

# README Deep\_sort implementation fork

Sergio-Feliciano Mendoza-Barrera

## CONTENTS

<b>I</b>	<b>README</b>	<b>1</b>
I-A	Intro . . . . .	1
I-B	Dependencies . . . . .	1
I-C	Getting started . . . . .	1
I-D	Some numbers : . . . . .	2
I-E	Disclamer : . . . . .	2
I-F	Citation . . . . .	2

The purpose of this little project is to add object tracking to yolov2 and achieve real-time multiple object tracking.

The idea is contained on this paper.

## I. README

### A. Intro

The purpose of this little project is to add object tracking to YOLOv2 and achieve real-time multiple object tracking.

The current architecture is set to only track one type of objects, but it should be easy to generalise over all objects.

Currently support people tracking (as the provided weights for deep\_sort were trained on people tracking)

### B. Dependencies

python  
numpy  
opencv 3  
tensorflow 1.0  
Cython.  
sklearn.

for using sort :

scikit-learn scikit-image FilterPy

1) Setup:

1) Clone this repository: `git clone https://github.com/bendidi/Tracking-with-darkflow.git`

2) Initialize all submodules: `git submodule update --init --recursive`

3) Go to darkflow directory and do in place build: `python3 setup.py build_ext --inplace`

### C. Getting started

Download the weights :

Read more about YOLO (in darknet) and download weight files here, in case the weight file cannot be found, thtrieu has uploaded some of his here, which include yolo-full and yolo-tiny of v1.0, tiny-yolo-v1.1 of v1.1 and yolo, tiny-yolo-voc of v2.

The artchitecture I used/tested in this project is `cfg/yolo.cfg` with the weights `bin/yolo.weights`.

Next you need to download the deep\_sort weights here (networks folder), provided by nwojke extract the folder and copy it to `deep_sort/resources`

Edit Flags in `run.py` following your configuration:

- demo : path to video file to use, set to "camera" if you wish to use your camera
- model : what model configuration to use for YOLO, you can get more information and .cfg files in here (put them in `darkflow/cfg/` folder)
- load : The corresponding weights to use with the chosen model (put them in `darkflow/bin/`) more info in here
- threshold : the confidance threshold of the YOLO detections
- gpu : How much GPU to use, 0 means use cpu
- track : to activate tracking or not
- trackObj: which objects to track as a list (notice that deep\_sort's encoder was only trained on people , so you need train your own encoder, more information in here)
- saveVideo : whether to save video or not
- BK\_MOG : add opencv's MOG background subtraction module, only useful when YOLO can't detect people in a video (low quality, . . . ) use it to detect boxes around moving objects

- tracker : which tracker to use : "deep\_sort" or "sort"

NOTE: "deep\_sort" only supports people tracking as it was only trained to track people(the code for training is not yet pub

TODO: add support for GOTURN tracker(tensorflow implementation)

TODO: add support for opencv trackers (MIL,KCF,TLD,MEDIANFLOW)

- skip: skip frames to increase fps, might decrease accuracy!
- csv : save csv file of detections in the format (frame\_id, object\_id, x, y ,~w~, h)
- display : display video while processing or not

Next you just have to run python run.py, and enjoy!

#### D. Some numbers :

speed using yolo.cfg:

YOLO with track Flag set to False : 30fps

YOLO with track Flag set to True (deep\_sort) : 14 fps

YOLO with track and background subtraction Flags set to Ture : 10.5 fps

Tests done on (1024, 1280, 3) resolution video on Nvidia GTX 1080

skipping up to 3 frames allows for more speed up while keeping accuracy of tracking‘

#### E. Disclaimer :

this project is using code forked from:

thtrieu/darkflow: for the real-time object detections and classifications.

nwojke/deep\_sort: for Simple Online Realtime Tracking with a Deep Association Metric.

Please follow the links to get an understanding of all the features of each project.

#### F. Citation

1) YOLOv2 ::

```
@article{redmon2016yolo9000,
  title={YOLO9000: Better, Faster, Stronger},
  author={Redmon, Joseph and Farhadi, Ali},
  journal={arXiv preprint arXiv:1612.08242},
  year={2016}
}
```

2) deep\_sort::

```
@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}
}
```

3) sort ::

```
@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},
  keywords={Benchmark testing;Complexity theory;Detectors;Kalman filters;Target tracking;Visualization;Computer Vision;},
  doi={10.1109/ICIP.2016.7533003}
}
```

EOF