We will use an image database with total 13718 different images. The images are divided into three different sets: training set (60%), validation set (20%) and test set (20%). These percentage may change a little according to our actual training result.

For each image in training set, we will generate some randomized perturbation of it because of two benefits. First, including randomized perturbation will make the network more robust to the deviated and rotated faces [1(3)]. Second, we can control the number of perturbation of each image to make the database more balance. In the database we used, there are more images with label “HAPPINESS” and “NEURAL”, and less images with the rest labels. So, we will produce less perturbation for images with label “HAPPINESS” and “NEURAL”. This can solve the problem caused by imbalance database. The training episode is expected to set to 20.

After the training, we use validation set to optimize the parameter of CNN. After optimization, we use both training data and validation data to train the final model. Finally, we will use test data to test our final model. The evaluation criterial is accuracy. We will calculate the overall accuracy, the recall as well as the false positive rate of different labels.

In order to deal with the overfitting issue, we will use the technic such as weight regulation, dropout and early stop. [2(book)].

Work plan

We will use GitHub to manage our project. Our project is mainly divided into four parts: data pre-processing, CNN building, optimization and visualization. There are totally three people in our project team.

1. Data pre-processing: analysis the original data (the number of data that belongs to different labels), data rescaling, converting to greyscale and randomized perturbation.
2. Building the CNNs and multiple network learning frame work.
3. Data visualization and most work of optimization. (The rest two teammate will also take part in optimization)

We plan to build a prototype CNN in 11.20, using a smaller data set. After building the prototype, we will evaluate our method and make changes accordingly. Finally, we will train our final model.