tugot17 / YOLO-Object-Counting-API

The code of the Object Counting API, implemented with the YOLO algorithm and with the SORT algorithm

GPL-3.0 License

46 stars 19 forks

Star Watch

Code Pull requests Actions Security Insights

master Go to file

tugot17 Update README.md ... on 10 Apr 580

View code

README.md

YOLO-Object-Counting-API

Real time Object Counting api. Implemented with the YOLO algorithm and with the SORT algorithm

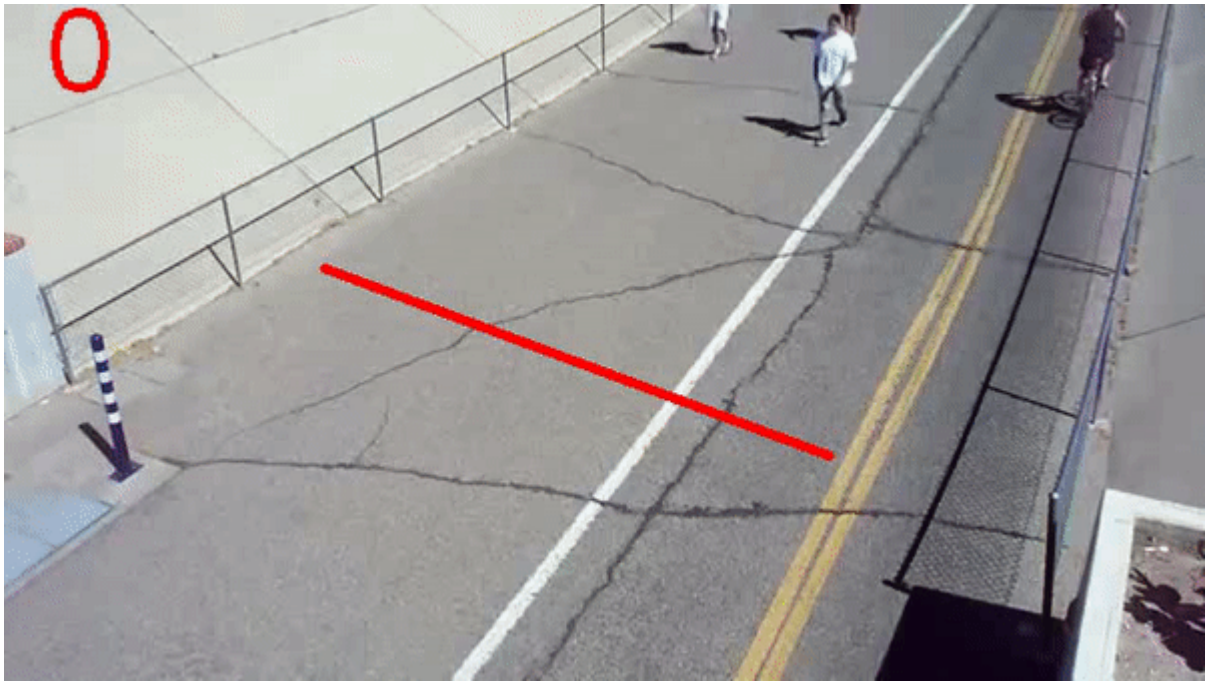
The implementation is using model in same format as darkflow and darknet. Weight files, as well as cfg files can be found here. Darklow supports only YOLOv1 and YOLOv2. Support for YOLOv3 has not yet been implemented.

In order to achieve the best performance, you should have Cuda and tensorflow-gpu installed on Your device.

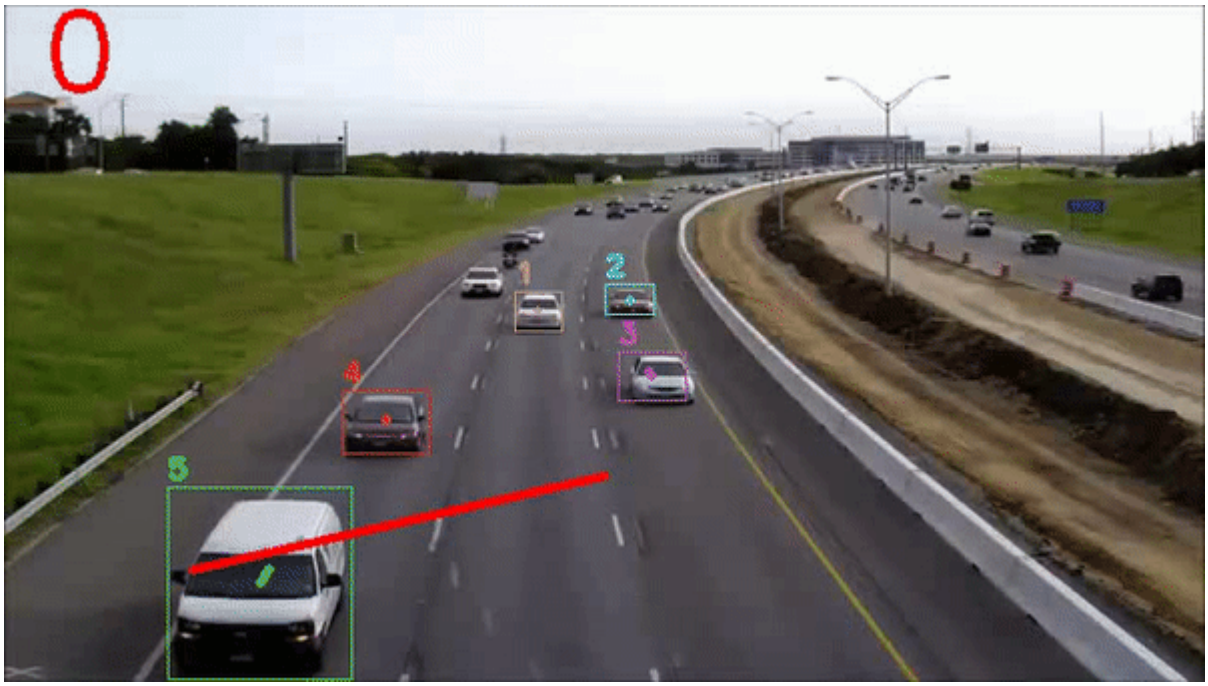
Demo

Count objects of a specified class crossing a virtual line

Counting pedestrains



Highway traffic counting



Count objects on a video

Count objects on a single frame

Set up

Dependencies

```
-tensorflow 1.0
-numpy
-opencv 3
```

Getting started

You can choose *one* of the following three ways to get started with darkflow.

1. Just build the Cython extensions in place. NOTE: If installing this way you will have to use `./flow` in the cloned darkflow directory instead of `flow` as darkflow is not installed globally.

   ```
   python3 setup.py build_ext --inplace
   ```

2. Let pip install darkflow globally in dev mode (still globally accessible, but changes to the code immediately take effect)

   ```
   pip3 install -e .
   ```

3. Install with pip globally

   ```
   pip3 install .
   ```

Required files

The YOLO algoritym impementation used in this project requires 3 files. Configuration of network (.cfg), trained weights (.weights) and labels.txt.

YOLO implementation used in this project enables usage of YOLOv1 and YOLOv2, and its tiny versions. Support for YOLOv3 has not yet been implemented.

.cfg files

Configuration file determines a network architecture. Configurations can be found here. In example scripts we assume that the configuration is placed in cfg/ folder. Location of used .cfg file is specyfied in the options object used in the code.

The .cfg file can be downloaded using the following command:

```
wget https://raw.githubusercontent.com/pjreddie/darknet/master/cfg/yolov2.cfg -O cfg
```

.weights files

The .weights files contain trained parameters of a network. In example scripts we assume the weights are placed in bin/ folder. Location of used .weights file is specyfied in the options object used in the code.

The .weights file can be downloaded using the following command:

```
wget https://pjreddie.com/media/files/yolov2.weights -O bin/yolov2.weights
```

labels.txt files

This file is list of classes detected by a YOLO netowork. It shoud contain as many classes as it is specyfied in a .cfg file.

Run counting

Once You have all dependencies instaled and all required files You can start counting objects. Object counting is carried out by an ObjectCuntingAPI object.

Examples of counting below

Count cars on crosing a virtual line

```
python3 count_cars_crosing_virtual_line.py
```

Count objects on video from Video Camera

```
python3 count_objects_from_camera.py
```

Count people on image

```
python3 count_people_on_image.py
```

Credits

The following open source projects were used in the implementation

Darkflow

The YOLO algorithm impementation - Darkflow

Python Traffic Counter

Object counting with YOLO and SORT. Similar project, but instead of using the darklow YOLO implementation, it uses the opencv YOLO implementation, so there is no GPU acceleration.

Deep Sort

Object tracking and counting - SORT

Images and Videos sources

Highway surveillance video

Pedestrian surveillance video

Authors

- tugot17

License

This project is licensed under the GNU General Public License v3.0 - see the LICENSE file for details

That's all.

Releases

No releases published

Packages

No packages published

Languages

- **Python** 100.0%