SIMPLE FACE DETECTION

0.1 Goal

The goal is to get basic practical experience with SVM classification as well as with the visual object category detection in still images. A simple face detector based on the common "scanning-window" technique is presented. The implementation of the detector will contain the following steps:

- **Part 1**: Load, format and normalize positive and negative training images
- **Part 2**: Learn and evaluate linear SVM classifier; choose the best C value
- **Part 3**: Use SVM classifier to score image patches and to detect faces in test images

0.2 Exercise description

0.2.1 Part 1: Preparing training data

Go through the steps of loading and visualization training images. Run mean-variance normalization, then format images into SVM-acceptable input by running provided lines of the code. Make sure you understand the format of variables Xtrain, ytrain, Xval, yval as you will need to operate with them in the next steps.

0.2.2 Part 2: SVM classification

Train and test a linear SVM classifier. SVM training and evaluation is done with svmclass and svmvalmod functions respectively. Next, implementing the following steps:

- Compute linear hyper-plane W from SVM support vectors and alpha- coefficients
- Re-compute confidence values for training and validation using W and bias b. Make sure your accuracy values correspond to the ones returned by svmvalmod

Illustrate W as an image using the provided code. Why does it remind a face? How do you explain different values of W? Next, re-train SVM for different values of C:

- Fill-in the for-loop to train SVM for the changing C-values, compute W and classification accuracy for training and test samples in each iteraction. Select SMV model maximizing accuracy on the validation set.
- Visualize W as an image at each iteration. Why W looks more like a face for small C-values?

The best classification hyper-plane W looks like an average face image. Cannot we just use such an average image as a classification hyper-plane?

0.2.3 Part 3: Face detection

Follow the provided code and its comments to read a test image; extract its overlapping pixel patches and use linear SVM to classify the patches. Display bounding boxes of patches with the highest classification score.

Scanning-window style classification of image patches typically results in multiple responses around the target object. A standard practice to deal with this is to remove any detector responses in the neighborhood of detections with the locally maximal confidence score (non-maxima suppression or NMS). NMS is usually applied to all detections in the image with confidence above a certain threshold. Try NMS face detection for different threshold values and in different images:

- Try different threshold values to pre-select windows passed to the NMS step by modifying parameter confthresh.
- Try different threshold values for NMS detections by modifying confthreshnms
- Try detection with the different thresholds for images: img1.jpg, img2.jpg, img3.jpg, img4.jpg. Can you find unique NMS threshold giving perfect face detection in all images?