**ECEN 649, Fall 2019**

Texas A&M University

Electrical and Computer Engineering Department

Dr. Tie Lui

Project Report

**Face Detection using AdaBoost and Haar Features**

**Name: Abhay Vashist**

**UIN: 524008495**

**Date: December 9, 2019**

**Aggie Code of Honor**

**An Aggie does not lie, cheat or steal or tolerate those who do.**



Abstract

The main focus of the project is the implementation of the Viola-Jones Algorithm. The algorithm was a breakthrough in the field of face detection because it provided computationally feasible and accurate results. The algorithm uses a Haar features and Adaboost algorithm using a decision stamp classifier to perform the classification. In the project, we are extracting 5 different types of features types from 19 by 19 images. Then implement the Adaboost algorithm on the features to perform the classification of the image. Finally, the classifier error definition is changed to consider only the case of False negative and False positive.

Features extraction

Haar Features

Table 1 (Feature Count)

|  |  |
| --- | --- |
| Type 1 (Horizontal Edge Detection) | 7440 |
| Type 2 (Vertical Edge Detection) | 7440 |
| Type 3 (Horizontal Line Detection) | 3462 |
| Type 4 (Vertical Line Detection) | 3462 |
| Type 5 (Four Feature detection) | 3600 |
| Total | 25424 |

The feature size was limited by two constraints, one that area of the two black and white regions must be equal and the max size of the filter is 8 by 8. The computations were performed using an integral image to decrease computation time.

Type 1 Horizontal Edge Detector Sizes of m by n.

m = range[1,8], n =2\* range[1,4]

Type 2 Vertical Edge Detector Sizes of m by n.

m = 2\*range[1,4], n = range[1,8]

Type 3 Horizontal Line Detector Sizes of m by n

m = range[1,8], n = 4\*range[1,2]

The smallest size has to be 1 by 4, to insure the we have 2 white pixels for 2 black pixels.

Type 4 Vertical Line Detector Sizes of m by n

m = 4\*range[1,2], n = range[1,8]

The smallest size has to be 4 by 1, to insure the we have 2 white pixels for 2 black pixels.

Type 5 Four Feature Detector Sizes of m by n

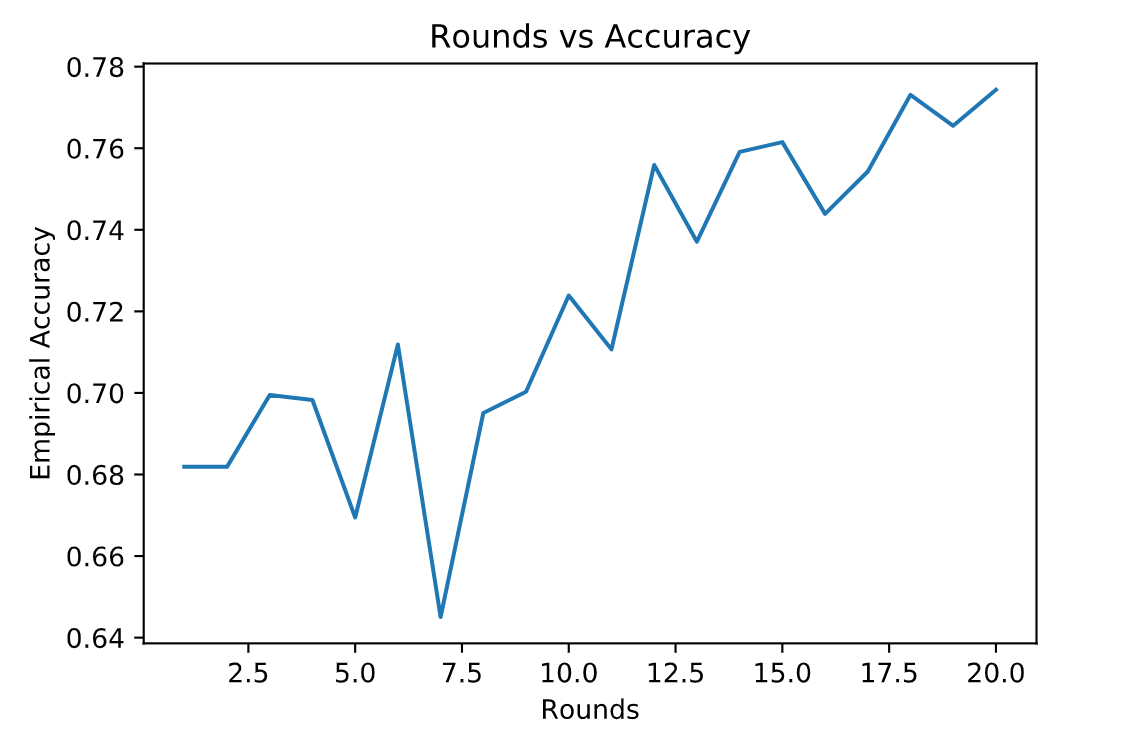
m = 2\*range[1,4], n = 2\*range[1,4]

The smallest size has to be 2 by 2, to insure one pixel per area.

AdaBoost

Table 2 (Round Statistics)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Round | Empirical | False Positive | False negative | Feature Type | Strong Predictor accuracy |
| 1 | 0.31813 | 0.09364 | 0.22449 | Type 1 | 0.68187 |
| 3 | 0.44498 | 0.07923 | 0.36575 | Type 1 | 0.69948 |
| 5 | 0.47539 | 0.09124 | 0.38415 | Type 2 | 0.66947 |
| 8 | 0.42497 | 0.17887 | 0.17967 | Type 1 | 0.69508 |
| 10 | 0.44458 | 0.08523 | 0.35934 | Type 1 | 0.72389 |

Figure 1 (Empirically accuracy over rounds)

Round 1

(Type 1) Horizontal Edge Detection

M = 6

N = 1

Position = (0, 0)

Threshold = -85.5

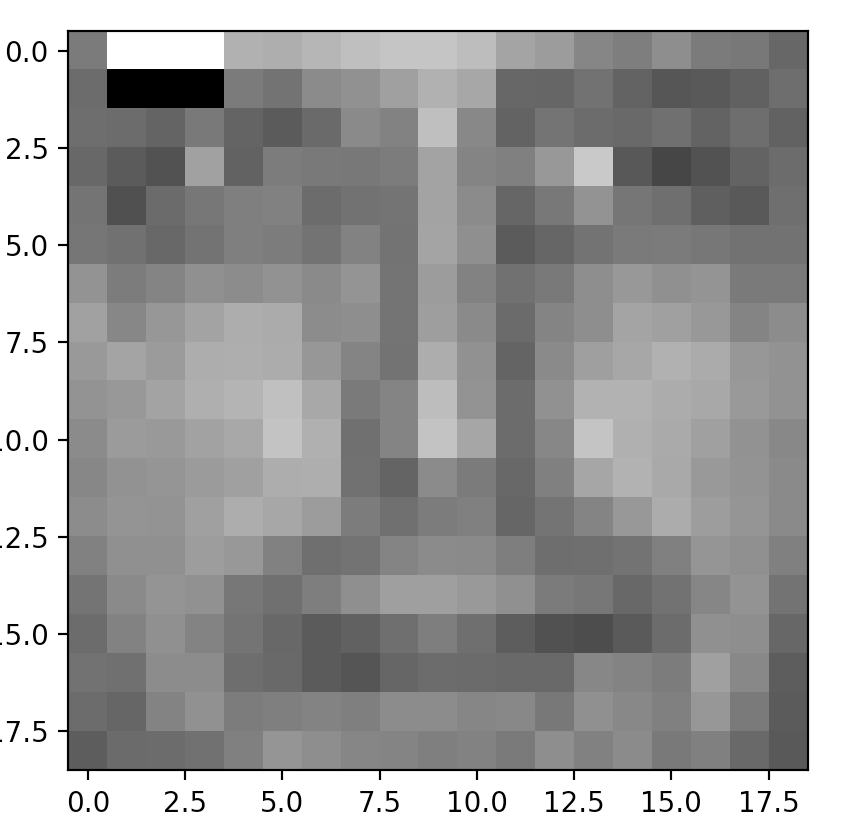


Figure 2(Round 1 features)

Round 3

(Type 2) Vertical Edge Detection

M = 4

N = 1

Position = (9, 16)

Threshold = -18.5]

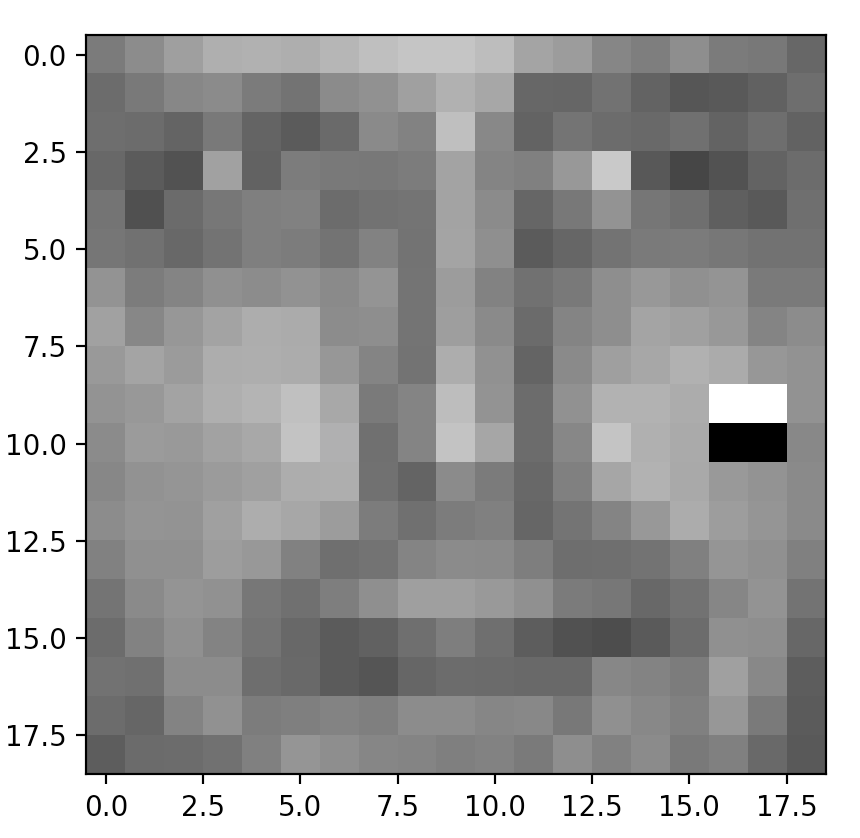


Figure 3(Round 3 features)

Round 5

(Type 2) Horizontal Edge Detection

M = 2

N = 2

Position = (9, 14)

Threshold = -16.5

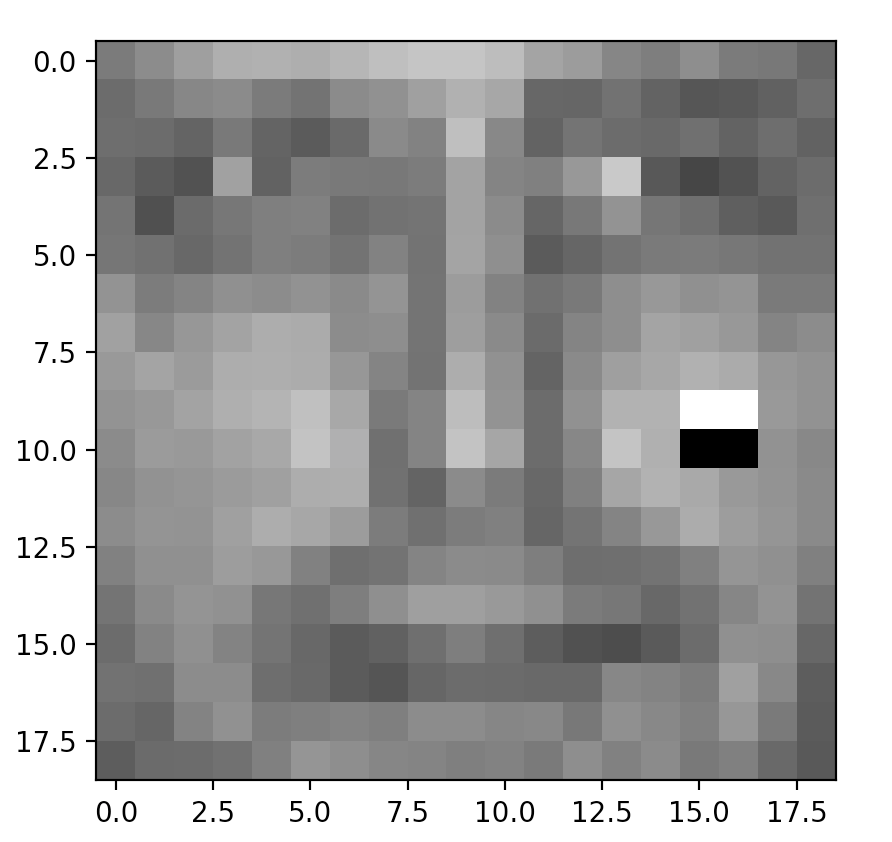


Figure 4(Round 5 features)

Round 8

(Type 1) Horizontal Edge Detection

M = 4

N = 1

Position = (7, 16)

Threshold = -48.5

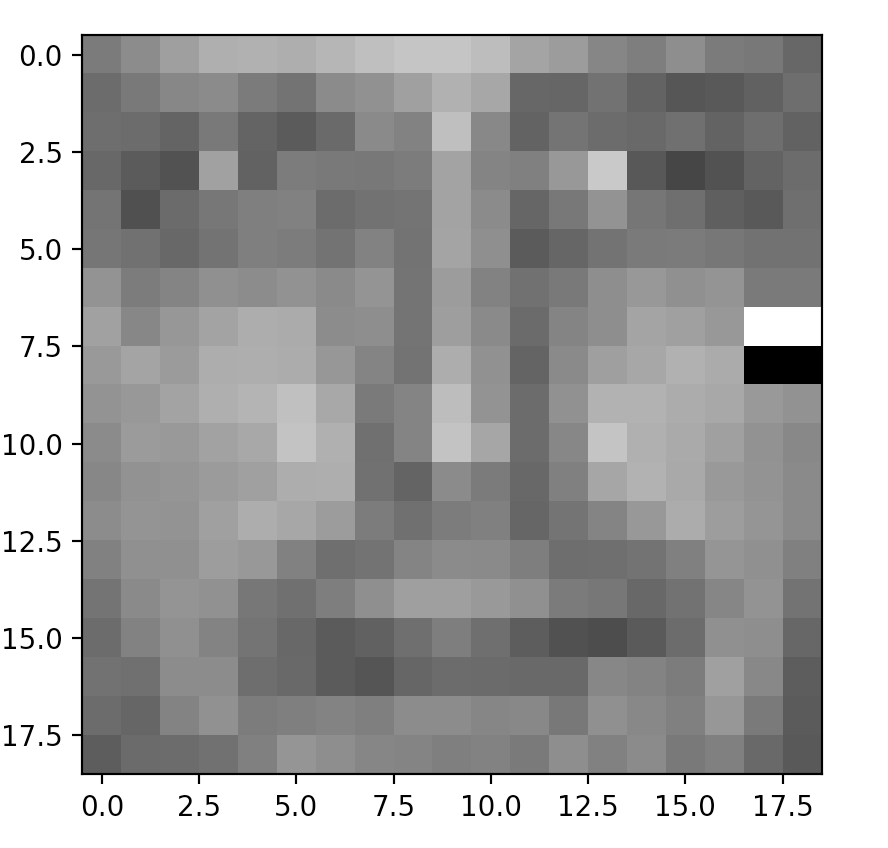


Figure 5(Round 8 features)

Round 10

(Type 1) Horizontal Edge Detection

M = 4

N = 1

Position = (8, 16)

Threshold = -22.5

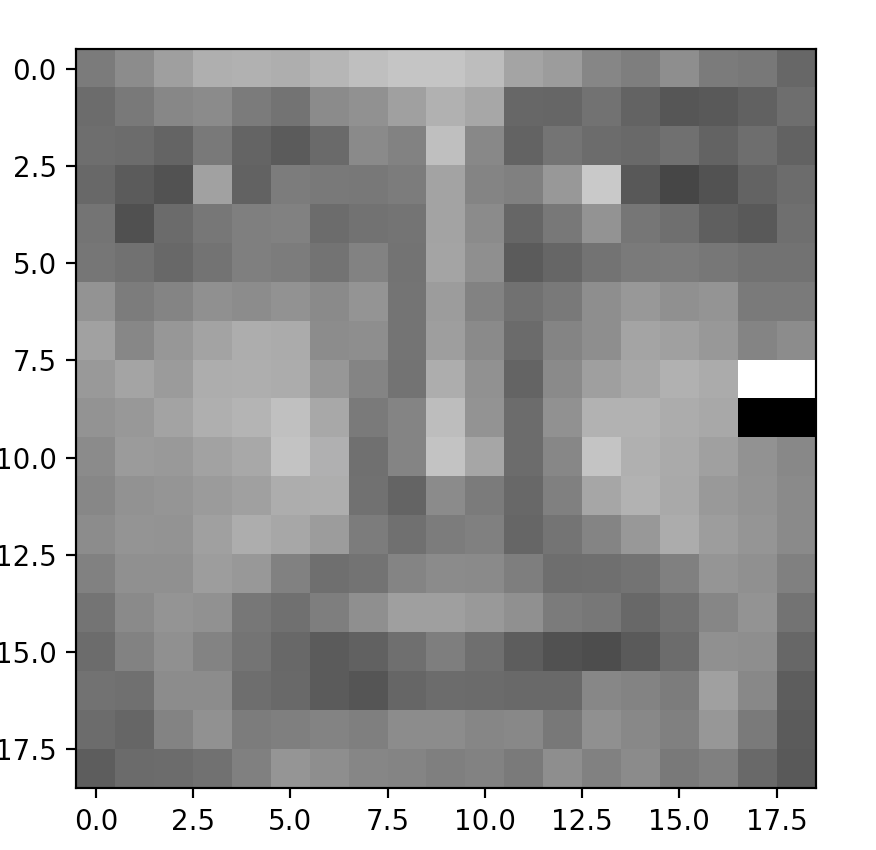


Figure 6(Round 10 features)