Viola Jones Project

Yipeng LU 829009541

1 Extract haar Features

Firstly, I created a class called ‘haarFeature’. The properties of the features, which are types, sizes, positions and values, are stored in the class. To extract all the features, I firstly looped over all the types. Then, I used a ‘nested for loop’ to loop over all the possible sizes of the feature. Lastly, I used another ‘nested for loop’ to loop over all the possible positions of the feature. With the restriction of the maximum size to be 8*8, the total number of hear features is 26168. There are 7440 type 1 features, 7440 type 2 features, 3844 type 3 features, 3844 type 4 features, and 3600 type 5 features.

2 Build an adaboost detector

I followed the procedures described in the paper, and my result is shown below. When measuring the accuracy of the detector, I gave positive images and negative images different weight, which are 1/(2*numPos) for positive images and 1/(2*numNeg) for negative images. Because the number of negative images is much more than the number of positive images, a simple detector labeling all images as negative will get a high accuracy rate if the weight of images is the same, which is not accurate.

For 1 round,

Feature 1: type is twoHorizontal, position is (8, 3), width is 2, height is 8, threshold is -106.000, polarity is 1, alpha is 2.798. Accuracy of the classifier is 0.882, false positive rate is 0.079(316/2000), false negative rate is 0.039(39/499).

Top feature is feature 1. The plot is shown in figure 1.

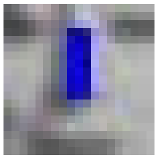

Figure 1

For 3 round,

Feature 2: type is twoHorizontal, position is (10, 8), width is 8, height is 4, threshold is -231.000, polarity is 1, alpha is 3.103. Accuracy of the classifier is 0.795, false positive rate is 0.100(400/2000), false negative rate is 0.105(105/499).

Feature 3: type is twoHorizontal, position is (10, 8), width is 8, height is 4, threshold is -236.000, polarity is -1, alpha is 3.343. Accuracy of the classifier is 0.208, false positive rate is 0.400(1600/2000), false negative rate is 0.392(391/499).

Top feature is feature 3. The plot is shown in figure 2.

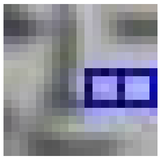

Figure 2

For 5 round,

Feature 4: type is twoVerticle, position is (15, 2), width is 4, height is 8, threshold is -276.000, polarity is 1, alpha is 3.871. Accuracy of the classifier is 0.819, false positive rate is 0.173(692/2000), false negative rate is 0.008(8/499).

Feature 5: type is twoVerticle, position is (6, 0), width is 8, height is 2, threshold is 143.000, polarity is -1, alpha is 3.999. Accuracy of the classifier is 0.722, false positive rate is 0.047(188/2000), false negative rate is 0.230(230/499).

Top feature is feature 5. The plot is shown in figure 3.

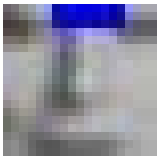

Figure 3

For 10 round,

Feature 6: type is twoVerticle, position is (14, 17), width is 4, height is 2, threshold is 1.000, polarity is -1, alpha is 4.510. Accuracy of the classifier is 0.735, false positive rate is 0.199(796/2000), false negative rate is 0.066(66/499).

Feature 7: type is twoHorizontal, position is (11, 9), width is 8, height is 1, threshold is -26.000, polarity is 1, alpha is 4.381. Accracy of the classifier is 0.755, false positive rate is 0.143(572/2000), false negative rate is 0.102(102/499).

Feature 8: type is twoHorizontal, position is (11, 9), width is 8, height is 1, threshold is -43.000, polarity is -1, alpha is 4.348. Accracy of the classifier is 0.265, false positive rate is 0.378(1512/2000), false negative rate is 0.358(357/499).

Feature 9: type is four, position is (11, 11), width is 4, height is 2, threshold is -14.000, polarity is 1, alpha is 5.021. Accuracy of the classifier is 0.760, false positive rate is 0.110(440/2000), false negative rate is 0.130(130/499).

Feature 10: type is twoVerticle, position is (6, 15), width is 1, height is 4, threshold is -4.000, polarity is 1, alpha is 5.182. Accuracy of the classifier is 0.778, false positive rate is 0.184(736/2000), false negative rate is 0.038(38/499).

Top feature is feature 10. The plot is shown in figure 4.

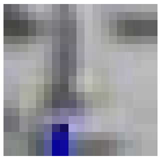

Figure 4

The change of accuracy, false positive rate and false negative rate on training data is shown in form 1 and figure 5.

Form 1

round	1	2	3	4	5	6	7	8	9	10
accuracy	88.2	79.5	88.2	81.9	91.5	87.2	92.1	86.9	93.2	89.8
FPR	7.9	10	7.9	17.3	5.5	9.9	7	10.8	5.9	9
FNR	3.9	10.5	3.9	0.8	3	2.9	0.9	2.3	0.9	1.2
FP(/2000)	316	400	316	692	220	396	280	432	236	360
FN(/499)	39	105	39	8	30	29	9	23	9	12

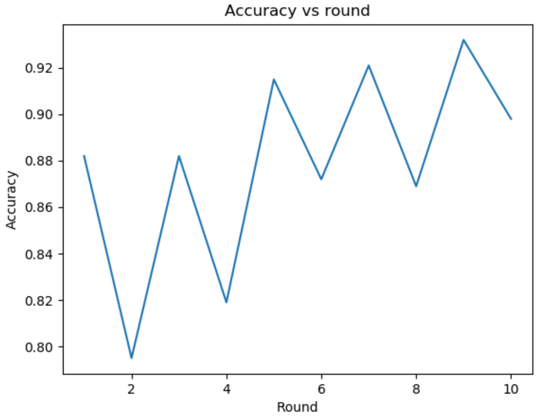

Figure 5

From figure 5 we can see that the accuracy on the testing set shows an increasing trend as the round grows. The up and down between each round is because the weight changes too much in the updating process each round.

The change of accuracy, false positive rate and false negative rate on testing data is shown in form 2 and figure 6.

Form 2

round	1	2	3	4	5	6	7	8	9	10
accuracy	58.4	58.7	58.4	79.9	60.6	57.2	59.5	58.6	57.7	59.5
FPR	4.5	9.4	4.6	12.1	1.3	6.1	4.5	7.9	2	3.9
FNR	37.1	31.9	37	8.1	38	36.7	36	33.5	40.3	36.5
FP(/2001)	180	376	184	484	52	244	180	316	80	156
FN(/472)	350	301	349	76	359	346	340	316	380	345

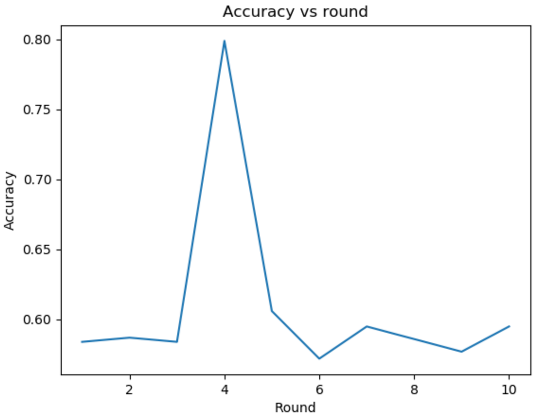

Figure 6

From figure 6 we can see that the features captured in the detector do not generalize very well as the accuracy is wandering near 60% for most of the time. From form 2 we can also see that the detector tends to label images with face as negative.

3 Threshold adjustments

In order to use different criteria to train the detector, I introduce a parameter, λ, into my algorithm. The selection of threshold, polarity for each feature and the selection of features for each round is based on which candidate has the lowest L =λ*FP+(1-λ)*FN. When λ=0.5, L equals half the loss of the candidate. When we want to decrease the FP in the loss, we just need to let λ<0.5, and when we want to decrease FN in the loss, we just need to let λ>0.5.

When I use λ=0.9 to run 5 round of adaboost: For training set, total accuracy is 0.770, false positive rate is 0.021, false negative rate is 0.209. For testing set, total accuracy is 0.510, false positive rate is 0.002, false negative rate is 0.488. Form 4 shows the changes in accuracy on testing set when the criteria changes.

When I use λ=0.1 to run 5 round of adaboost: For training set, total accuracy is 0.804, false positive rate is 0.195, false negative rate is 0.001. For testing set, total accuracy is 0.796, false positive rate is 0.136, false negative rate is 0.068. Form 3 shows the changes in accuracy on training set when the criteria changes.

Form 3

Criterion	Total accuracy	False positive	False negative(/499)
Empirical error	91.5	5.5(220/2000)	3(30/499)
False positive	77	2.1(84/2000)	20.9(209/499)
False negative	80.4	19.5(780/2000)	0.1(1/499)

Form 4

Criterion	Total accuracy	False positive	False negative
Empirical error	60.6	1.3(52/2001)	38(359/472)
False positive	51	0.2(8/2001)	48.8(461/472)
False negative	79.6	13.6(544/2001)	6.8(64/472)

4 Build the cascade system

The adaboost algorithm is written in face_detection_algorithm.py and the cascade algorithm is written in cascade.py. My cascade system consists of four stages. The first stage consists of one simple classifier, second stage has five, third stage has ten, and fourth stage has twenty simple classifiers. After each stage, the images labeling as negative will be abandoned and the positive image will go to next stage. On the last stage, those images that were classified as positive will be labeled as positive by the entire cascade system, and all other images will be classified as negative. The training process of each stage of cascade system is exactly the same as adaboost algorithm described above and the λ is set to be 0.1 to lower down the FN rate as we do not want to misclassify any images with faces. Because each stage eliminate a large portion of negative images, the training process became much faster than the original one-stage adaboost algorithm. Also, the result on the training set got better by constantly deleting images that have a high possibility to be non-faces images. The result on the testing set is just slightly better than the original one-stage adaboost because the strong classifiers at each stage do not generalize very well and lots of images with faces are deleted, too. The result is shown below.

For training set,

Stage 1: Delete 1 faces image, and 723 non-faces image

Stage 2: Delete 4 faces image, and 482 non-faces image

Stage 3: Delete 8 faces image, and 319 non-faces image

Total accuracy is 0.921, false positive rate is 0.066(264/2000), false negative rate is 0.013(13/499).

For testing set,

Stage 1: Delete 48 faces image, and 831 non-faces image

Stage 2: Delete 101 faces image, and 415 non-faces image

Stage 3: Delete 75 faces image, and 360 non-faces image

Total accuracy is 0.665, false positive rate is 0.084(336/2001), false negative rate is 0.251(237/472).