Lab 3 - Boosting
vicli815
jonto406

Lab 3 - Boosting

Q1

Figure 1 represents the accuracy of training data and test data, with respect to the number of weak classifiers being used. When running our algorithm which gives the plot of figure 1, we used the following parameters:
Number of training data: 1000 (split even between non-faces and faces)
Number of test data: 11788 (split even between non-faces and faces)
Number of Haar-Features: 100
Number or weak classifiers: 50

We have now revised our code to also take into account for alpha, see line 115, 133-145 and 164-175 in our AdaBoost script. Before we used a mode function, but now we weight the importance of each week classifier with alpha when building our strong classifiers.

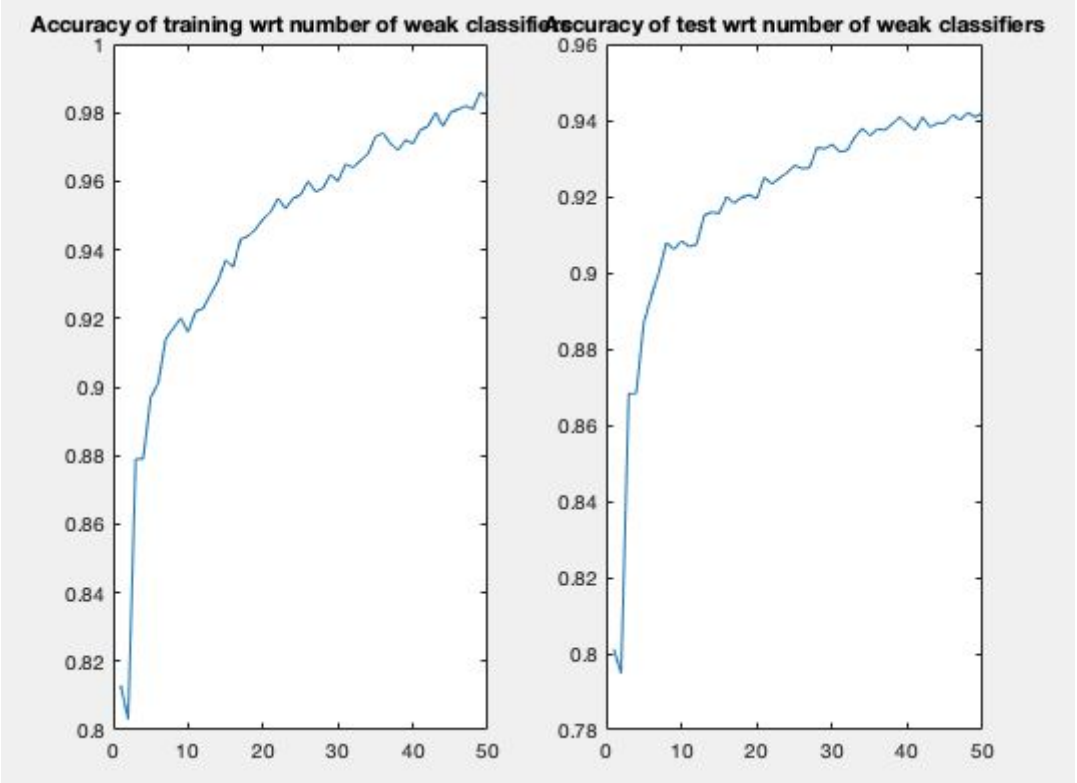

Figure 1: Accuracy of classification wrt number of classifiers

New figures.

Lab 3 - Boosting
vicli815
jonto406

Q2

We started with 30 weak classifiers, 500 training images and 25 Haar-Features. That resulted in a test accuracy of 89% and a train accuracy of 97%, as visible in figure 2. We then started to raise the number of training images and Haar-Features, giving the algorithm a better chance of finding the best features. By raising the number of Haar-Features, the accuracy rose to around 92.5%. This means that we are already close to the desired accuracy of 93 % without raising the number of classifiers. We then proceeded by slowly raising the number of classifiers by 2, and broke 93 % accuracy around 45 classifiers. Since the features are being selected randomly, the accuracy is somewhat changing by each evaluation of the algorithm, so we settled with 50 numbers of classiers to have a little margin above 93 % accuracy. See figure 1 (for our final classifier).

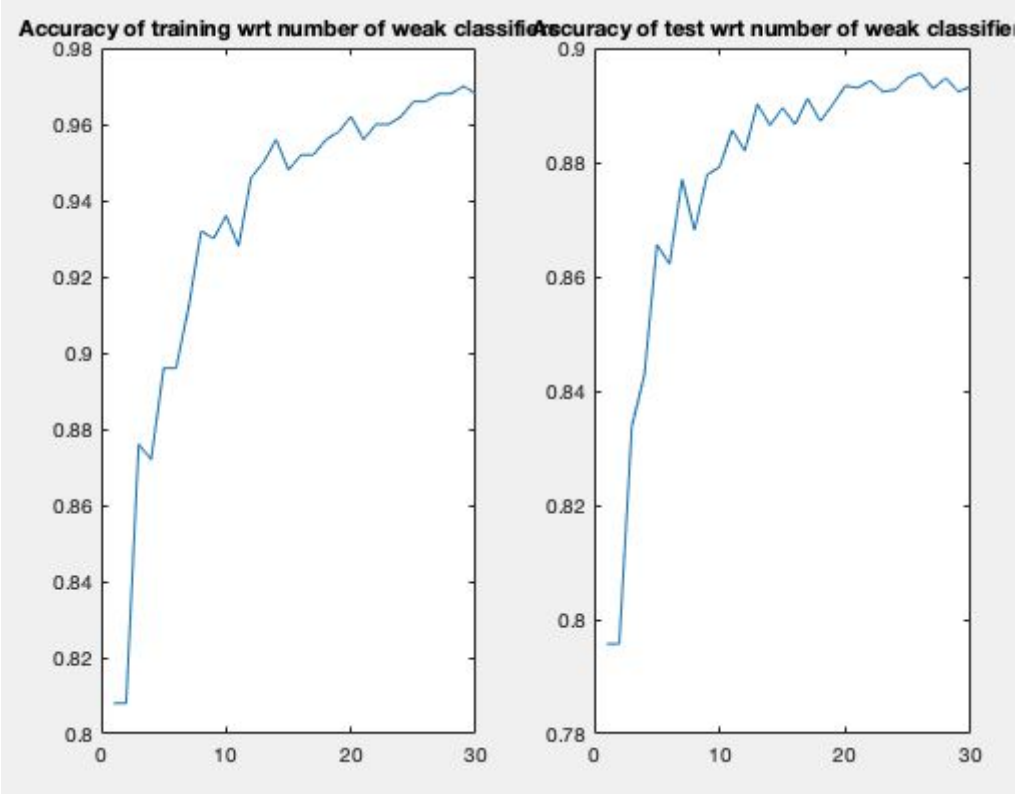

Figure 2: Accuracy when using 25 Haar-Features, 500 training images and 30 weak classifiers. (New figure 2).

Lab 3 - Boosting
vicli815
jonto406

Q3

The final accuracy achieved is 99.0% for training data and 93.4% for test data (See figure 3)
.

We chose to focus on finding the best Haar-Features, rather than increasing the number of weak classifiers. We consider the algorithm to be better when achieving a high accuracy without using too many weak classifiers. By finding the best features to be used for classification rather than using a lot of classifiers, the algorithm will have a better performance with respect to runtime once training is done. This is because we can use fewer classifiers when classifying the test data. Remember that the Haar-Features are being generated randomly, so their performance will vary slightly with each run.

By increasing training data we will give the algorithm more data to find the optimal weak classifiers. As long as the training data is considered as “good”, there is no problem of adding test data. If the test data being added is an outlier it might interfere with the algorithm, since incorrectly classified data (such as outliers) will gain more weight and have more influence over the classifier during training.

```
%Test evaluation
for c = 1:nbrWeakClassifiers
   Test_eval(c,:) = WeakClassifier((adaVar(c,1)),(adaVar(c,2)),xTest(adaVar(c,3),:));
   for i = 1:nbrTestImages

     %  OLD CODE, cant use mode
     %Test_classification(i) = mode(Test_eval(:,i));
     %NEW code
     for a=1:c
       aW(a)= Alpha(a)*Test_eval(a,i);
       end
       weigthed_decision= sum(aW);
       aW(:)=0;

       if  weigthed_decision>= 0
           Test_classification(i)=1;
       else
             Test_classification(i)=-1;
       end
       %%%--------------------------------

    Test_result(i) = Test_classification(i) ~= yTest(i);
   end
   Test_acc(c) = 1 - sum(Test_result)/nbrTestImages;
end
Final_test_acc = Test_acc(nbrWeakClassifiers);
```

```
Final_train_acc =

    0.9900


Final_test_acc =

    0.9337
```

Figure 3: Code that calculates the strong classifier’s accuracy for training and test data.

Lab 3 - Boosting
vicli815
jonto406

Q4

In figure 4, 20 of our chosen Haar-Features are plotted. While it is not obvious why all of these are chosen, we can see why some of them are good features to indicate if the image is of a face or not. Feature (2, 2) measures the vertical area of the middle of the face, where the mouth and nose often is lighter than eyes and cheeks, when an image is taken straight in front of a face. Feature (3,3) measures the area horizontal area of the middle of the face, where the eyes often are darker than the bottom of nose and top of mouth. Other features are harder for us to understand their usefulness, such as feature (4, 2).

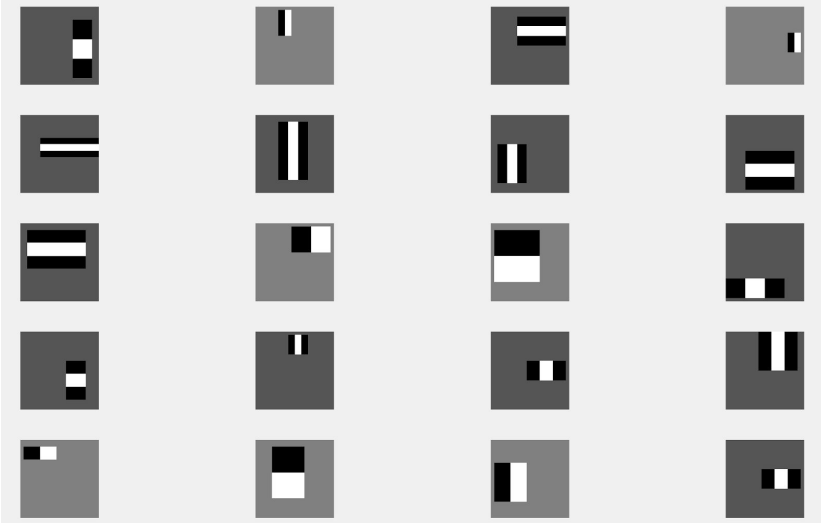

Figure 4: 20 of our chosen Haar-Features.

Q5

In figure 5, misclassified faces and non-faces are plotted. Some reasons why they might be misclassified are weird angles, the person is wearing glasses and that the picture is very dark or very light. There are also children or elderly people, which might be a reason depending on the training data. The misclassified non-faces are harder to find reasons for the misclassification. Most likely it is because the Haar-Features are measuring areas of the non-faces where the light/dark pixels correspond to the features. To further optimize our classifiers we would need to find even better Haar-Features, and perhaps raise the amount of weak classifiers as well.

Lab 3 - Boosting
vicli815
jonto406

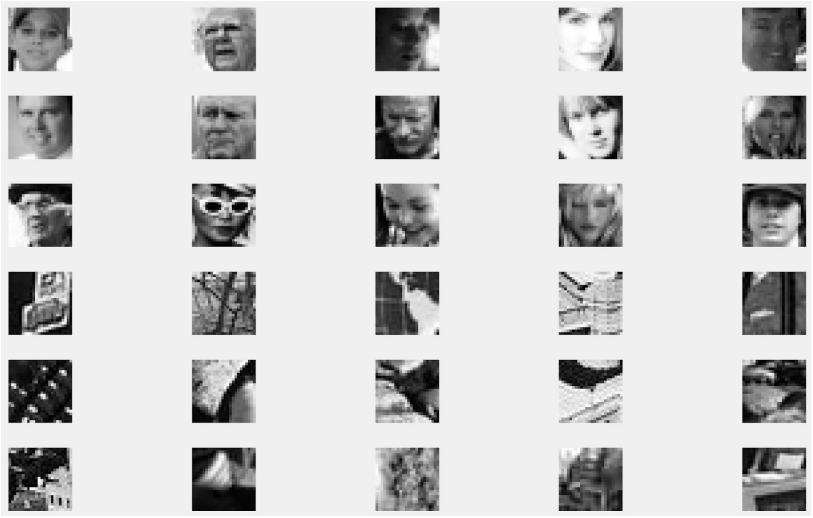

Figure 5: Incorrectly classified faces (first three rows) and non-faces (last three rows).

Q6

Yes, the result is reasonable. When using one weak classifier, the algorithm would have an accuracy of at least 50 % (A little bit better than guessing). From this point, adding classifiers will gradually improve the algorithm. As stated earlier, by finding good Haar-Features, each classifier will have less error. Thus, given the hyper-parameters we are using, the result is reasonable. The algorithm manages to classify more than 93 % correct, but since we are trying to focus on performance as long as the accuracy is above 93 %, there are not enough hyper-parameters, there will always be some misclassifications.

To improve the accuracy we can raise the amount of weak classifiers and Haar-Features, as seen in figure 6. The final training accuracy is 99.9 % and the test accuracy is 93.7 %.

Another way of improving is to analyze the training and test data, to make sure that we are using good training data, given what we want to classify.

Lab 3 - Boosting
vicli815
jonto406

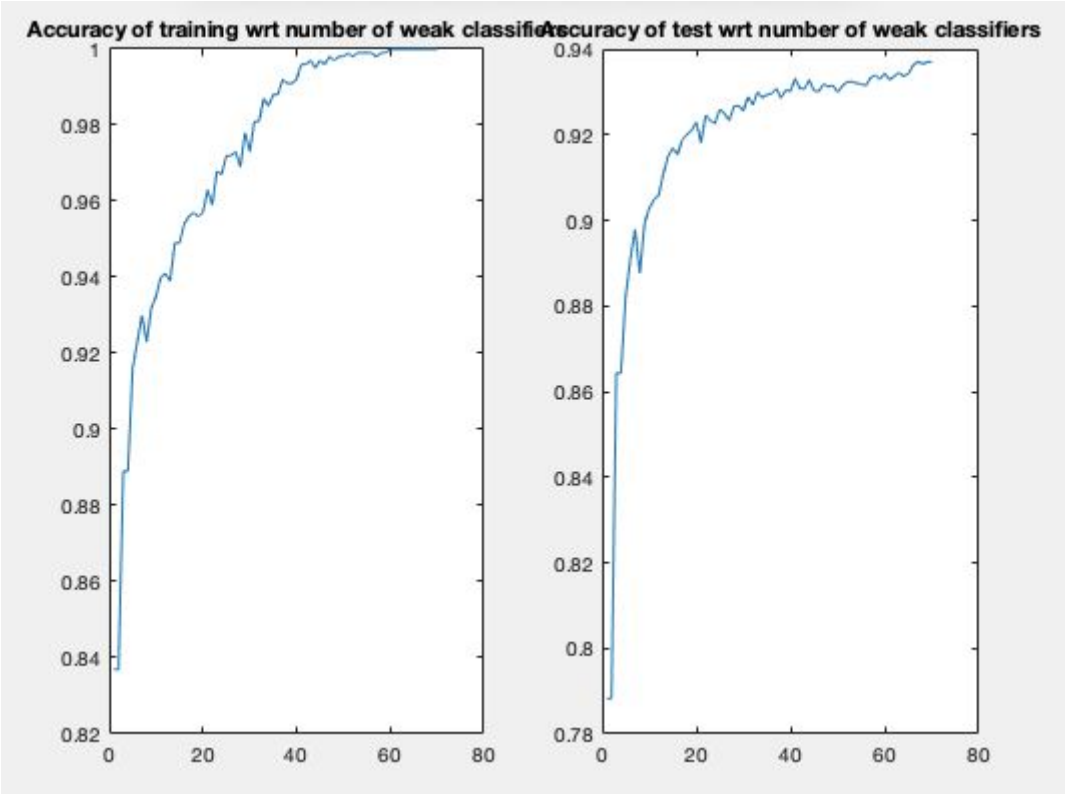

Figure 6: Training and test accuracy, using 130 Haar-Features, 1000 training images and 70 weak classifiers.

Q7

No, we cannot expect perfect results. Since we are using a brute force algorithm to find optimal weak classifiers, we are using data points from the training data as thresholds. Thus, we are finding weak classifiers that are exactly where the chosen, most important training data for the classifier is located. If we assume that the test data is very similar but not exactly the same, it is very likely to be some data that is just on the wrong side of our weak classifiers and thus being incorrectly classified.