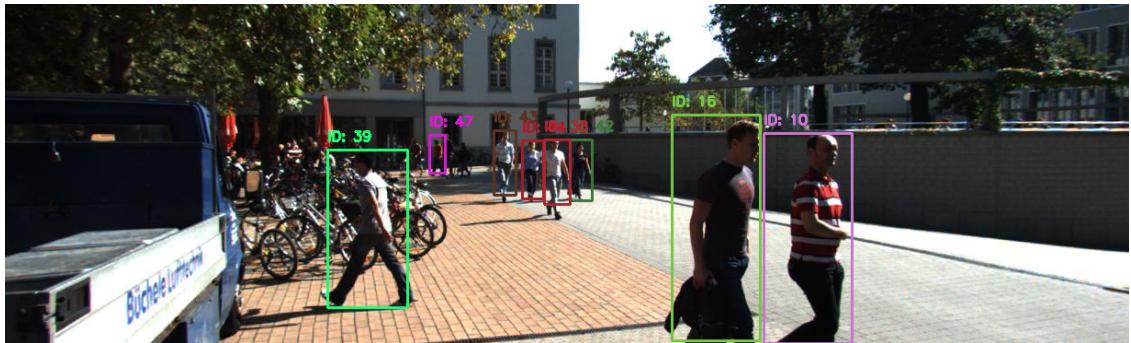


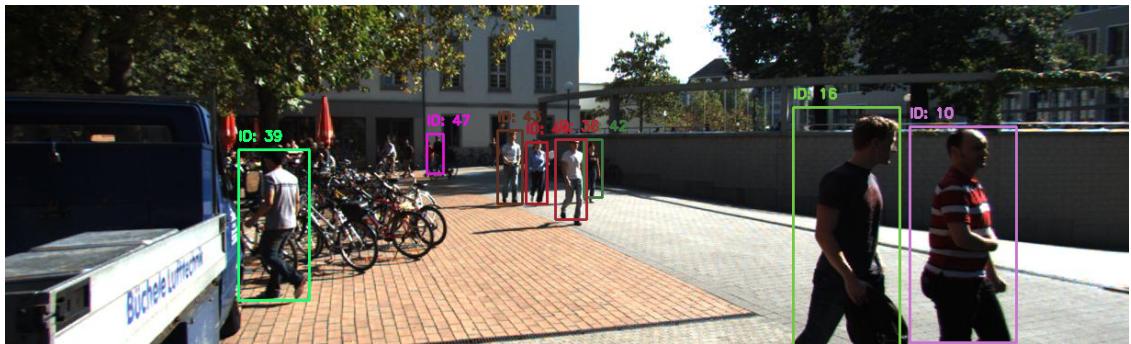
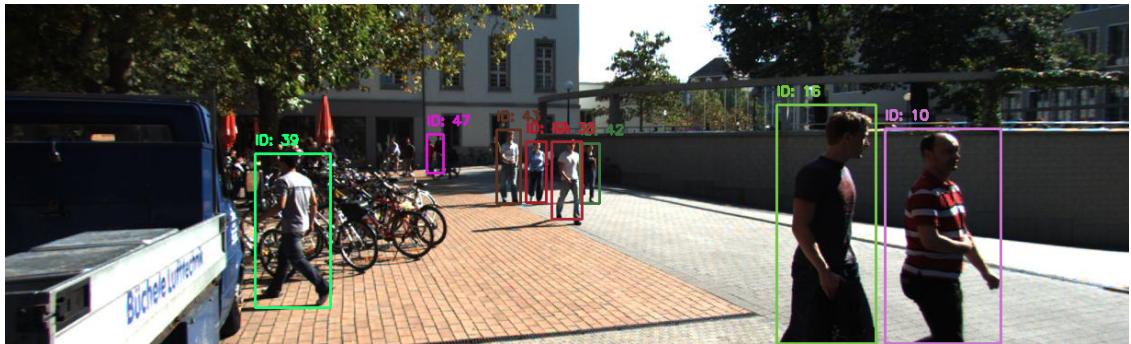
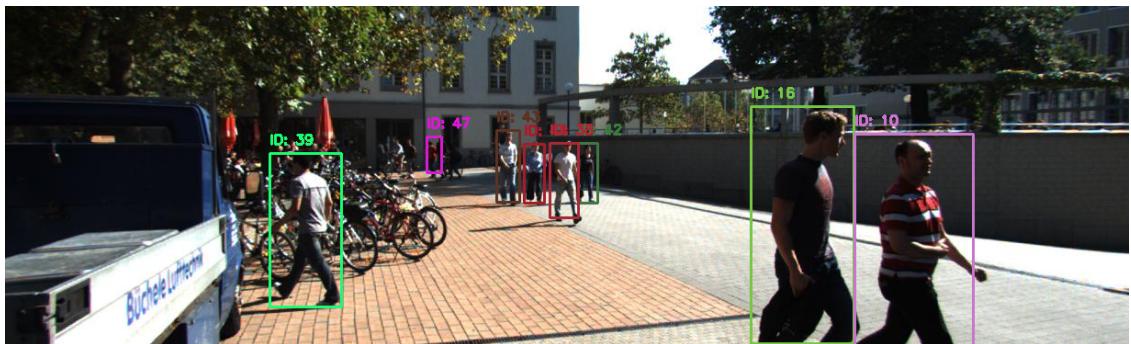


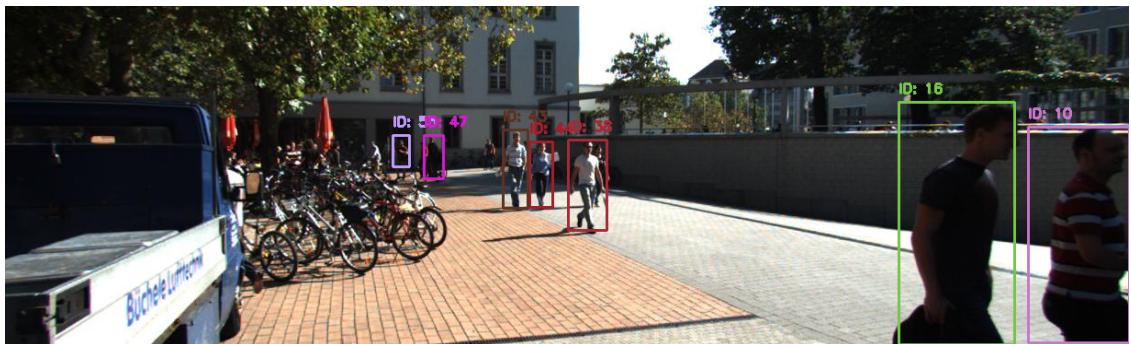
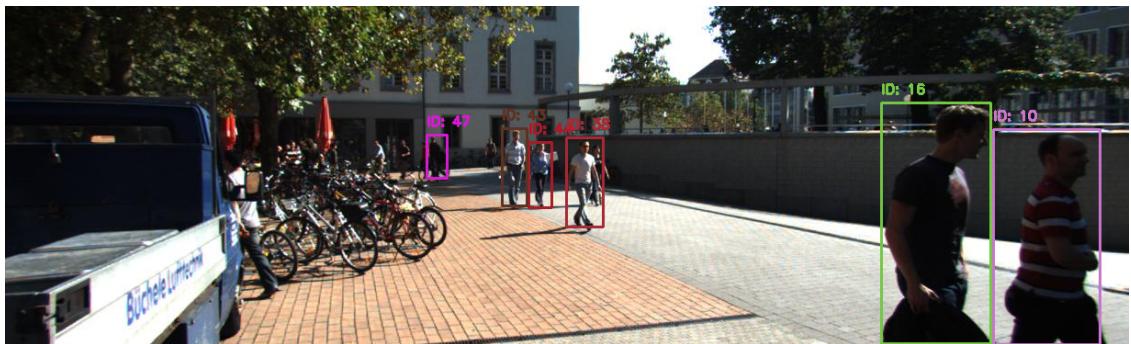
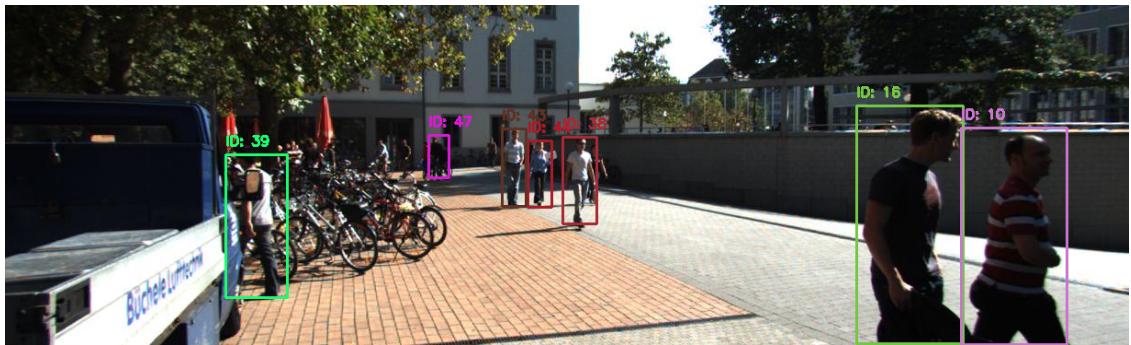
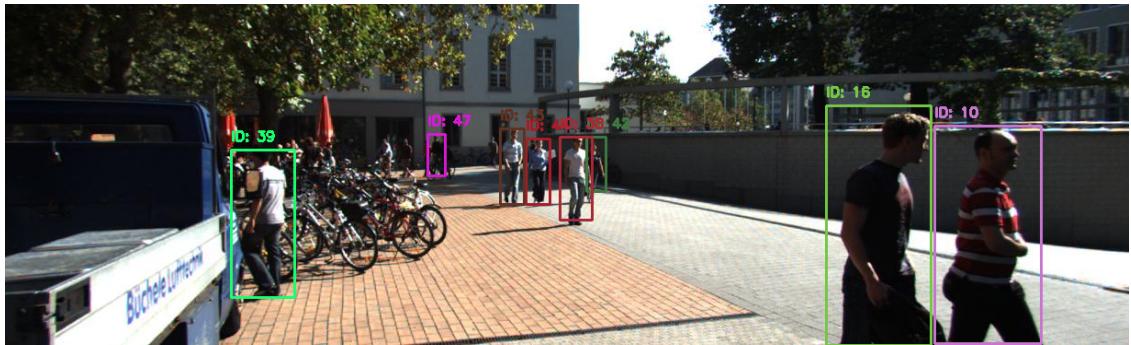


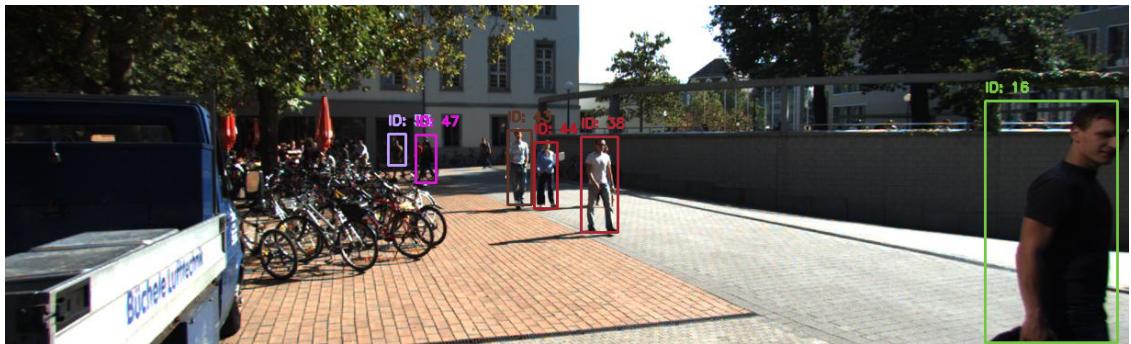
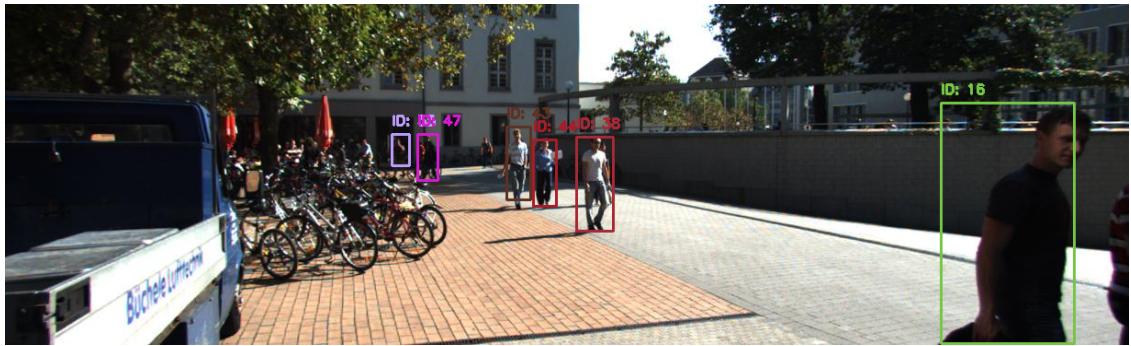
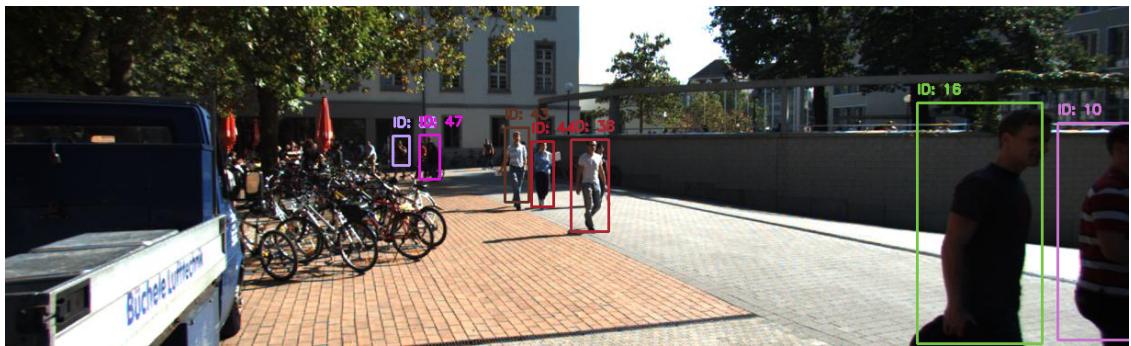


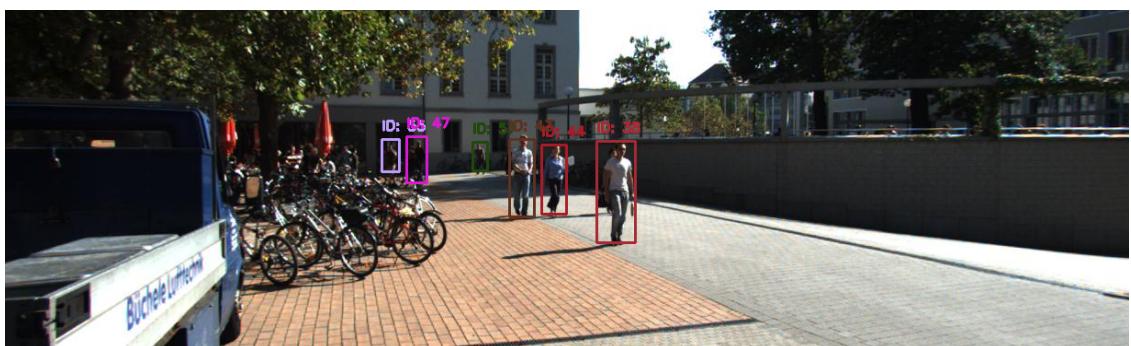
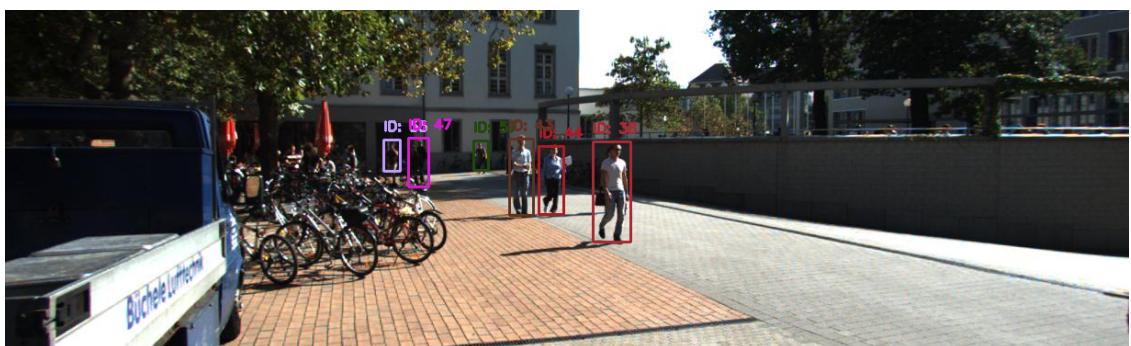
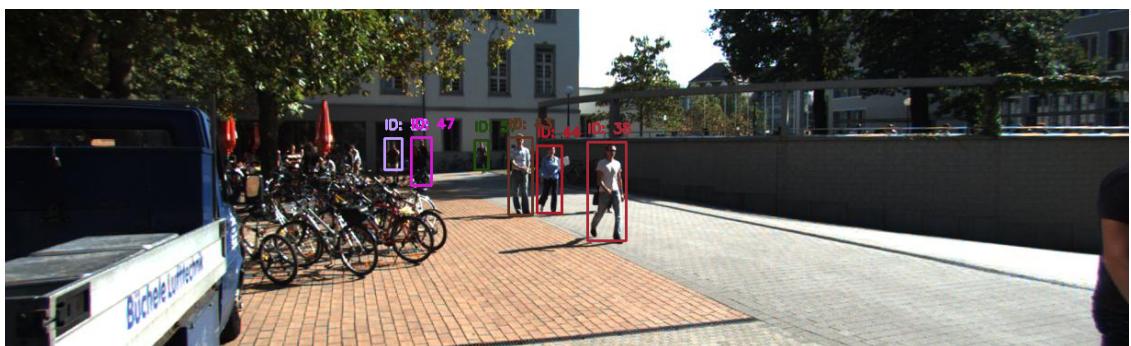
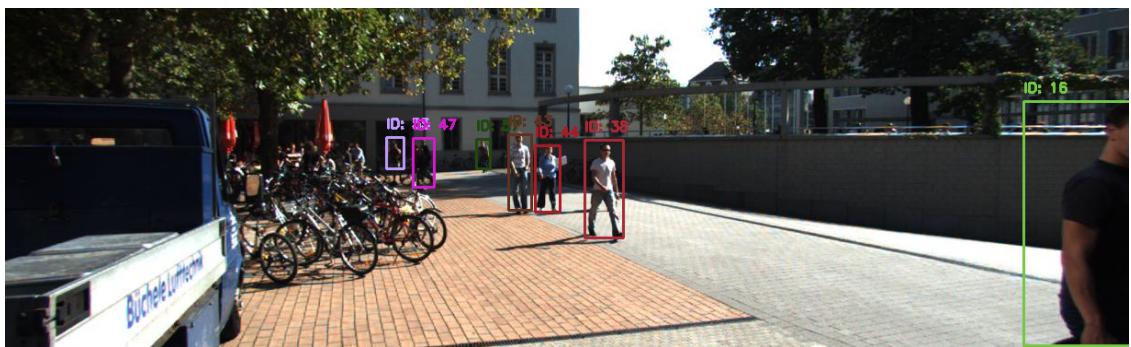


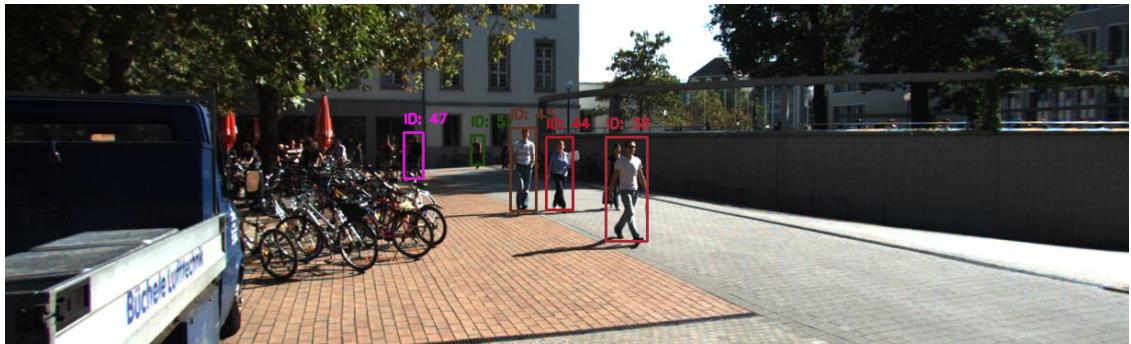
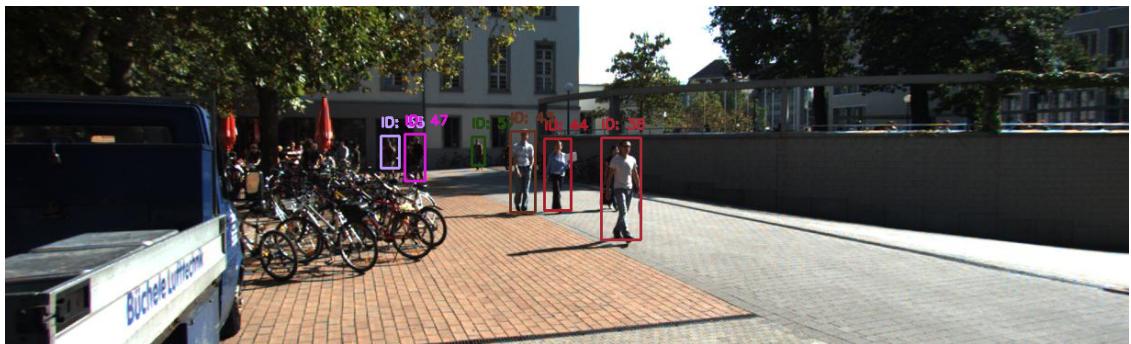




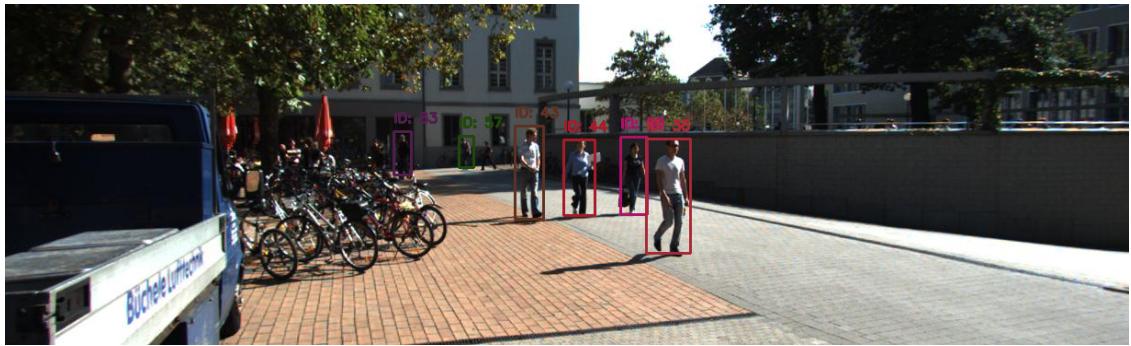
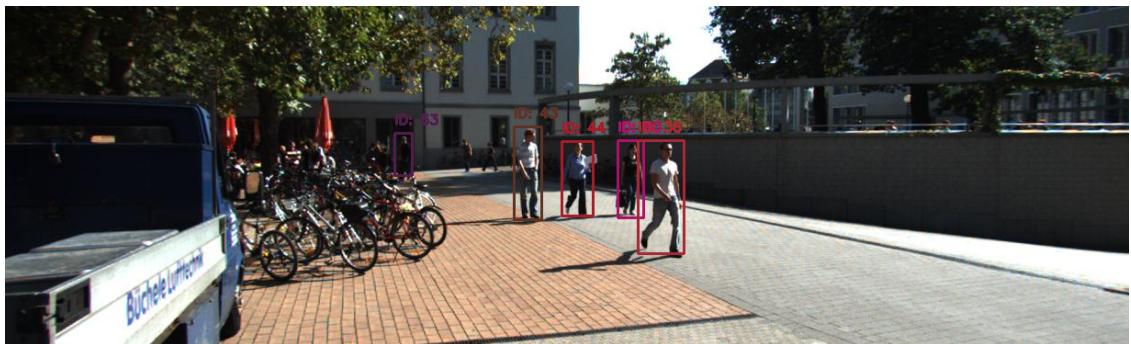


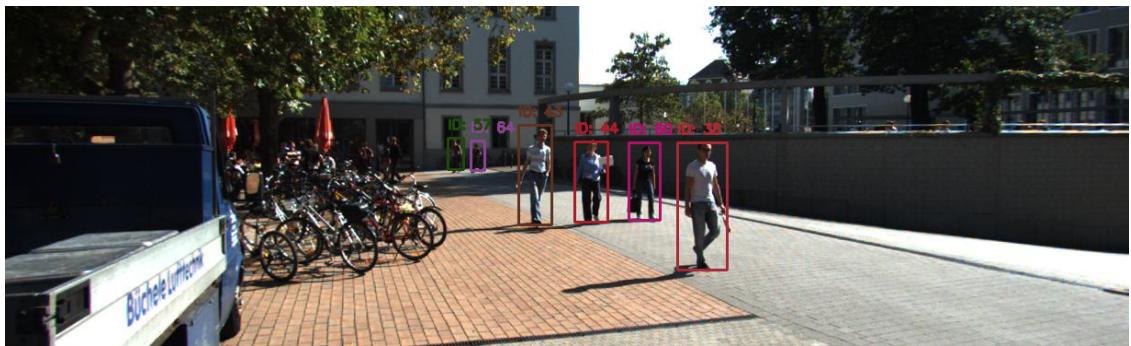
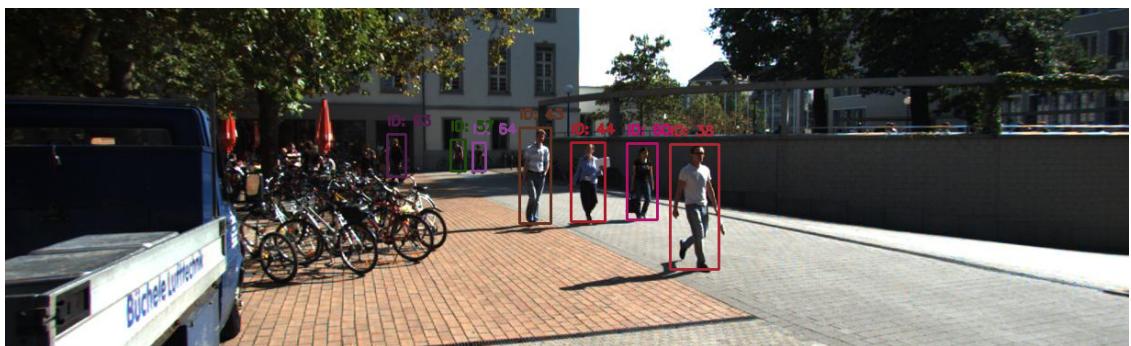
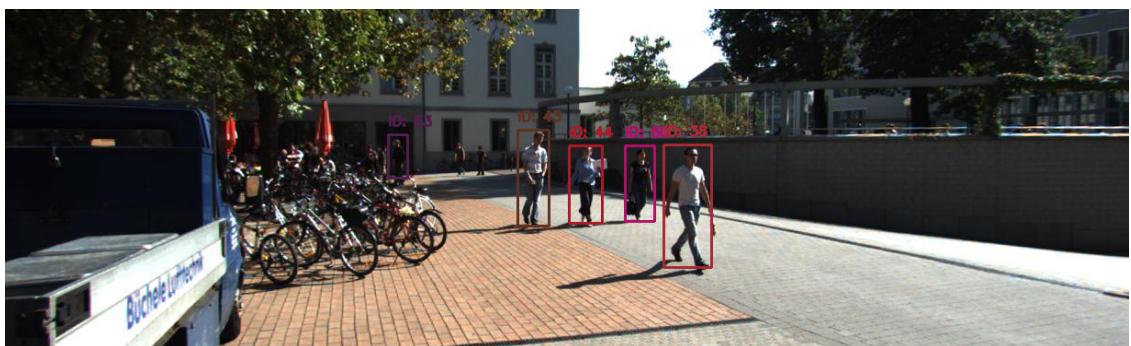
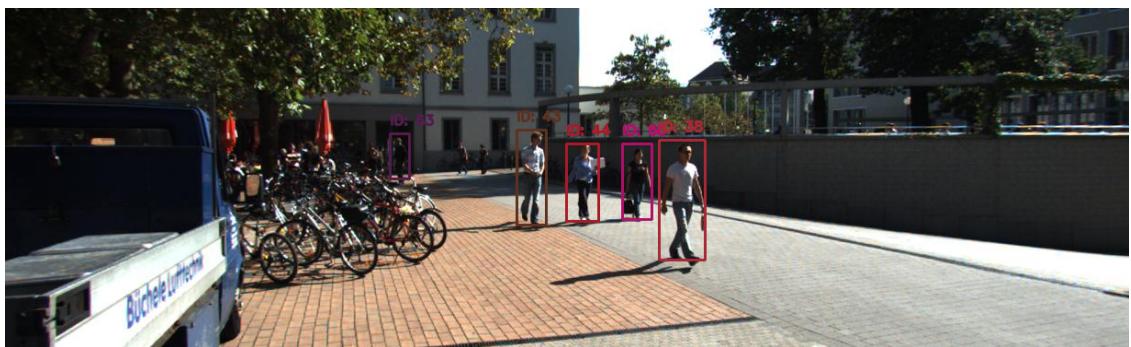


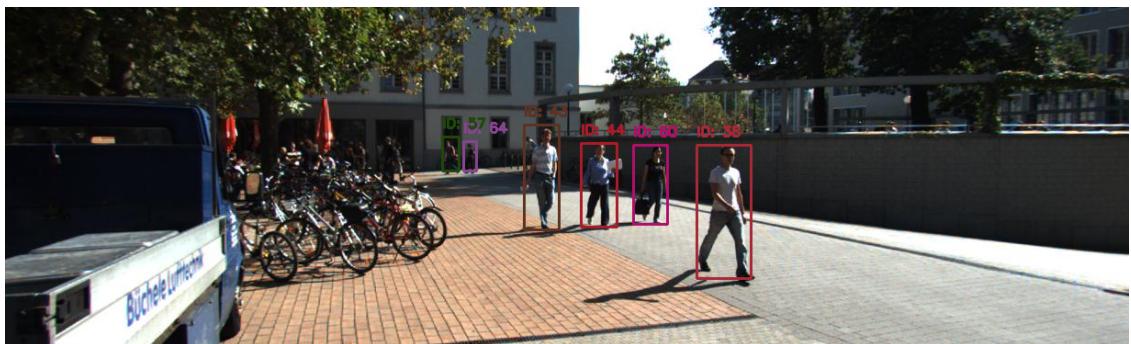
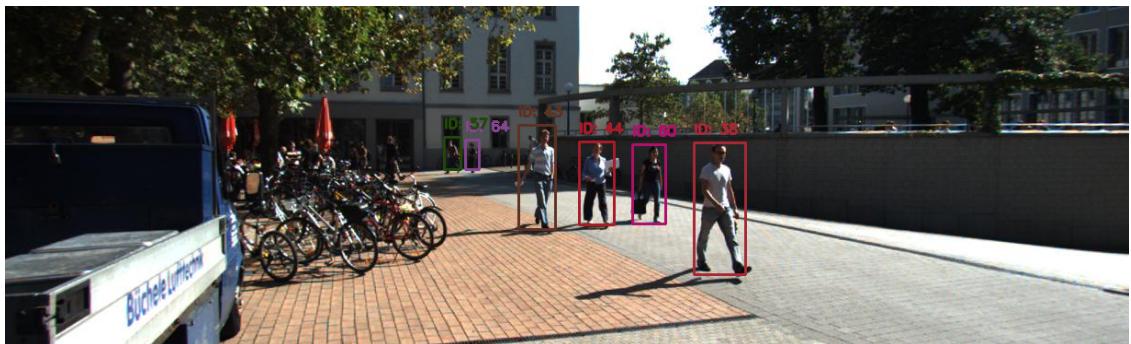
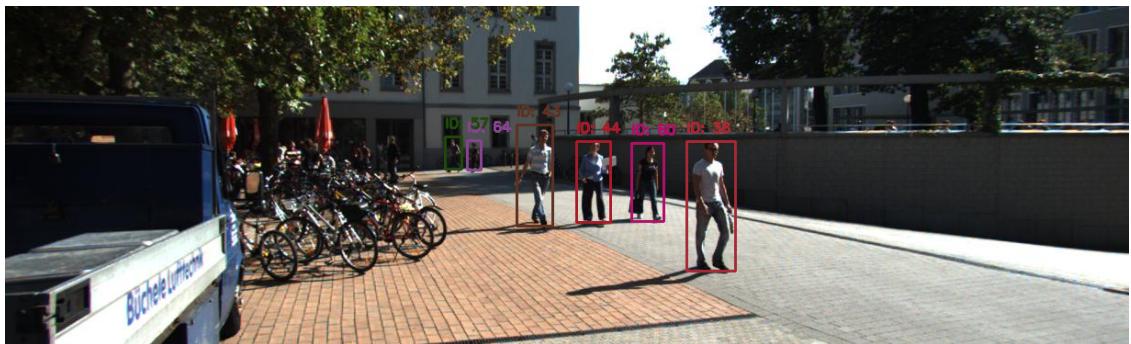


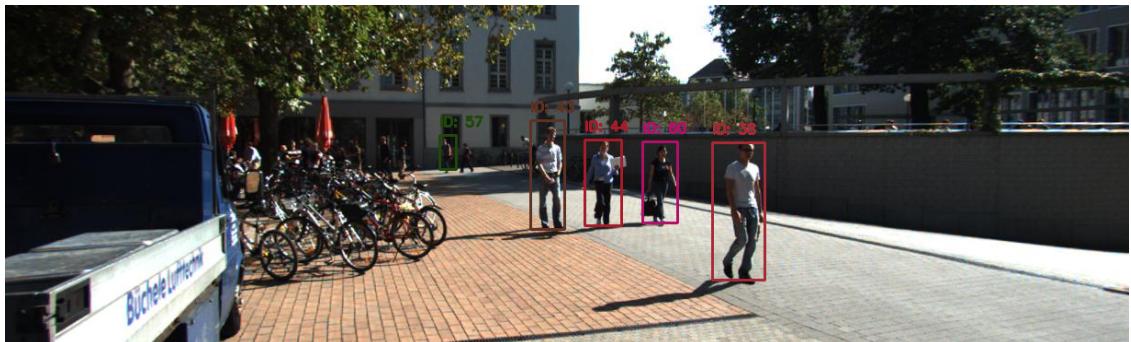
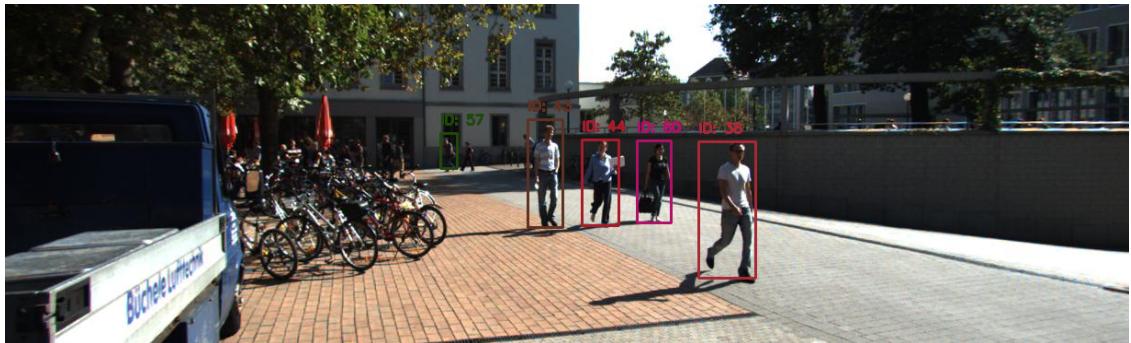


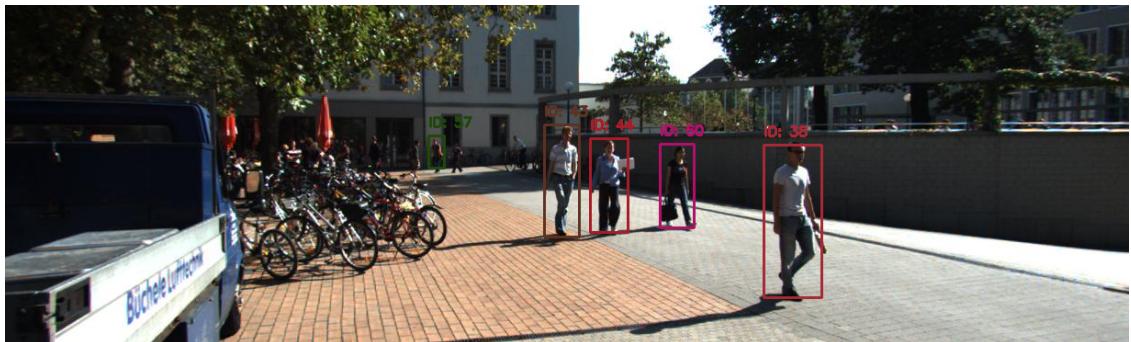
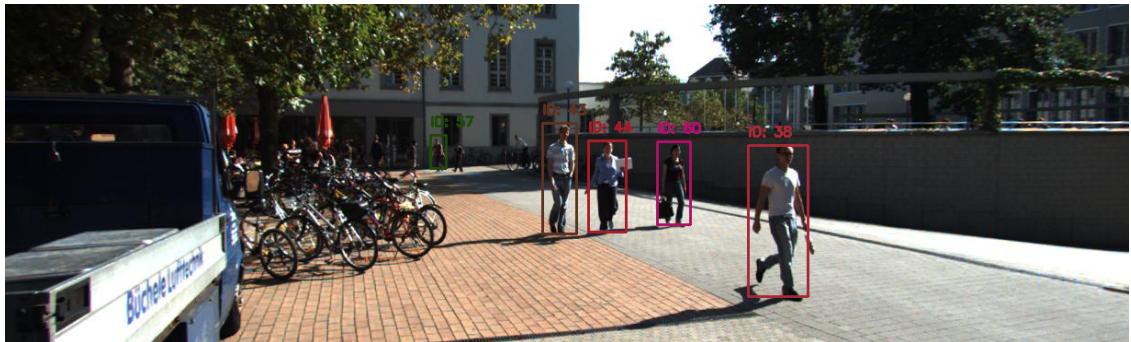
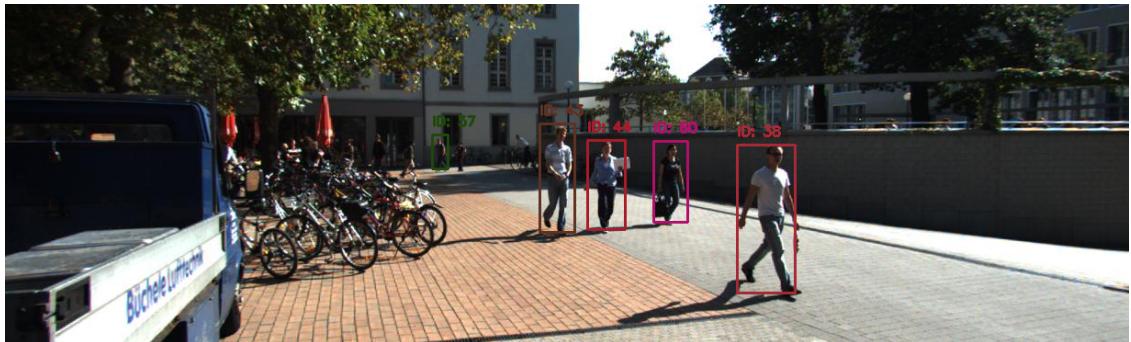


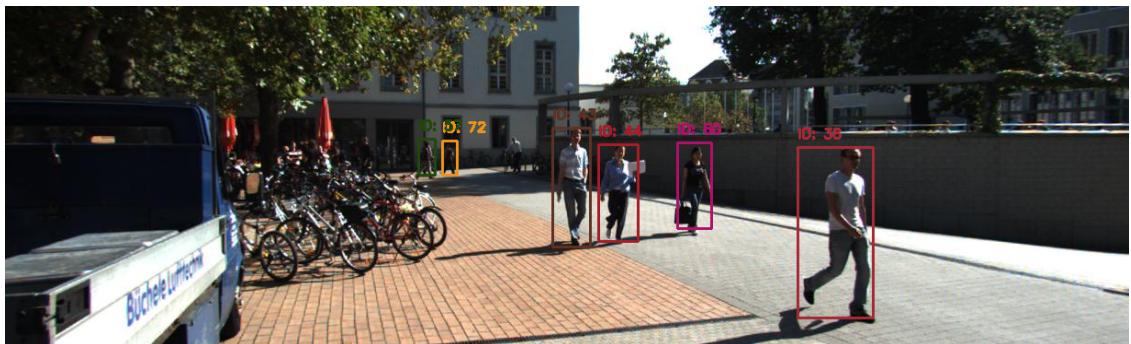
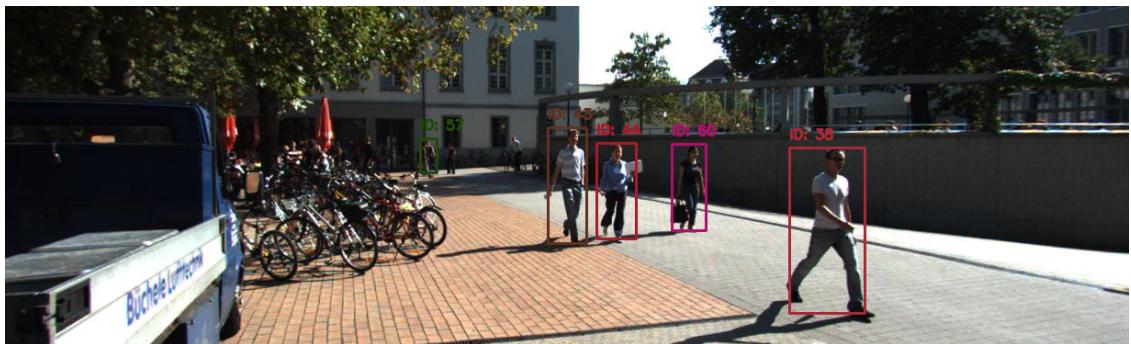
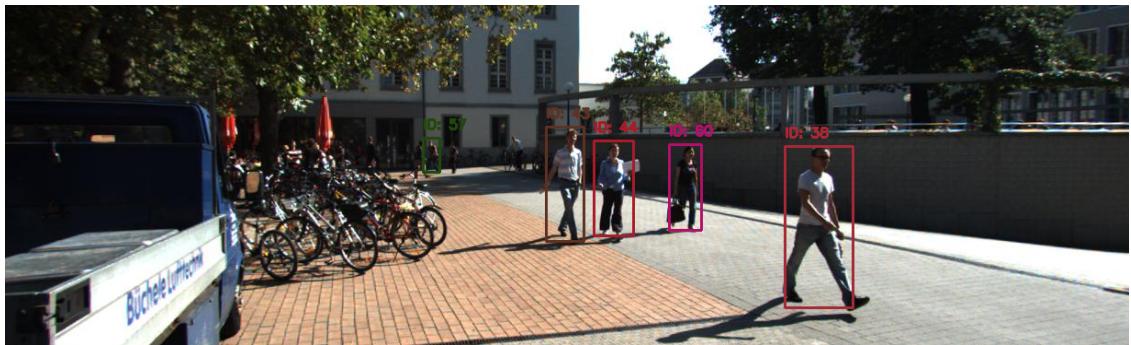
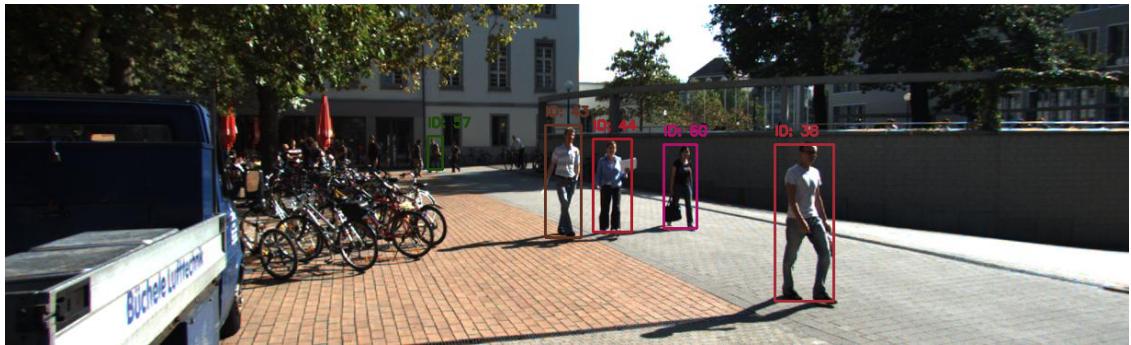


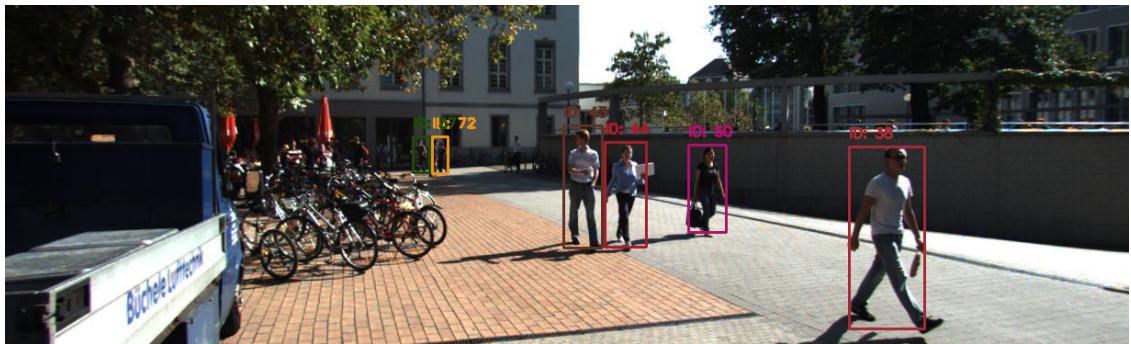
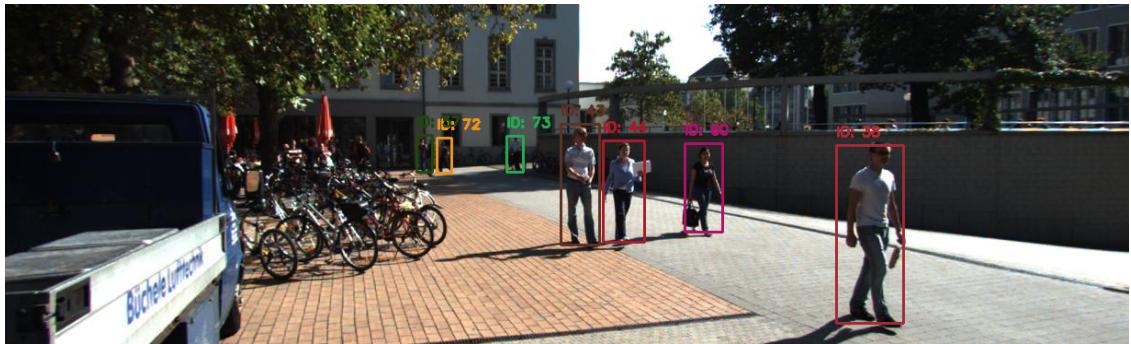
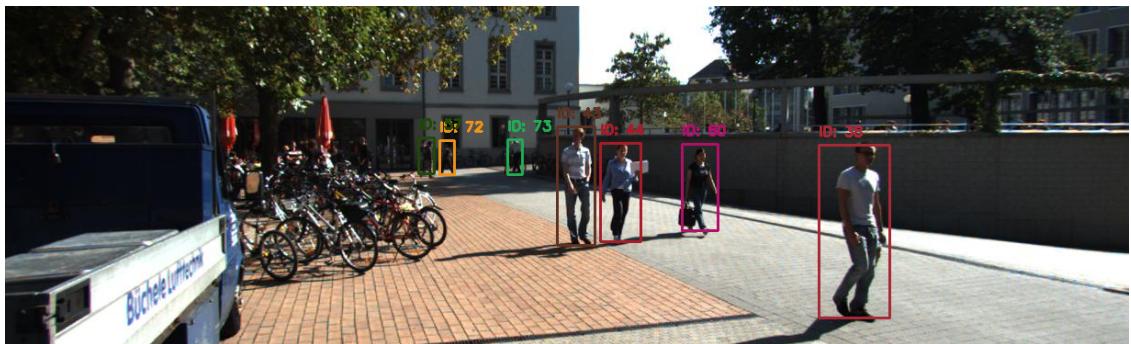


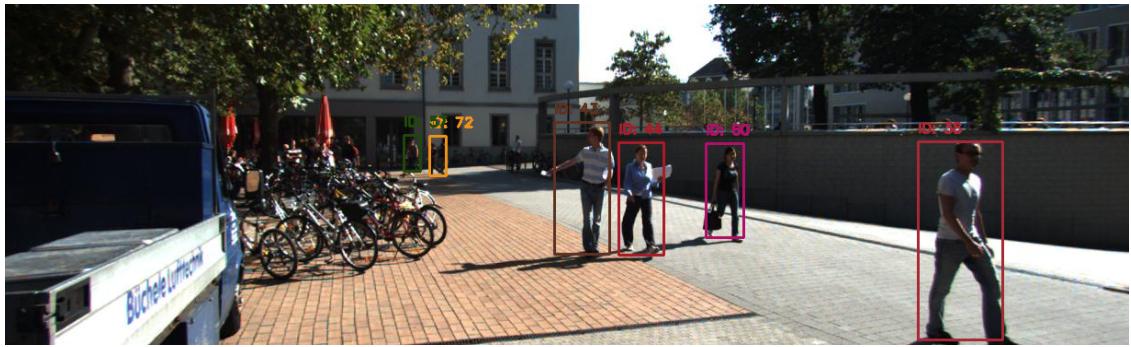
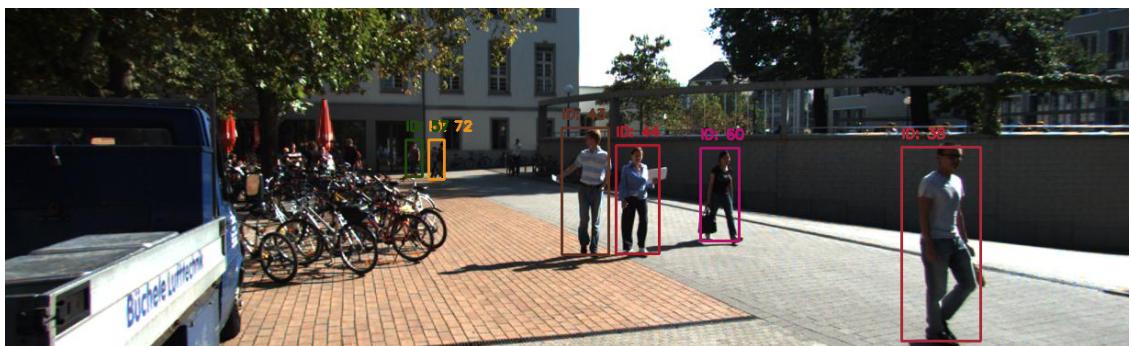
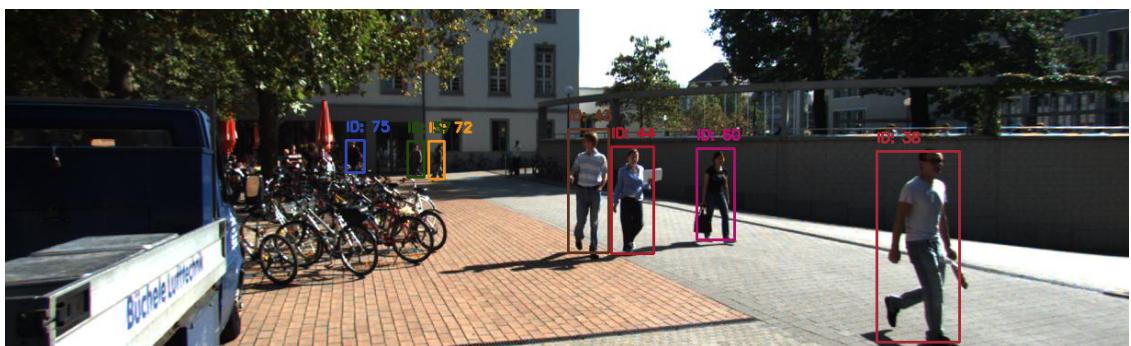
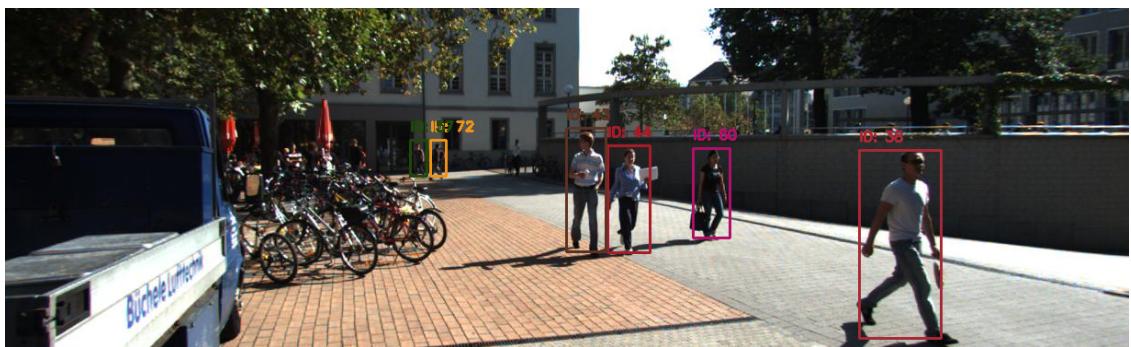


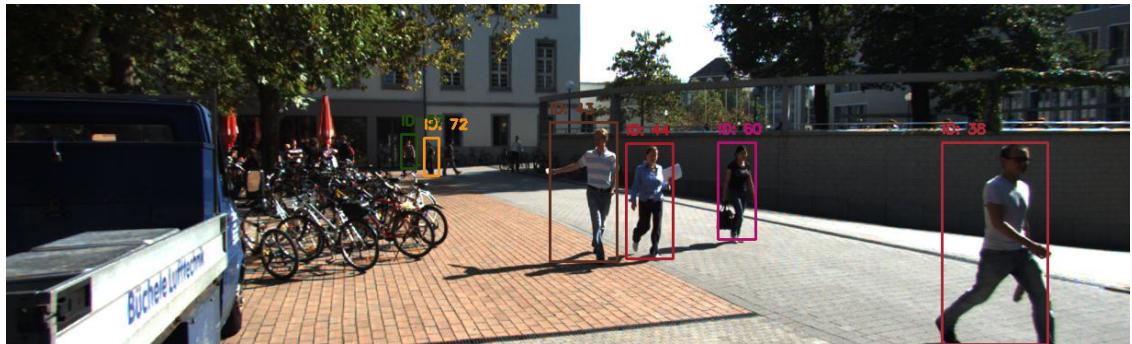
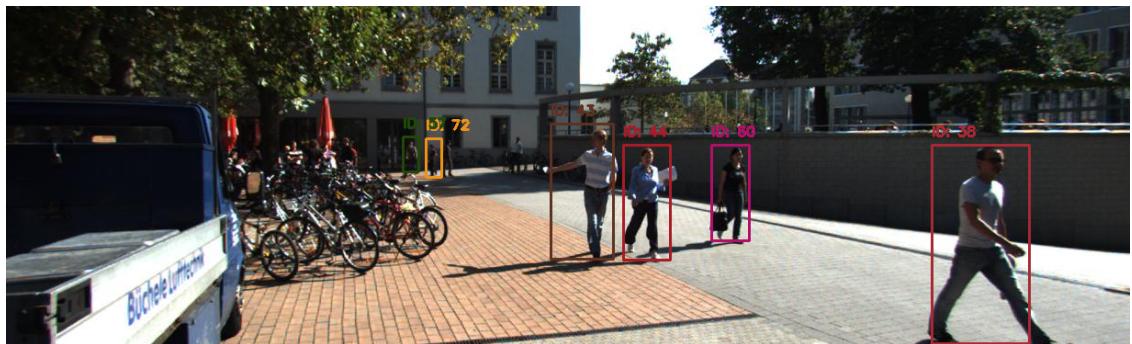


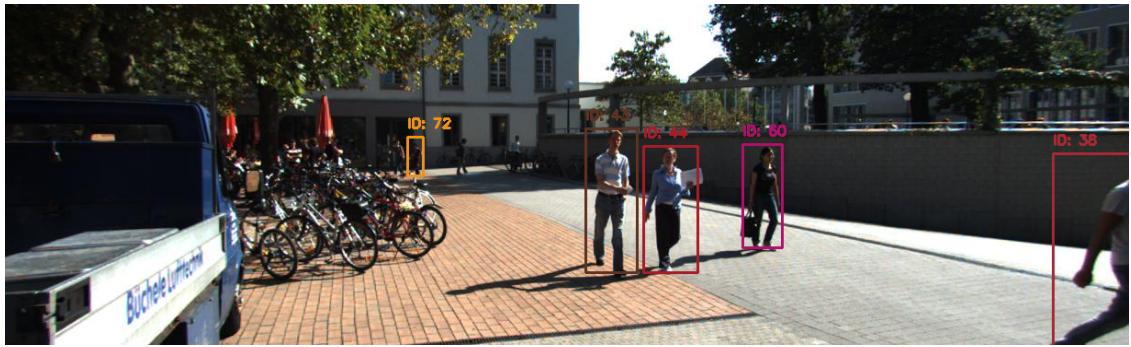
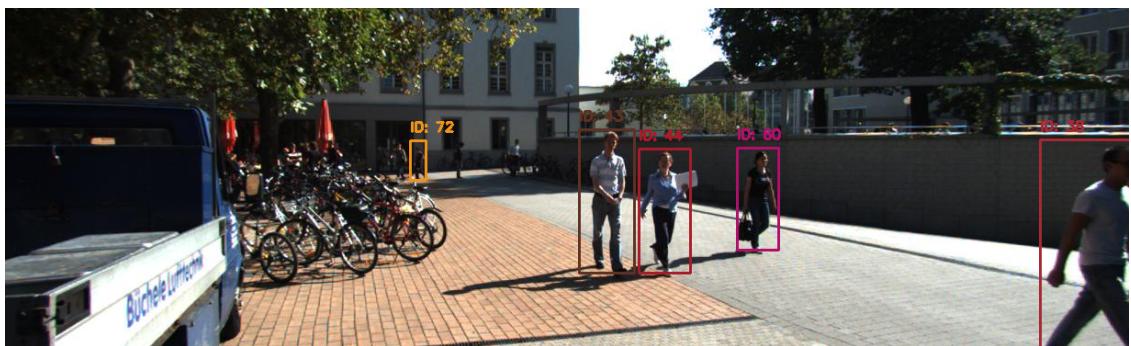
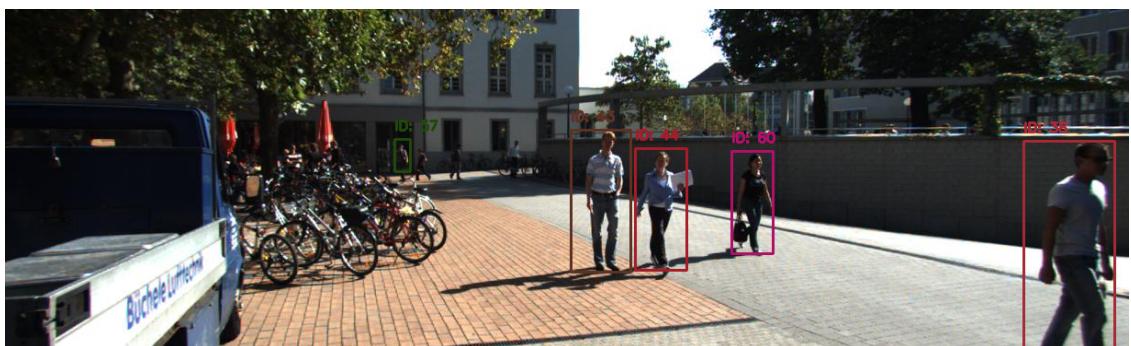
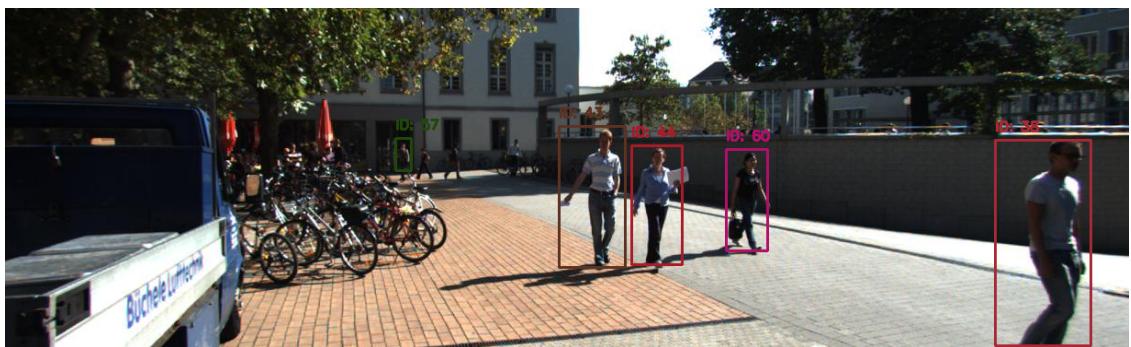


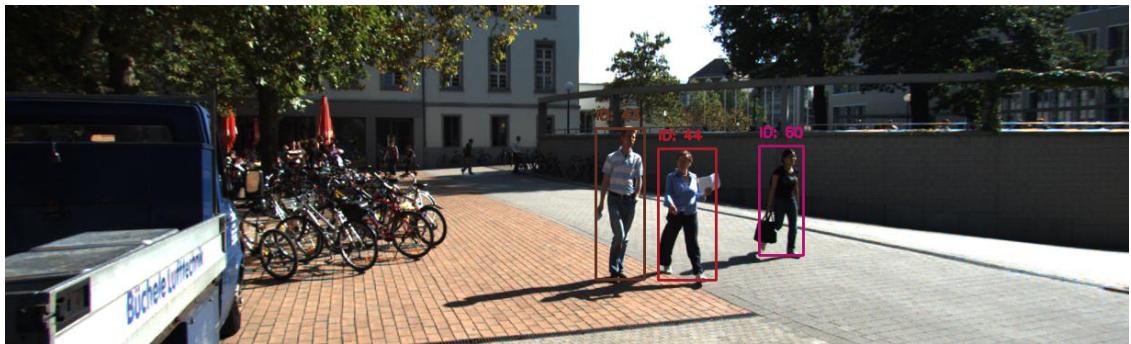
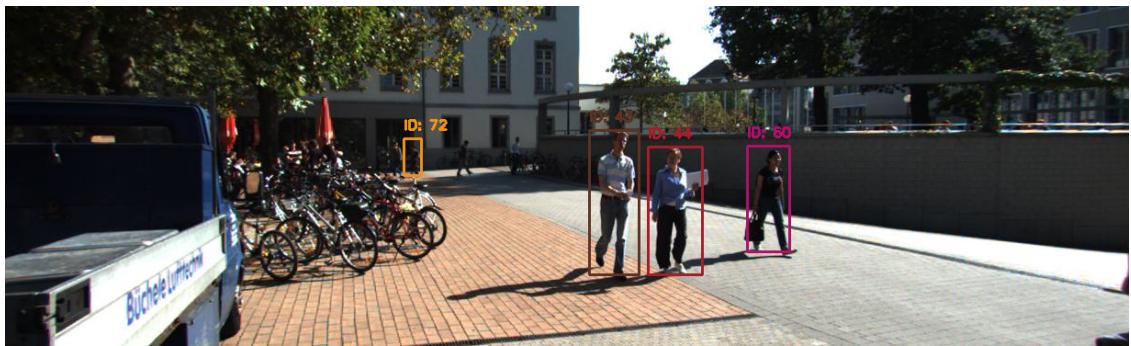


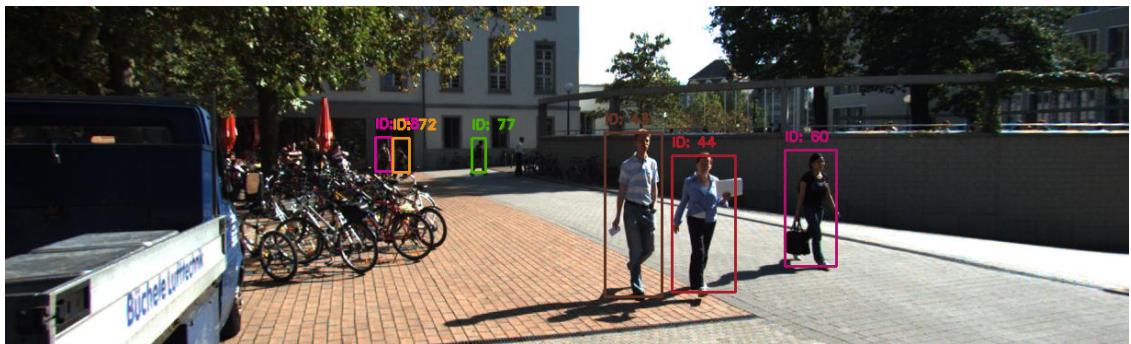
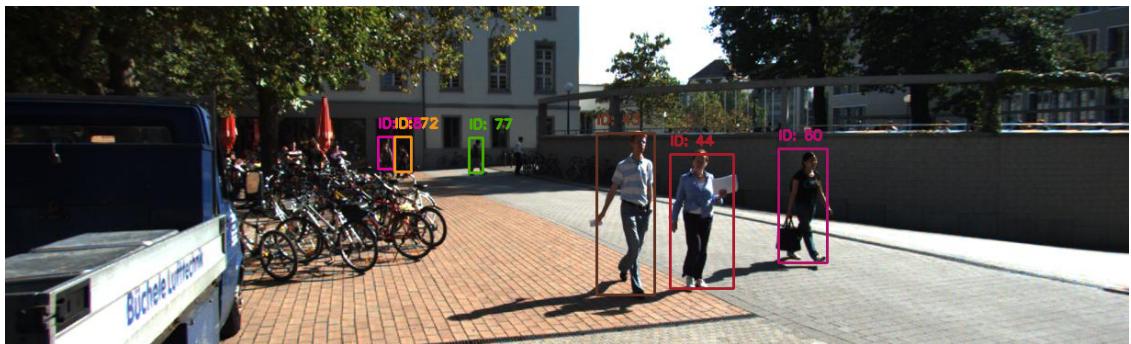
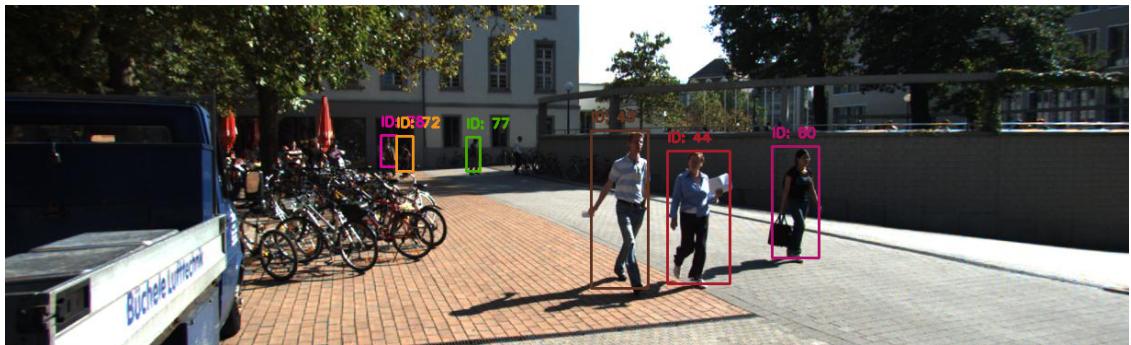
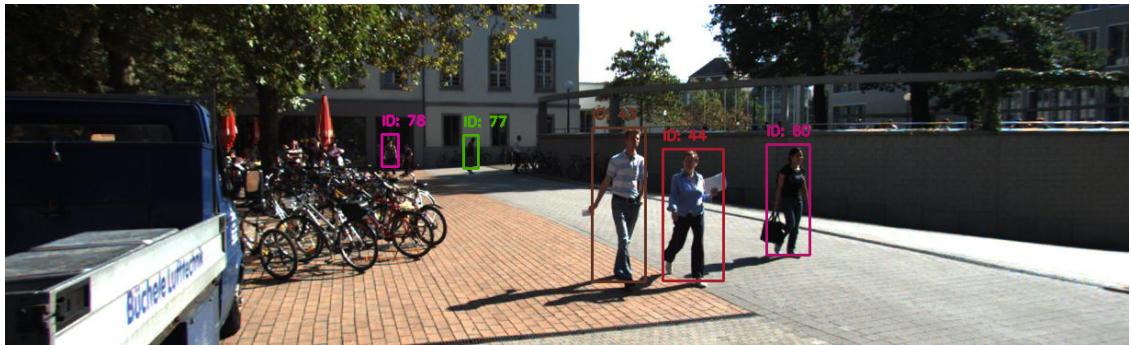


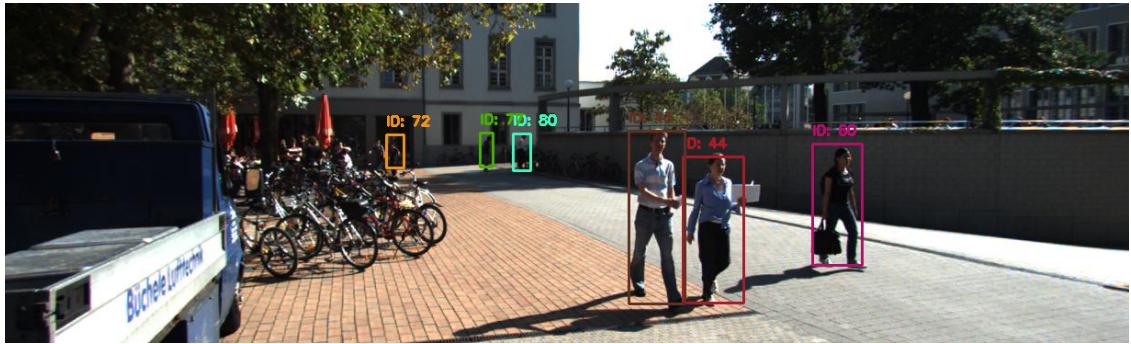
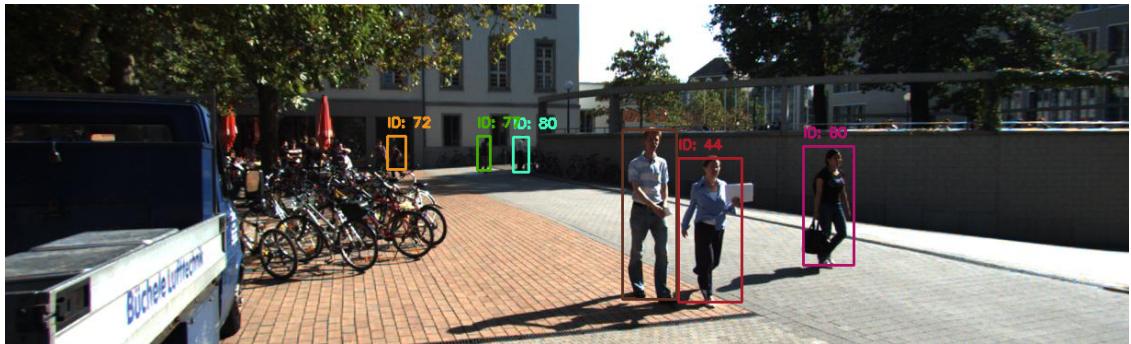
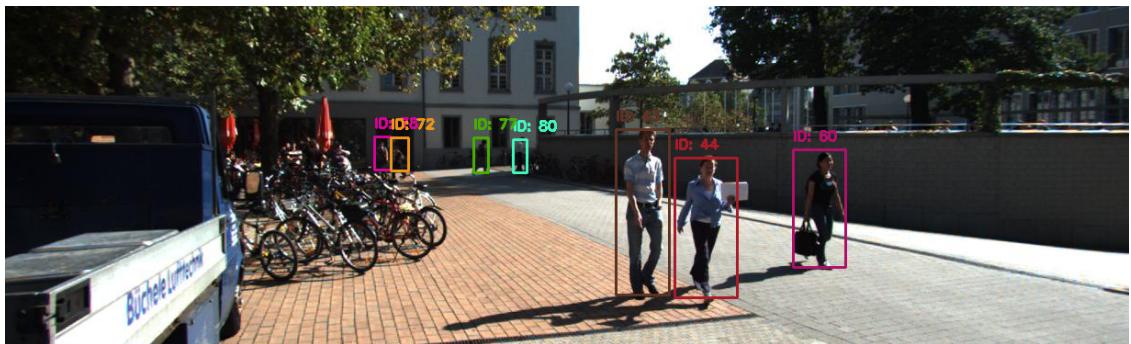


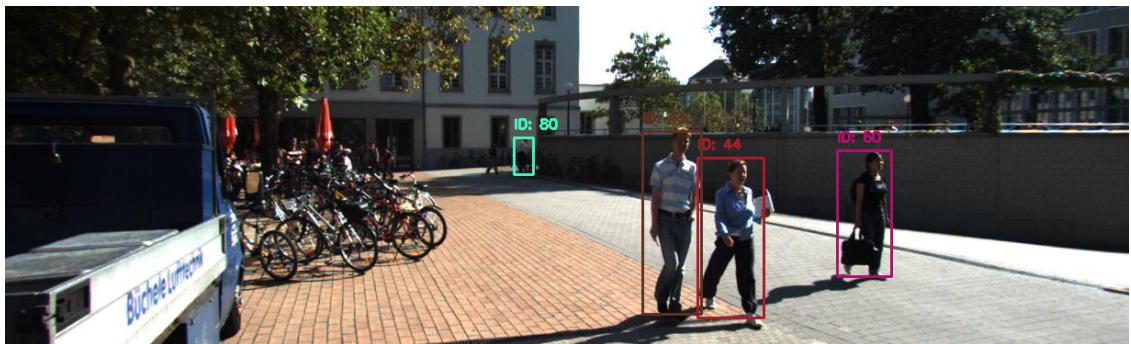
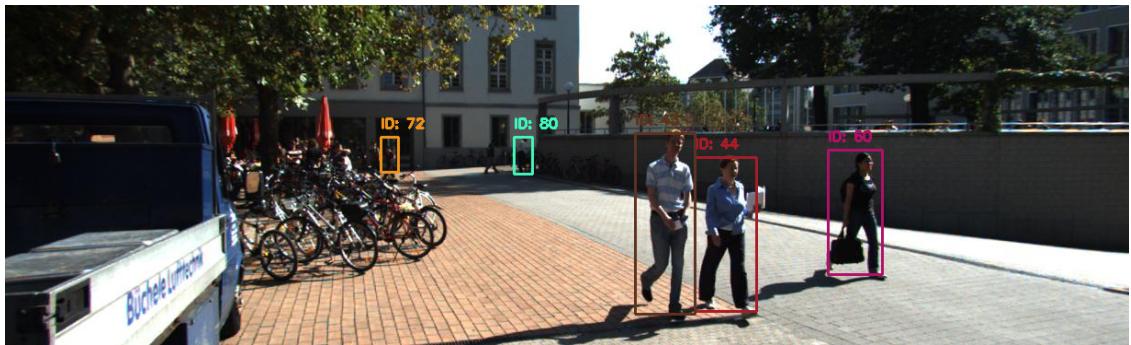
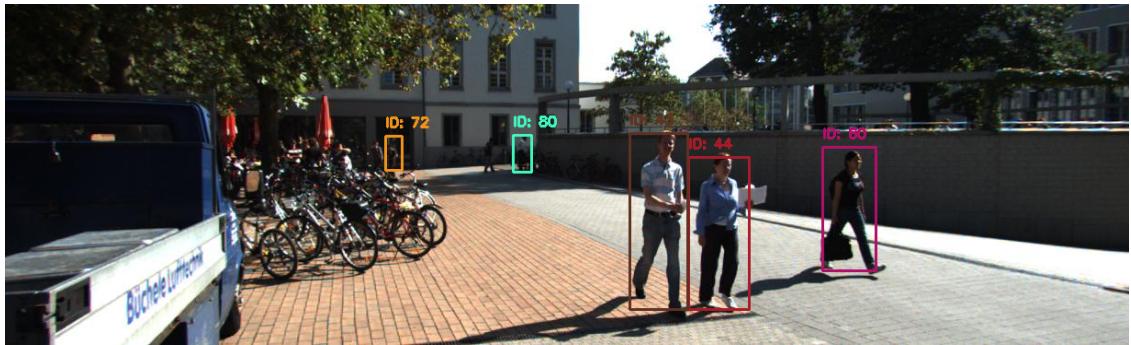
























5 7. Report Evaluation Metrics

```
[ ]: # Use the Track-Eval kit to report the complete set of performance and accuracy
    ↪metrics
# Comment on and interpret MOTA and MOTP values
```

```
[ ]: !unzip '/content/drive/MyDrive/ECE344: CV (Computer Vision)/Assignments/
    ↪Assignment-4/hw4/TrackEval-master.zip'
```

Archive: /content/drive/MyDrive/ECE344: CV (Computer Vision)/Assignments/Assignment-4/hw4/TrackEval-master.zip
12c8791b303e0a0b50f753af204249e622d0281a
creating: TrackEval-master/
inflating: TrackEval-master/.gitignore
inflating: TrackEval-master/LICENSE
inflating: TrackEval-master/Readme.md
creating: TrackEval-master/docs/
inflating: TrackEval-master/docs/BDD100k-format.txt
inflating: TrackEval-master/docs/DAVIS-format.txt
creating: TrackEval-master/docs/How_To/
inflating: TrackEval-master/docs/How_To/Add_a_new_metric.md
inflating: TrackEval-master/docs/KITTI-format.txt

```
creating: TrackEval-master/docs/MOTChallenge-Official/
inflating: TrackEval-master/docs/MOTChallenge-Official/Readme.md
inflating: TrackEval-master/docs/MOTChallenge-format.txt
inflating: TrackEval-master/docs/MOTS-format.txt
creating: TrackEval-master/docs/OpenWorldTracking-Official/
inflating: TrackEval-master/docs/OpenWorldTracking-Official/Readme.md
creating: TrackEval-master/docs/RobMOTS-Official/
inflating: TrackEval-master/docs/RobMOTS-Official/Readme.md
inflating: TrackEval-master/docs/TAO-format.txt
inflating: TrackEval-master/docs/YouTube-VIS-format.txt
inflating: TrackEval-master/minimum_requirements.txt
inflating: TrackEval-master/pyproject.toml
inflating: TrackEval-master/requirements.txt
creating: TrackEval-master/scripts/
inflating: TrackEval-master/scripts/comparison_plots.py
inflating: TrackEval-master/scripts/run_bdd.py
inflating: TrackEval-master/scripts/run_burst.py
inflating: TrackEval-master/scripts/run_burst_ow.py
inflating: TrackEval-master/scripts/run_davis.py
inflating: TrackEval-master/scripts/run_headtracking_challenge.py
inflating: TrackEval-master/scripts/run_kitti.py
inflating: TrackEval-master/scripts/run_kitti_mots.py
inflating: TrackEval-master/scripts/run_mot_challenge.py
inflating: TrackEval-master/scripts/run_mots_challenge.py
inflating: TrackEval-master/scripts/run_person_path_22.py
inflating: TrackEval-master/scripts/run_rob_mots.py
inflating: TrackEval-master/scripts/run_tao.py
inflating: TrackEval-master/scripts/run_tao_ow.py
inflating: TrackEval-master/scripts/run_youtube_vis.py
inflating: TrackEval-master/setup.cfg
inflating: TrackEval-master/setup.py
creating: TrackEval-master/tests/
inflating: TrackEval-master/tests/test_all_quick.py
inflating: TrackEval-master/tests/test_davis.py
inflating: TrackEval-master/tests/test_metrics.py
inflating: TrackEval-master/tests/test_mot17.py
inflating: TrackEval-master/tests/test_mots.py
creating: TrackEval-master/trackeval/
inflating: TrackEval-master/trackeval/__init__.py
inflating: TrackEval-master/trackeval/_timing.py
creating: TrackEval-master/trackeval/baselines/
inflating: TrackEval-master/trackeval/baselines/__init__.py
inflating: TrackEval-master/trackeval/baselines/baseline_utils.py
inflating: TrackEval-master/trackeval/baselines/non_overlap.py
inflating: TrackEval-master/trackeval/baselines/pascal_colormap.py
inflating: TrackEval-master/trackeval/baselines/stp.py
inflating: TrackEval-master/trackeval/baselines/threshold.py
inflating: TrackEval-master/trackeval/baselines/vizualize.py
```

```
creating: TrackEval-master/trackeval/datasets/
inflating: TrackEval-master/trackeval/datasets/__init__.py
inflating: TrackEval-master/trackeval/datasets/_base_dataset.py
inflating: TrackEval-master/trackeval/datasets/bdd100k.py
inflating: TrackEval-master/trackeval/datasets/burst.py
creating: TrackEval-master/trackeval/datasets/burst_helpers/
inflating: TrackEval-
master/trackeval/datasets/burst_helpers/BURST_SPECIFIC_ISSUES.md
extracting: TrackEval-master/trackeval/datasets/burst_helpers/__init__.py
inflating: TrackEval-master/trackeval/datasets/burst_helpers/burst_base.py
inflating: TrackEval-master/trackeval/datasets/burst_helpers/burst_ow_base.py
inflating: TrackEval-
master/trackeval/datasets/burst_helpers/convert_burst_format_to_tao_format.py
inflating: TrackEval-
master/trackeval/datasets/burst_helpers/format_converter.py
inflating: TrackEval-
master/trackeval/datasets/burst_helpers/tao_categories.json
inflating: TrackEval-master/trackeval/datasets/burst_ow.py
inflating: TrackEval-master/trackeval/datasets/davis.py
inflating: TrackEval-master/trackeval/datasets/head_tracking_challenge.py
inflating: TrackEval-master/trackeval/datasets/kitti_2d_box.py
inflating: TrackEval-master/trackeval/datasets/kitti_mots.py
inflating: TrackEval-master/trackeval/datasets/mot_challenge_2d_box.py
inflating: TrackEval-master/trackeval/datasets/mots_challenge.py
inflating: TrackEval-master/trackeval/datasets/person_path_22.py
inflating: TrackEval-master/trackeval/datasets/rob_mots.py
inflating: TrackEval-master/trackeval/datasets/rob_mots_classmap.py
inflating: TrackEval-master/trackeval/datasets/run_rob_mots.py
inflating: TrackEval-master/trackeval/datasets/tao.py
inflating: TrackEval-master/trackeval/datasets/tao_ow.py
inflating: TrackEval-master/trackeval/datasets/youtube_vis.py
inflating: TrackEval-master/trackeval/eval.py
creating: TrackEval-master/trackeval/metrics/
inflating: TrackEval-master/trackeval/metrics/__init__.py
inflating: TrackEval-master/trackeval/metrics/_base_metric.py
inflating: TrackEval-master/trackeval/metrics/clear.py
inflating: TrackEval-master/trackeval/metrics/count.py
inflating: TrackEval-master/trackeval/metrics/hota.py
inflating: TrackEval-master/trackeval/metrics/identity.py
inflating: TrackEval-master/trackeval/metrics/ideucl.py
inflating: TrackEval-master/trackeval/metrics/j_and_f.py
inflating: TrackEval-master/trackeval/metrics/track_map.py
inflating: TrackEval-master/trackeval/metrics/vace.py
inflating: TrackEval-master/trackeval/plotting.py
inflating: TrackEval-master/trackeval/utils.py
```