

# Assignment 5: YOLO Object Detection on PASCAL VOC

Due date: Wednesday, November 29, 11:59:59 PM



In this assignment you will implement a YOLO-like object detector on the PASCAL VOC 2007 dataset to produce results like in the above image.

The top-level notebook ([A5.ipynb](#)) will guide you through all the steps. You will mainly focus on implementing the loss function of YOLO in the `yolo_loss.py` file. You will be provided a pre-trained network structure for the model. The network structure has been inspired by [DetNet](#), however you are not required to understand it. In principle, it can be replaced by a different network architecture and trained from scratch, but to achieve a good accuracy with a minimum of computational expense and tuning, you should stick to the provided one.

As you start this part, you will realize that training is more computationally intensive than what you are used to. In order to get an idea whether your implementation works without waiting a long time for training to converge, here are some typical values to expect:

## Epoch mAP

5	0.
10	0.032
15	0.177
20	0.256
25	0.291
30	0.318
35	0.352
40	0.352
45	0.363
50	0.364

## Useful Resources

The instructions in the `yolo_loss.py` file should be sufficient to guide you through the assignment, but it will be really helpful to understand the big picture of how YOLO works and how the loss function is defined.

The following resources are useful for understanding YOLO in detail:

- [Original YOLO paper](#) (**recommended**)
- [Great post about YOLO on Medium](#)
- [Differences between YOLO, YOLOv2 and YOLOv3](#)
- [Great explanation of the Yolo Loss function](#) (**recommended**)

## Extra Credit (15 points, the extra points will be given based on your effort)

- Pick a fun video like [this one](#), run your detector on it (a subset of frames would be OK), and produce a video showing your results.
- Try to replace the provided pre-trained network with a different one and train with the YOLO loss on top to attempt to get better accuracy.
- Or any other methods that you try to improve the performance.

## Report

In your report, please include:

- a. A brief discussion on your implementation
- b. Report the best train and validation accuracy in all of your experiments and discuss any strategies or tricks you've employed.
- c. Report the results for extra credits and also provide a discussion, if any

## Submission Instructions:

The assignment deliverables are as follows.

1. Upload files to CU:
  1. **sid\_a5\_output.pdf**: Your IPython notebook with output cells converted to **PDF format**
  2. **yolo\_loss.py**: Your main YOLO **python** implementation for ease of inspection
  3. **sid\_a5\_code.zip**: All of your code (python files and ipynb file) **in a single ZIP file**
  4. **sid\_a5\_report.pdf**: Your assignment report **in PDF format**
  
2. Upload your YOLO output file to the [Kaggle competition](#) for the YOLO detector.

## Grading:

Your grade will be based on Kaggle competition (85%) and the report (15%).