In this MP, we use Adaboost algorithm combined with Viola-Jones features to conduct face recognition of images. We trained 40 iterations of weak classifier use 126 labeled images, and constructed a strong classifier for face detection of the rest 42 images. In this report, we report unweighted error rates of both weak and strong classifiers and also weighted error rate of weak classifiers in each iteration.

First, we assume that we have $N$ training images, and each of them have $R$ labeled rectangle. The general Adaboost Algorithm is shown as following.

Initialize a weight matrix $W\_1$, such that $W\_1\in\mathbb{R}^{N\times R}$.

At $t$ th iteration, we can find an optimal weak classifier $h\_t(\mathbb{X},p,\theta)$, and corresponding weighted error $\epslion\_t$

In this MP, we use Viola-Jones features for face detection. In each grayscale image $I$, we are able to find its integral image $II$ as

$$II(x,y)=\sum\_{i\leq x,j\leq y}i{ij} $$

For each labeled rectangle in the set $Y$, we have identify its location in image with four parameters $[X\_m,Y\_m,W,H]$, where $[X\_m,Y\_m]$ is the pixel coordinate of its upper right corner, and $W$ and $H$ are its width and height. Similarly, we can identify the location of some subrectangle in side as $[f\_X,f\_Y,f\_W,f\_H]$, such that the upper right corner of the subrectangle is $[X\_m+Wf\_X,Y\_m+Hf\_Y]$. The width and height of subrectangle is $[Wf\_W,Hf\_H]$.

In order to find the optimal subrectangle, we divide each labeled rectangle into $6\times6$ grids and do exhaustic searches to find a combination of subrectangle and feature order with minimum weighted error

We first introduce a weight parameter $\alpha\_t$ for each iteration $t$, and define it as

$$\alpha\_t=-\ln\frac{\epslion\_t}{1-\epslion\_t}$$

such that $\alpha\_t$ is monotonically decreasing function of $\epsilon\_t$. Then we can use our trained weak classifier to obtain a strong classifier for facial detection. First

Since we have 40 iterations of training, we obtained 40 optimal subrectangles for classification. First, we use our subrectangles to calculate scalar features of test images by function \texttt{rectfeature}. With trained threshold, and polarity, we are able to use weak classifier to obtain the classification $h\_t$ of our test images by equation 1.

After iterating through all trained classifier, we are able to construct a strong classifier $H\_t$ as following,