Implementation of Adaboost

Dr. Vangelis Metsis, Texas State University

Report by: Adam Cashel, Anabella Dominguez

Table Of Contents

Abstract

We present an efficient implementation of face detection using Adaboost. In particular utilizing the concepts of Skin detection, Bootstrapping, and Classifier Cascades.

Face Detection

Adaboost

Faces and nonfaces with their corresponding labels become the input data set for the Adaboost algorithm.The Adaboost algorithm is adaptive in that misclassified data is boosted during training and then it is assigned a higher weight in order to be classified correctly. The training images are used as the input data set and are assigned classification labels. Adaboost will call weak classifiers repeatedly over the training dataset. In the implementation we have used the Adaboost algorithm in order to find the error rate within 35 rounds of training.

Skin Detection

Not implemented.

Bootstrapping

Through the use of bootstrapping we grew and trained our dataset. First by only taking a small dataset of 100. And then we used the boosted_predict function to test the rest of the dataset for face and nonface pictures based on our strong classifier model. Given the parameters, a strong classifier, a test face or nonface image, list of weak classifiers, and a number that indicated how many weak classifiers are to be used for our model. After each call of the function if the training image was over the prediction threshold then it will be predicted as a face image. If the training image is under the prediction threshold then it will be considered a nonface image. The next step was to check whether or not the training image was correctly predicted. In the case that the image was identified wrong it would be added to the training list face or nonface matrix. Otherwise if it is identified correctly the training image is not added to the training matrix.

Classifier Cascades

For our classifier cascades algorithm we made a function called cascade_classify which takes in an image, strong classifier, and the list of weak classifiers. Inside of this function we go through each subwindow size box of 35x35. Starting with our strong classifier we gave it just a few weak classifiers that scan each subwindow and call the boosted_predict function. Then get the highest prediction value out of all of the subwindows and compare it to our threshold to label whether or not it crosses it for it to be considered a face image. If it does not cross the threshold to be considered a face picture then the function returns a 0 and does not go further with more weak classifiers. If the highest prediction does cross the threshold then it will go to the next round where another 4 weak classifiers will be added to the strong classifier. Our classifier cascades algorithm has 60 total rounds of testing. If a given image passes the last threshold then it will be labeled as a face picture.

Approach

Order of Implementation

Within further implementation, we mapped out all of the required functions. As a result we decided to implement the more basic functions first, including the adaboost algorithm, and only then we build the bootstrapping and cascades algorithms.

Bootstrapping

We chose our image training samples and implemented the boosted_predict function to the dataset. With each call of the function the training images would be put above a threshold to predict a face, otherwise if below it is not a face. Once that is done, mistakes will be identified, if any mistake is detected then it would be added to the face or nonface training set.

Cascades

The Goal of Constructing a cascade of classifiers is to achieve increased detection performance. The key insight is that boosted classifiers can be constructed, which reject many of the negative sub-windows while detecting almost all positive instances. Simpler classifiers are used to reject the majority of sub-windows before more complex classifiers are called upon to achieve low false positive rates

Thresholding

Our Methodology for deciding on a particular threshold was to do multiple rounds of testing balancing out the face & nonface accuracies for each threshold. We favored the face accuracy over nonface accuracy leading to more false positives for each layer of the face detector.

For the adaboost section we choose to implement our strong classifier using 35 weak classifiers. We decided on the number 35 due to the fact that multiple rounds of testing showed that the accuracy was not going up as much as the run time increased. The adaboost had 3 parameters, The labels of 1 or -1 for face or nonface, responses, and 35 rounds of training. We chose 35 rounds of training due to that we found that after 35 there was diminished returns to training more.

The classifier cascades we choose to select a threshold of 3 due to that most non face images we tested had a prediction of around -22.8. This gave us a little bit lower face detection accuracy but a very high nonface accuracy. With the classifier cascades we used three parameters in the cascade function. The first parameter is the image, second is the strong classifier, and the third is the weak classifier list.

The bootstrapping strategy we choose a threshold of 3 to determine whether an image is a face or nonface. We chose this number to follow our methodology in that we wanted the highest nonface accuracy even though our face accuracy might be slightly lower because of that.

Results

Adaboost

The accuracy of testing through the whole dataset trained on the first 100 images gave us an accuracy of 87.0172% overall when combining face and nonface training images.

Skin Detection

Not implemented.

Bootstrapping

The accuracy for bootstrapping for detecting faces in the training set is 99.9710% with a false negative rate of 0.0290. For nonfaces in the training set the accuracy is 72.0917% with a false positive rate of 27.9083%

Classifier Cascades

Through our implementation of classifier cascades we achieved a 99.0% accuracy for nonfaces on the training set. And an accuracy of 76.50% for faces on the training set.

Analysis

Testing our final face detector over the 3 different test folders our face detector achieved a 98.3030% overall accuracy including both nonface and face images.