Training YOLO v3 for Objects Detection with Custom Data

Testing Darknet framework

Checking installation of Darknet framework

In order to detect objects by installed *Darknet framework*, navigate to the directory with executable file and type in *Terminal* or *Anaconda Prompt* specific command. Commands on *Linux* and *MacOS* from one side and commands on *Windows* from other side differ only by first word of executable file.

Commands on Linux and MacOS

All commands on *Linux* and *Mac* start from executable file that is inside root directory of *Darknet framework*:

```
./darknet
```

Commands on Windows

All commands on Windows start from executable file that is inside directory *darknet\build\darknet\x64* of *Darknet framework*:

```
darknet.exe
```

Pay attention! Working directory on Windows is ***darknet\build\darknet\x64***
Therefore, folders **cfg**, **data**, **weights** might need to be moved in the working directory, in order to be next to the executable file.

Commands to detect objects on image

Navigate to the directory with executable file of *Darknet framework* and type in *Terminal* or *command line* specific commands that are shown below.

General detection of objects on image

- **For Linux and MacOS** navigate to root directory where *Darknet framework* was installed and type in following command:
```
./darknet detector test cfg/coco.data cfg/yolov3.cfg weights/yolov3.weights
data/test-image.jpg
```

- **For Windows** navigate to *darknet\build\darknet\x64* and type in following command:
```
darknet.exe detector test cfg\coco.data cfg\yolov3.cfg weights\yolov3.weights
data\test-image.jpg
```

Thresholding while detecting

In order to **threshold objects** with weak predictions, add *thresholding argument* at the end of command *before image path* and specify *thresholding rate*: `-thresh 0.85`

- **For Linux and MacOS** navigate to root directory where *Darknet framework* was installed and type in following command:
```
./darknet detector test cfg/coco.data cfg/yolov3.cfg weights/yolov3.weights
-thresh 0.85 data/test-image.jpg
```

- **For Windows** navigate to *darknet\build\darknet\x64* and type in following command:
```
darknet.exe detector test cfg\coco.data cfg\yolov3.cfg weights\yolov3.weights
-thresh 0.85 data\test-image.jpg
```

Coordinates of detected objects

In order to **print coordinates** of bounding boxes around detected objects, add following *argument* at the end of command *before image path*: `-ext_output`

- **For Linux and MacOS** navigate to root directory where *Darknet framework* was installed and type in following command:
```
./darknet detector test cfg/coco.data cfg/yolov3.cfg weights/yolov3.weights
-ext_output data/test-image.jpg
```

- **For Windows** navigate to *darknet\build\darknet\x64* and type in following command:
```
darknet.exe detector test cfg\coco.data cfg\yolov3.cfg weights\yolov3.weights
-ext_output data\test-image.jpg
```

Commands to detect objects on video

Navigate to the directory with executable file of *Darknet framework* and type in *Terminal* or *command line* specific commands that are shown below. Resulted and processed video will be saved in root directory of *Darknet framework*: `result.avi`

General detection of objects on video

- **For Linux and MacOS** navigate to root directory where *Darknet framework* was installed and type in following command:

```
./darknet detector demo cfg/coco.data cfg/yolov3.cfg weights/yolov3.weights
data/test-video.mp4 -out_filename result.avi
```

- **For Windows** navigate to *darknet\build\darknet\x64* and type in following command:

```
darknet.exe detector demo cfg\coco.data cfg\yolov3.cfg weights\yolov3.weights
data\test-video.mp4 -out_filename result.avi
```

Thresholding while detecting

In order to **threshold objects** with weak predictions, add *thresholding argument* at the end of command *before path to video* and specify *thresholding rate*: `-thresh 0.85`

- **For Linux and MacOS** navigate to root directory where *Darknet framework* was installed and type in following command:

```
./darknet detector demo cfg/coco.data cfg/yolov3.cfg weights/yolov3.weights
-thresh 0.85 data/test-video.mp4 -out_filename result.avi
```

- **For Windows** navigate to *darknet\build\darknet\x64* and type in following command:

```
darknet.exe detector demo cfg\coco.data cfg\yolov3.cfg weights\yolov3.weights
-thresh 0.85 data\test-video.mp4 -out_filename result.avi
```

Switch off window while processing video

In order to **switch off window** with processed frames while detecting, add *argument* at the end of command *before path to video*: `-dont_show`

- **For Linux and MacOS** navigate to root directory where *Darknet framework* was installed and type in following command:

```
./darknet detector demo cfg/coco.data cfg/yolov3.cfg weights/yolov3.weights
-dont_show data/test-video.mp4 -out_filename result.avi
```

- **For Windows** navigate to *darknet\build\darknet\x64* and type in following command:

```
darknet.exe detector demo cfg\coco.data cfg\yolov3.cfg weights\yolov3.weights
-dont_show data\test-video.mp4 -out_filename result.avi
```

Commands to detect objects in real time by camera

Navigate to the directory with executable file of *Darknet framework* and type in *Terminal* or *command line* specific commands that are shown below.

General detection of objects in real time

- **For Linux and MacOS** navigate to root directory where *Darknet framework* was installed and type in following command:
```
./darknet detector demo cfg/coco.data cfg/yolov3.cfg weights/yolov3.weights -c 0
```

- **For Windows** navigate to *darknet\build\darknet\x64* and type in following command:
```
darknet.exe detector demo cfg\coco.data cfg\yolov3.cfg weights\yolov3.weights -c 0
```

Thresholding while detecting

In order to **threshold objects** with weak predictions, add *thresholding argument* at the end of command *before camera* and specify *thresholding rate*: `-thresh 0.85`

- **For Linux and MacOS** navigate to root directory where *Darknet framework* was installed and type in following command:
```
./darknet detector demo cfg/coco.data cfg/yolov3.cfg weights/yolov3.weights -thresh 0.85 -c 0
```

- **For Windows** navigate to *darknet\build\darknet\x64* and type in following command:
```
darknet.exe detector demo cfg\coco.data cfg\yolov3.cfg weights\yolov3.weights -thresh 0.85 -c 0
```

Useful Links

Check out these links with official resources for installing and using *Darknet framework* as well as the links to *possible issues*:

[1] Fork of AlexeyAB – the most popular fork of *Darknet framework* **chosen for this course**, that has improvements on performance, answers on the popular issues

[2] Issues - try to find answer to encountered issue from more than 2000 community posts on how to install and use *Darknet framework*