# ECEN 649 : pattern recognition

# HW5 Report

Hsing-Yu Chen

ECEN, Texas A&M University

UIN 829009530

# Fall, 2020

* **Project Description**

In this project, I programmed Adaboost to train / test classifiers using given data. The code is written in Python 3.8 on Anaconda environment. My project can be accessed at: <https://github.com/peterchen3301/Viola-Jones-Adaboost.git>

* **Imported tools:**

Following packages are imported in this project:

*Numpy* : for faster image / array manipulation

*OpenCV* : only for reading and showing images, without using any built-in classifier functions.

*Matplotlib* : for demonstration of performance metrics

*JSON* : for recording the training results

* **Training / Testing datasets:**

In this project, I use CMU face recognition dataset that consists 500 positive (face) images and 2000 negative (non-face) samples in training set; 472 positive, 2000 negative in test set. All images are grayscale, 19 \* 19 in size.

* **Extracting Haar features:**

I defined 9 feature patterns that maps along the sample image with various sizes and positions. 8055 Haar features are generated in total.

haar0 = [ [1, -1], [1, -1] ]

haar1 = [ [1, 1], [-1, -1] ]

haar2 = [ [1, -1, 1], [1, -1, 1] ]

haar3 = [ [1, 1], [-1, -1], [1, 1] ]

haar4 = [ [1, -1], [-1, 1] ]

haar5 = [ [1, 1, 1], [1, -1, 1], [1, 1, 1] ]

haar6 = [ [1, 1, 1], [-1, -1, -1], [1, 1, 1] ]

haar7 = [ [1, -1, 1], [1, -1, 1], [1, -1, 1] ]

haar8 = [ [1, 1, 1], [-1, -1, -1] ]

These Haar features define feature values from given sample images, which are transferred to integral images in order to speed up the calculations of rectangular sums.

* **Implementing ERM for decision stumps**

In this program, we train each weak classifier (corresponding to each of 8055 Haar features) by looping every training samples and get the best threshold (decision stump), which is the feature value at where the minimal sample error occurs. The definition of sample error is:

Where , note for the total weight of positive / negative samples, while , note for the accumulated positive / negative weight till current iteration.

The best threshold for each weak classifier is used to determine whether it classifies a sample correctly if following equation holds and *vice versa*.

* **Implementing Adaboost predictor**

In each round of training, all 8055 weak classifiers loop over training samples to solve for total weighted error. After completion, we select the one with lowest error and append it as one of the cascades in the strong classifier.

In this project, we are required to train 10 rounds and get the cascades of size 1, 3, 5 and 10 within the strong classifier.

* **Answers to given problems**

1. The 10 best weak classifiers obtained from my program are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # of cascade | Pattern | Size (in pixels) | Position (upper left) | Training error |
| 1st | |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  | | 3 \* 3 | (10, 3) | 0.18 |
| 2nd | |  |  | | --- | --- | |  |  | |  |  | | 2 \* 2 | (1, 17) | 0.26 |
| 3rd | |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | | 2 \* 3 | (0, 15) | 0.36 |
| 4th | |  |  | | --- | --- | |  |  | |  |  | | 2 \* 2 | (2, 15) | 0.48 |
| 5th ~ 10th | |  |  | | --- | --- | |  |  | |  |  | | 2 \* 2 | (2, 15) | 0.50 |

1. The combined strong classifier is equivalent to the cascade of 1, 3, 5 and 10 weak classifiers in this project, where: