

Object detection with YOLOX

Tomasz Stańczyk
tomasz.stanczyk@inria.fr

Inria

3iA Côte d'Azur
Interdisciplinary Institute
for Artificial Intelligence

Overview

In this practical assignment you are going to get familiar with YOLOX and use it to perform pointed tasks

You can either:

- Work on Google Colab
- Work on your local machine (if you have a GPU)
- On the university cluster (if you have an access)

YOLOX github repository:

<https://github.com/Megvii-BaseDetection/YOLOX>



Working on Google Colab

Go to your Google Drive

Select New -> More -> Google Colab

Change the runtime: Runtime -> Change runtime type -> Hardware accelerator -> GPU

Everytime you want to run a terminal-like instruction in a code shell, you need to precede the whole instruction with an exclamation mark, e.g.

!git clone... or !python file.py...

If you are changing directory with *cd*, you need to do it as a one instruction with the proceeding command:

!cd && python file.py...

You can save your output (ctrl+S), edit files (double click) in and upload them (e.g. drag and drop) into your Google Colab space, but mind that they might be automatically erased after some time. Thus do make copies (download them into your PC, right click -> download)

Working on your local machine

Only if you have a GPU

Get familiar with Anaconda: <https://www.anaconda.com/>

Create a virtual environment with Python

Activate the environment and work on it

In this manner, you will avoid package conflicts with your other projects

YOLOX github

It contains the instructions on how to set it and run it (**=everything is there**)

To clone it, use:

```
!git clone github.com/Megvii-BaseDetection/YOLOX.git
```

(No need for an exclamation mark if working locally on your machine)

See Quick Start -> Installation, Demo

Try to read the instructions and adapt them where needed (e.g. the path to model weights, the .pth files)

You can download the model weights from the links above the Quick Start:

Benchmark -> Standard models

Use *!wget model_link* to download them directly into your space

!mkdir your_dir to create a separate directory for the weights

Use consistent weights with model architecture, e.g. [...] *-n yolox-l -c your_dir/yolox_l.pth [...]*

Your tasks

1. Run demo as described on the github page
2. Run YOLOX on 3 images which you upload yourself (=different than in ./assets directory), save outputs to a text file.
3. Run YOLOX on a video (upload one there yourself), save outputs to a text file.
4. Find the file(s) of the model (neural architecture) and see if you understand particular components. Even broad understanding is already good.

More instructions on the next page!

Your tasks - remarks

```
image.png
1 25 40 100 150 0.87 cat
2 136 420 120 190 0.73 dog
```

Mind that for 2. and 3. you will need to provide/add extra lines of code on your own and potentially modify some of the existing parts. (= you need to get your hand "dirty" and delve into the code)

The following .txt output is expected from 2.:

image_name.extension (optional)
object_no left top width height confidence class

Image name only once, whereas the next line for each object separately

Object number (*object_no*) is just for enumeration, to see how many objects were there

left, top, width and *height* are integers (rounding down) of the bounding box top-left corner and its size

confidence is a floating point number between 0.0 and 1.0

class is the object predicted class (a string)

Your tasks - remarks

The following .txt output is expected from 3.:

```
video.mp4
1 1 25 40 100 150 0.87 cat
1 2 136 420 120 190 0.73 dog
2 1 26 39 100 151 0.84 cat
2 2 138 423 121 187 0.75 dog
```

video_name.extension (optional)

frame_no object_no left top width height confidence class

Analogously, video name only once, whereas the next line for each object per frame separately (and done for each frame)

frame_no is the frame number (as you will be processing video as images, frame by frame)

Other parameters as for the single image processing

Good luck!

Do it earnestly

It will help you understand how YOLOX works and how to use repositories with existing solutions for your purposes

Besides, next week's tracking assignment will be based on YOLOX

Your work will be briefly checked during the next class. Don't be afraid to come though, it is all for you to learn :)

When facing issues and errors, google it at first and see if someone has had such an issue before. That's how you learn in practice

Feel free to discuss the stuff with your classmates, but ensure it is you who does and understands the assignment

If you are seriously stuck and cannot proceed, you can contact me via e-mail

Happy learning!