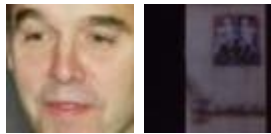


The code can be split into several parts:

Read-in training pictures (line 11 to line 27)

We have 4000 pictures on hand, 2000 of which are faces, when the rest are common backgrounds without faces. The dimension of each picture is 64 * 64 pixels. Here is an example of face picture and a background picture:



Determine features (line 28 to line 59, line 202 to line 222)

https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework

This part contains three functions:

- `feature_pairs` sets the location points using Haar like features;
- `compute_feature` calculates the feature of certain areas determined by 8 coordinators (two rectangles); if the area only has 6 coordinators, use `compute_feature2`.

Find weak classifier (line 60 to line 87)

https://en.wikipedia.org/wiki/Decision_stump

Adaptive boosting algorithm only shows how to integrate multiple weak classifiers into an effective strong classifier and the criteria (minimize exponential loss) of finding weak classifier, but not suggest the method to build weak classifiers.

Here I use decision stump (one-level decision tree) with varying threshold ($\hat{\theta}$) to find reasonable parameters and build weak classifier.

Adaboost (line 88 to line 180)

<https://en.wikipedia.org/wiki/AdaBoost>

This is the main algorithm of the facial recognition. The adaptive boosting algorithm is slightly altered, the hypothesis returned by regular AdaBoost is $h(x) = \text{sgn}(\sum_{t=1}^T \alpha_t h_t(x))$, However, in a classifier cascade it is critical that each classifier have a low false negative rate. So instead, what I am using here is $h(x) = \text{sgn}(\sum_{t=1}^T \alpha_t h_t(x) - \Theta)$, in which Θ ensures that there is no false negative in the training pictures at all.

The code consists of three functions:

- **AdaBoost** is the main function. It includes the method to find optimal weak classifier (line 106 to line 120 by minimizing error), update the weights, construct the strong classifier, calculate the errors and check whether it satisfies the criteria (false positive rate falls below about 30%, false negative falls below 5%);
- **booster_fit** is the function to obtain strong classifier by calculating weighted sum of weak classifiers, wrapped in **AdaBoost**;
- **errors** is the function that calculates the prediction errors, including false positive and false negative rates.

Cascading (line 181 to line 193)

The method and reason for cascading is well-explained in the original paper: http://www.cs.utexas.edu/~grauman/courses/spring2007/395T/papers/viola_cvpr2001.pdf

Test on new pictures (line 194 to end)

Two ways of screening through the whole picture: first loop x then loop y, and first loop y then loop x. The results are the following:

