Rain Lawson, SJ Stroud, Haley Benson
CS 4379C
Dr. Metsis

Final Project Report

Implementation Choices:

For training non-faces, windows were taken in order to “bridge the gap” in numbers of positive examples versus negative examples. Taking windows enabled us to have an equivalent number of training faces and non-face samples. We read in all training data as grey images, and the non-face image windows were taken in equal quantity to training faces. We had no memory or problematic speed issues with the training set like this.

For testing, we decided to not use the testing face photos. We did this project under serious time constraints, and, therefore, we were unable to crop or identify face windows in all of the images. Additionally, there were some issues with sending large numbers of files/file sizes that also stunted any effort in this field. An individual sample from the testing face photos was used to test the run time of skin detection versus regular analysis. For the skin detection portion of the assignment, we ran into problems with trying to create large arrays of large RGB files, as well as problems with the identification of face subwindows and problems with reading in the right number of files. We ended up only reading in one image file at a time to deal with the memory use issue we were having, but this required that we specify the names of individual files, so we mostly used the testing non-face images for testing skin detection (with only one exception).

Parameters:

Our hard-coded parameters for the train.m file include the size of the training data array, the number of training samples sent to the AdaBoost function, the number of rounds given to the AdaBoost function, and the number of rounds of bootstrapping that we did. We decided to have an equal number of training faces and non-faces, so the size of the training data array was hard coded to be the number of faces times two windows of 100 by 100 pixels (the size of the training faces). We chose 600 samples to send to AdaBoost for each training round because it takes about fifteen seconds to run each round of AdaBoost, which is a relatively quick time to get results back, and we wanted to have the results have plenty of error to leave room for improvement via other methods. This is also why we chose to have AdaBoost run nine rounds in the first session, in order to keep it brief and relatively inaccurate. We chose to have three rounds of bootstrapping because it takes about 30 seconds to run the three rounds, which gives good results while keeping the run time short enough to run the training file frequently for testing and experimentation. For the AdaBoost rounds parameter on the later bootstrapping layers, we gradually increased the number of rounds in order to get more training on the more difficult training sets. We also chose 18 and 24 because it kept the training run time about 30 seconds, which was about as long as we could stand to have it, since we were running tests quite frequently and were doing this project under serious time constraints.

For the parameters of the test.m file, the number of tests was hard coded and the skin detection threshold was set. The number of tests was set to have equivalent numbers of face and non-face windows to get a more even and fair distribution of tests. It also allowed the non-face windows to be set to 100 by 100, the same size as the face windows, again making the tests more fair. Finally, the skin probability threshold from the return of the detect_skin function was set to be .25 because we wanted to minimize noise in the skin frame and it seemed to return good results while reducing the most noise.

Results:

The base results using only nine rounds of AdaBoost were fairly good (see below). It had an average false positive rate of only 1.84% and an average false negative rate of 2.6%. Training time was short, just under 15 seconds on average, with full set testing time being about 0.207 seconds. These results are not the best, but they are a good start and leave plenty of room for improvement.

trial #	F Pos Rate	F Neg Rate	Pos Success Rate	Neg Success Rate
1	1.30%	2.47%	97.53%	98.70%
2	0.78%	3.38%	96.62%	99.22%
3	4.55%	2.99%	97.01%	95.45%
4	1.56%	1.95%	98.05%	98.44%
5	1.04%	2.21%	97.79%	98.96%
avg:	1.84%	2.60%	97.40%	98.16%

trial #	training runtime	single test runtim	full test runtime
1	15.165695	0.0000789	0.202186
2	14.445848	0.0000813	0.211039
3	14.808203	0.0001303	0.207407
avg	14.806582	9.68333E-05	0.206877333

Bootstrapping caused a significant improvement in accuracy at the cost of roughly three times the training time (see below). The average false positive rate was reduced to .68% and the average false negative rate was reduced to .75%. Positive and negative success rates were both above 99%. It is worth noting that the average training time was roughly 30 seconds, about double the training time of normal AdaBoost, but we believe that doubling the training time for roughly three times less inaccuracy is a good trade

trial #	F Pos Rate	F Neg Rate	Pos Success Rate	Neg Success Rate
1	1.04%	1.17%	98.83%	98.96%
2	0.78%	0.78%	99.22%	99.22%
3	0.65%	0.65%	99.35%	99.35%
4	0.52%	0.65%	99.35%	99.48%
5	0.39%	0.52%	99.48%	99.61%
avg:	0.68%	0.75%	99.25%	99.32%

Skin detection was largely accurate, and, on average, slightly faster than normal analysis (see below for time). Due to the unequal nature of our tests, we only compared single detection time where the environments for generic face detection and skin face detection were roughly equivalent. That being said, the average skin detection time was lower than the generic tests, but not by much. Additionally, the accuracy of the skin detection was only 97.3%. This is better than just using AdaBoost, however I believe there to be some problems with our implementation of skin detection, but we did not have the time to resolve any issues. Another problem is that the dataset for testing the skin detection was rather small, only 37 images were tested due to memory concerns encountered early in the development of the skin detection portion of the project, therefore if one test is failed the accuracy of the results drops 2.7%.

trial#	skin single detection time		generic single detection time	
1	1.8525		1.9747	
2	1.9225		1.9789	
3	1.932		1.8991	
avg	1.902333333		1.9509	