What is YOLO?

YOLO (You Only Look Once) is a method / way to do object detection. It is the algorithm /strategy behind how the code is going to detect objects in the image.

Earlier detection frameworks, looked at different parts of the image multiple times at different scales and repurposed image classification technique to detect objects. This approach is slow and inefficient.

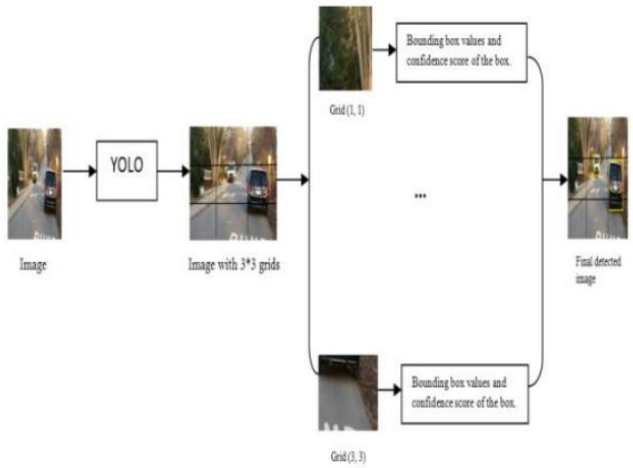
YOLO takes entirely different approach. It looks at the entire image only once and goes through the network once and detects objects. Hence the name. It is very fast. That's the reason it has got so popular.

There are other popular object detection frameworks like Faster R-CNN and SSD that are also widely used.

In this post, we are going to look at how to use a pre-trained YOLO model with OpenCV and start detecting objects right away.

Working of YOLO!

First, an image is taken and YOLO algorithm is applied. In our example, the image is divided as grids of 3x3 matrixes. We can divide the image into any number grids, depending on the complexity of the image. Once the image is divided, each grid undergoes classification and localization of the object. The objectness or the confidence score of each grid is found. If there is no proper object found in the grid, then the objectness and bounding box value of the grid will be zero or if there found an object in the grid then the objectness will be 1 and the bounding box value will be its corresponding bounding values of the found object. The bounding box prediction is explained as follows. Also, Anchor boxes are used to increase the accuracy of object detection which also explained below in detail.



Process of Implementation YOLO in python

- **Installing dependencies**
  Following things are needed to execute the code we will be writing.
  - Python 3
  - Numpy
  - OpenCV Python bindings

- **Python 3**
  If you are on Ubuntu, it's most likely that Python 3 is already installed. Run python3 in terminal to check whether its installed. If its not installed use

  sudo apt-get install python3

  For macOS please refer my earlier post on deep learning setup for macOS.

  I highly recommend using Python virtual environment. Have a look at my earlier post if you need a starting point.

- **Numpy**

  pip install numpy

  This should install numpy. Make sure pip is linked to Python 3.x (pip -V will show this info)

  If needed use pip3. Use sudo apt-get install python3-pip to get pip3 if not already installed.

- **OpenCV-Python**

  You need to compile OpenCV from source from the master branch on github to get the Python bindings. (recommended)

Adrian Rosebrock has written a good blog post on PyImageSearch on this. (Download the source from master branch instead of from archive)

If you are overwhelmed by the instructions to get OpenCV Python bindings from source, you can get the unofficial Python package using

pip install opencv-python

This is not maintained officially by OpenCV.org. It's a community maintained one. Thanks to the efforts of Olli-Pekka Heinisuo.

- **Command line arguments**

  The script requires four input arguments.

  - input image
  - YOLO config file
  - pre-trained YOLO weights
  - text file containing class names

  All of these files are available on the github repository I have put together. (link to download pre-trained weights is available in readme.)

  You can also download the pre-trained weights in Terminal by typing

  wget https://pjreddie.com/media/files/yolov3.weights

  This particular model is trained on COCO dataset (common objects in context) from Microsoft. It is capable of detecting 80 common objects. See the full list here.

  Input image can be of your choice. Sample input is available in the repo.

  Run the script by typing

  $ python yolo_opencv.py --image dog.jpg --config yolov3.cfg --weights yolov3.weights --classes yolov3.txt