

Computer Vision

James L. Crowley and Nachwa Aboubakr

MoSIG M2
Project 2:

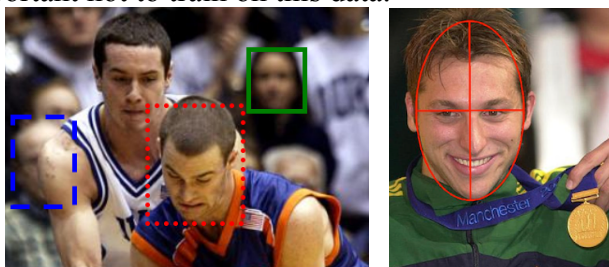
Fall Semester
15 October 2020

Face Detection using a Multi-layer Perceptron.

The objective for this exercise is to use the Keras Library to classify an image window as face or not. You can use Jupyter Notebook or Google Colab to write your code. Test data and ground truth are provided by the “FDDB: Face Detection Data Set and Benchmark” of the University of Massachusetts.

This exercise is composed of three parts.

- 1) Construct a data-set of Positive and Negative face imagettes from the first 8 folds of FDDB. Divide this data set into Training Data and Validation data. The validation data is used to test for over-fit.
- 2) Construct and train a MultiLayer Perceptron (MLP) with variations in hyper-parameters. Train and validate each MLP using your test and validation data-sets.
- 3) Evaluate each MLP with the evaluation data set available on the Course web site, and display an ROC curves for each MLP. Note that the evaluation data has been constructed from folds 9 and 10 of FDDB, to is important not to train on this data.



Face Boxes

Face Ellipses

To do this exercise, you’ll need first to activate your python workspace, and install the Keras library under your conda environment. You should then, do the following in a jupyter notebook:

- 1) Split FDDB dataset (folds 01-08) into training and validation sets.
- 2) Generate training samples from dataset folds (01-08). Your training samples need to include a positive imagette for each face, and a negative imagette of the same size taken at random from the same image.
- 3) Build your MLP models using Keras. Remember to resize and flatten your input images into a fixed size vector before passing them through MLP network.
- 4) Download the test set, and test your model on the provided imagettes.
- 5) Tune your model parameters, and post your best model results on

<https://docs.google.com/spreadsheets/d/11UksOcAzK5to90rjXrIYiUjKk4QbrI4TaJhYLrio1eM/edit?usp=sharing>

Hyper-Parameters to tune:

- Face imagette resolution (input vector size): (e.g WxH: 8x8, 16x16, 16x24, 32x32, ...)
 - MLP architecture: number of hidden layers, number of units, activation function, learning rate, etc.
- 6) Document your work in the Jupyter Notebook by commenting it and send the .ipynb file to: James.Crowley@inria.fr, Nachwa.Aboubakr@inria.fr, Yangtao.Wang@inria.fr