

# Deep SORT

## Introduction

This repository contains code for *Simple Online and Realtime Tracking with a Deep Association Metric* (Deep SORT). We extend the original SORT algorithm to integrate appearance information based on a deep appearance descriptor. See the arXiv preprint for more information.

## Dependencies

The code is compatible with Python 2.7 and 3. The following dependencies are needed to run the tracker:

- NumPy
- sklearn
- OpenCV

Additionally, feature generation requires TensorFlow ( $\geq 1.0$ ).

## Installation

First, clone the repository:

```
git clone https://github.com/nwojke/deep_sort.git
```

Then, download pre-generated detections and the CNN checkpoint file from here.

*NOTE:* The candidate object locations of our pre-generated detections are taken from the following paper:

F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, J. Yan. POI: Multiple Object Tracking with High Performance Detection and Appearance Feature. In BMTT, SenseTime Group Limited, 2016.

We have replaced the appearance descriptor with a custom deep convolutional neural network (see below).

## Running the tracker

The following example starts the tracker on one of the MOT16 benchmark sequences. We assume resources have been extracted to the repository root directory and the MOT16 benchmark data is in `./MOT16`:

```
python deep_sort_app.py \  
  --sequence_dir=./MOT16/test/MOT16-06 \  
  --
```

```
--detection_file=./resources/detections/MOT16_POI_test/MOT16-06.npy \  
--min_confidence=0.3 \  
--nn_budget=100 \  
--display=True
```

Check `python deep_sort_app.py -h` for an overview of available options. There are also scripts in the repository to visualize results, generate videos, and evaluate the MOT challenge benchmark.

## Generating detections

Beside the main tracking application, this repository contains a script to generate features for person re-identification, suitable to compare the visual appearance of pedestrian bounding boxes using cosine similarity. The following example generates these features from standard MOT challenge detections. Again, we assume resources have been extracted to the repository root directory and MOT16 data is in `./MOT16`:

```
python generate_detections.py  
--model=resources/networks/mars-small128.ckpt \  
--mot_dir=./MOT16/train \  
--output_dir=./resources/detections/MOT16_train
```

For each sequence of the MOT16 dataset, the output is stored as separate binary file in NumPy native format. Each file contains an array of shape  $N \times 138$ , where  $N$  is the number of detections in the corresponding MOT sequence. The first 10 columns of this array contain the raw MOT detection copied over from the input file. The remaining 128 columns store the appearance descriptor. The files generated by this command can be used as input for the `deep_sort_app.py`.

## Highlevel overview of source files

In the top-level directory are executable scripts to execute, evaluate, and visualize the tracker. The main entry point is in `deep_sort_app.py`. This file runs the tracker on a MOTChallenge sequence.

In package `deep_sort` is the main tracking code:

- `detection.py`: Detection base class.
- `kalman_filter.py`: A Kalman filter implementation and concrete parametrization for image space filtering.
- `linear_assignment.py`: This module contains code for min cost matching and the matching cascade.
- `iou_matching.py`: This module contains the IOU matching metric.
- `nn_matching.py`: A module for a nearest neighbor matching metric.

- `track.py`: The track class contains single-target track data such as Kalman state, number of hits, misses, hit streak, associated feature vectors, etc.
- `tracker.py`: This is the multi-target tracker class.

The `deep_sort_app.py` expects detections in a custom format, stored in `.npy` files. These can be computed from MOTChallenge detections using `generate_detections.py`. We also provide pre-generated detections.

## Citing DeepSORT

If you find this repo useful in your research, please consider citing the following papers:

```
@article{Wojke2017simple,
  title={Simple Online and Realtime Tracking with a Deep Association Metric},
  author={Wojke, Nicolai and Bewley, Alex and Paulus, Dietrich},
  journal={arXiv preprint arXiv:1703.07402},
  year={2017}
}

@inproceedings{Bewley2016_sort,
  author={Bewley, Alex and Ge, Zongyuan and Ott, Lionel and Ramos, Fabio and Upcroft, Ben},
  booktitle={2016 IEEE International Conference on Image Processing (ICIP)},
  title={Simple online and realtime tracking},
  year={2016},
  pages={3464-3468},
  doi={10.1109/ICIP.2016.7533003}
}
```