

# TRAINING 2 - Vision

## **1 - Classifier - LeNet**

Implement the well known LeNet model for predicting handwritten digits

<https://medium.com/@mgazar/lenet-5-in-9-lines-of-code-using-keras-ac99294c8086>

## **2 - Object Detection - YOLO (PyTorch)**

Try to implement the well known [YOLO](#) algorithm:

<https://blog.paperspace.com/how-to-implement-a-yolo-object-detector-in-pytorch/>

## **3 - Serving models**

Let's serve the models you implemented. Allowing "clients" to call it

<https://www.pyimagesearch.com/2018/01/29/scalable-keras-deep-learning-rest-api/>

## **Cool applications:**

- Colorize image (PyTorch) - [Demo](#)  
<https://lukemelas.github.io/image-colorization.html>
- Neural style : [Demo](#)  
<https://github.com/leongatys/PytorchNeuralStyleTransfer>
- Human Pose estimation:  
<https://www.learnopencv.com/deep-learning-based-human-pose-estimation-using-opencv-cpp-python/>