

01_DetectExamples

September 7, 2018

1 Detect objects from Python.

```
In [1]: import pydarknet2
```

```
classifier = pydarknet2.Classifier("cfg/coco.data", "cfg/yolov3.cfg", "weights/yolov3.weights")
classifier
```

```
Out[1]: Classifier<cfg/coco.data, cfg/yolov3.cfg, weights/yolov3.weights>
```

```
In [2]: classifier.network
classifier.metadata
```

Loading network...Done

Loading metadata...Done

```
Out[2]: Metadata<80>
```

Load image with `load_image_color` in the shared library.

Libdarknet paths are always relative to the darknet directory. Because reasons.

```
In [3]: import os
```

```
In [4]: dog_path = os.path.join(classifier.root, "data/dog.jpg")
```

1.1 From File Path:

Pass in the `file_path` to detect objects in it with specified classifier.

```
In [5]: classifier.detect(dog_path)
```

```
Out[5]: [Classified<dog, (134.83998107910156, 214.18678283691406, 313.66184997558594, 542.5165...
Classified<truck, (476.8968048095703, 81.5092544555664, 684.7720184326172, 168.592765...
Classified<bicycle, (99.2535400390625, 124.20358276367188, 589.5813598632812, 447.620...
```

1.2 From Darknet image structure

```
In [6]: img = classifier.load_image_color(dog_path)
        img
```

```
Out [6]: Image<768x576x3>
```

```
In [7]: classifier.detect(dog_path)
```

```
Out [7]: [Classified<dog, (134.83998107910156, 214.18678283691406, 313.66184997558594, 542.5165
Classified<truck, (476.8968048095703, 81.5092544555664, 684.7720184326172, 168.592765
Classified<bicycle, (99.2535400390625, 124.20358276367188, 589.5813598632812, 447.620
```

1.3 From Pillow

<https://python-pillow.org/>

... the friendly PIL fork. PIL is the Python Imaging Library.

```
In [8]: import PIL.Image
```

```
In [9]: img = PIL.Image.open(dog_path)
        img
```

```
Out [9]:
```



```
In [10]: classifier.detect(image=img)
```

```
Out[10]: [Classified<dog, (134.86256408691406, 214.7501220703125, 313.55186462402344, 542.1865...
          Classified<truck, (477.36522674560547, 81.4034652709961, 684.870735168457, 168.99848...
          Classified<bicycle, (101.23869323730469, 125.75776672363281, 587.3878326416016, 447.7...
```

1.4 From imageio

<https://imageio.github.io/>

Imageio is a Python library that provides an easy interface to read and write a wide range of image data, including animated images, video, volumetric data, and scientific formats. It is cross-platform, runs on Python 2.7 and 3.4+, and is easy to install.

```
In [11]: import imageio
```

```
In [12]: img = imageio.imread(dog_path)
```

```
In [13]: classifier.detect(image=img)
```

```
Out[13]: [Classified<dog, (134.86256408691406, 214.7501220703125, 313.55186462402344, 542.1865...
          Classified<truck, (477.36522674560547, 81.4034652709961, 684.870735168457, 168.99848...
          Classified<bicycle, (101.23869323730469, 125.75776672363281, 587.3878326416016, 447.7...
```

1.5 From OpenCV

<https://opencv.org/>

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

Getting OpenCV Python3 library:

1. `sudo apt-get install python3-opencv` on Ubuntu 18.04.
2. Compile opencv2 and install in your virtualenv.

```
In [14]: # For working in a virtual environment, add the system packages.
import sys
sys.path.append("/usr/lib/python3/dist-packages/")
```

```
In [15]: import cv2
```

```
In [16]: img = cv2.imread(dog_path)
```

Swap color order.

```
In [17]: img_ = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

```
In [18]: classifier.detect(image=img_)
```

```
Out[18]: [Classified<dog, (134.83998107910156, 214.18678283691406, 313.66184997558594, 542.5168111111111),  
          Classified<truck, (476.8968048095703, 81.50925445555664, 684.7720184326172, 168.5927611111111),  
          Classified<bicycle, (99.2535400390625, 124.20358276367188, 589.5813598632812, 447.6211111111111)]
```