Pascal Gautam: p_g224
Xuan T le: xtl1
CS4379C- Intro to Computer Vision
Dr. Vangelis Metsis
Fall 2020

Advanced face detection with Adaboost, Skin detection, Bootstrapping, and Cascades

The main goal of computer vision is to create an algorithm that can intelligently describe images and videos. There are many applications of computer vision out of face detection is the most popular. Face detection is a computer technology that is used to find and identify human faces in digital images. For this project, we did face detection with Adaboost, cascades, bootstrapping, and skin detection.

At first, we implemented the Adaboost algorithm in our dataset. Different faces and nonfaces were used in our dataset with altogether 3047 faces and 130 nonfaces for training our model. We cropped nonfaces images into We cropped given training faces into 50 by 50 so that we eliminate other parts and only necessary face features. We created a script called create_training_data.m which loads all the training images and we saved different variables like faces, non_faces, faces_integrals, non_faces_integrals, face_size, face_horizontal, and face_vertical inside training_data.mat which we will later load into the train.m to perform training. We first choose 1000 random weak classifiers and saved weak_classifiers into classifiers1000. Then we precomputed responses on all of our training examples on all weak classifiers. We then trained Adaboost on 15 rounds and calculated classification accuracy (figure below). It ranged from 78% to 99%. Then we performed bootstrapping on all our datasets and calculated the accuracy of the whole training data. In the beginning, we performed our training on all 3047 faces and only 130 non faces images because of which our accuracy while testing performed poorly. Later we cropped that nonface image into different small sizes and our testing accuracy performed better (we can see the difference in pictures below of Michael Jordan). While testing we experimented with different scales and different result numbers. Different pictures of our project are presented below.

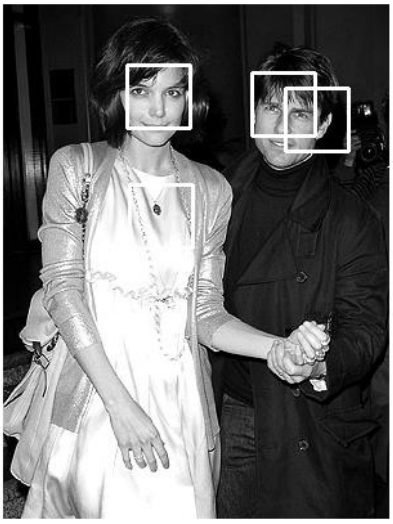
While Adaboost is a good method on its own but it was not able to give high accuracy results. So, we used bootstrapping to improve the accuracy and stability of our model. To perform bootstrapping, we ran our classifier on all training

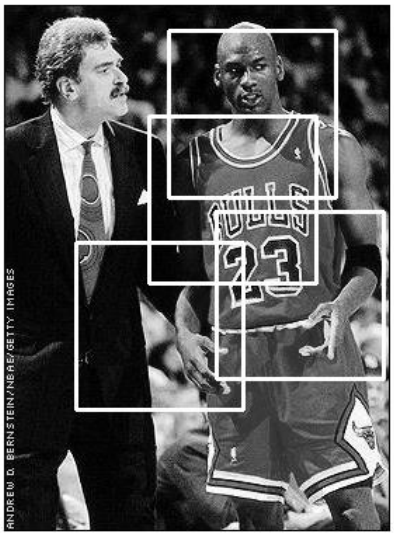
images and identified correctly predicted and misclassified images. Because of this accuracy of our model increased.

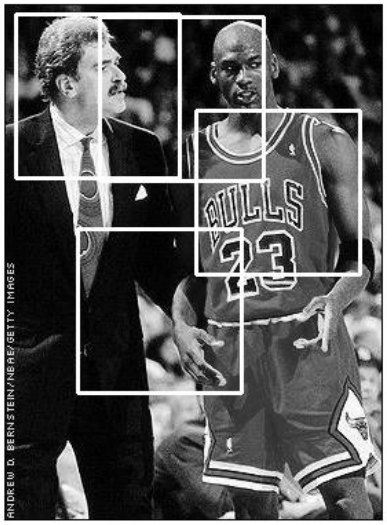
After training our model, we began testing it. We first tried testing without classifier cascade and skin detection on cropped faces. Our model did not perform well. Then we applied skin detection in color photos that were provided, and we got some good accuracy. Around 13 pictures out of 20 were classified fully correctly. Some were partially correct. Since we had histograms of positive and negative skin colors. It made it fast and accurate. Last was cascade classifier, unfortunately, we were not able to perform this on our model.

At last, this project was a good opportunity to demonstrate what we have learned throughout our class. We applied different algorithms to perform face detection. And computer vision is a very popular field right now and it has seen quite a bit of advancement in recent years, but still is far behind to accomplish its main goal. So, we would like to further our knowledge in this field, do some more research and expand our knowledge.

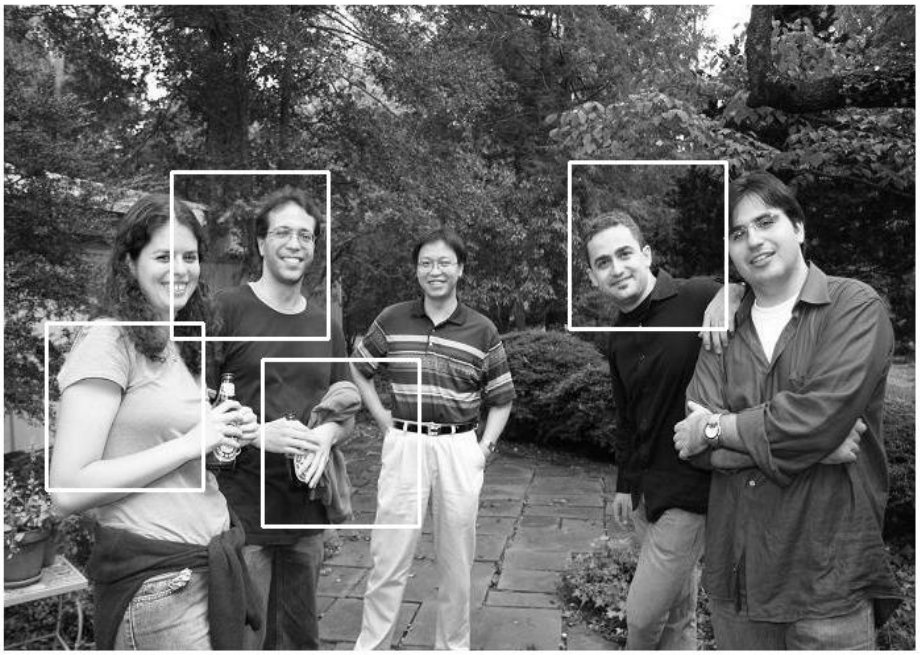
Some results




ANDREW D. BERNSTEIN/NBAE/GETTY IMAGES


ANDREW D. BERNSTEIN/NBAE/GETTY IMAGES


Apéritifs à marques
PRINCETON



Current Folder

Name

- data
- files
- boosted15.mat
- classifiers1000.mat
- codeCreateTraining_dataFile.m
- directories.m
- reponsesResults.mat
- test.m
- train.asv
- train.m
- training_data.mat

test.m (Script)

Advanced face detection with AdaBoost

Live Editor - train.mlx *

codeCreateTraining_dataFile.m | directories.m | train.m | test.m | train.mlx * | eval_weak_classifier.m | test.mlx *

```
response = 982

% Train on all training data - run 15 rounds of adaBoost
misclassified = 0;
tic
for i = 1:15
    order = boosted_classifier(i, 1);
    best_classifier = weak_classifiers{order};
    best_threshold = boosted_classifier(i, 3);
    for a = 1:example_number
        response = eval_weak_classifier(best_classifier, examples(:,:,a));
        if a <= size(faces, 3)
            if response <= best_threshold
                misclassified = misclassified + 1;
            end
        end
    end
    classification_accuracy = ((example_number - misclassified)/example_number)*100
    misclassified = 0;
end

classification_accuracy = 80.0733
classification_accuracy = 78.6653
classification_accuracy = 91.5667
classification_accuracy = 97.7731
classification_accuracy = 81.3519
classification_accuracy = 99.5905
classification_accuracy = 84.1822
classification_accuracy = 99.8204
classification_accuracy = 82.3217
classification_accuracy = 82.7168
classification_accuracy = 98.3406
classification_accuracy = 99.0015
classification_accuracy = 78.3062
classification_accuracy = 93.7576
classification_accuracy = 99.8707

toc
```

Workspace

Name	Value	Class
a	7252	doubl
acc	99.9080	doubl
best_classifier	1x9 cell	cell
best_threshold	121.2400	doubl
boosted_classifier	20x3 double	doubl
bootstrapDataSet	50x50x5449 dou...	doubl
classification_accuracy	99.8707	doubl
classification_accuracyBT	15x1 double	doubl
classifier	1x9 cell	cell
classifier_number	1000	doubl
code_directory	'C:\Users\gauta\...	char
count	5449	doubl
count2	7252	doubl
data_directory	'C:\Users\gauta\...	char
example	13921	doubl
example_number	13921	doubl
examples	50x50x13921 do...	doubl
face_horizontal	50	doubl
face_size	[50,50]	doubl
face_vertical	50	doubl
faces	50x50x3046 dou...	doubl
faces_integrals	50x50x3046 dou...	doubl
feature	1000	doubl
i	15	doubl
index	3041	doubl
integral	50x50 double	doubl
labels	13921x1 double	doubl
max	99.9080	doubl
misclassified	0	doubl
misclassified2	0	doubl
misclassified3	110	doubl
misl	0	doubl
non_faces	50x50x10875 do...	doubl
non_faces_integrals	50x50x10875 do...	doubl
num_faces	0	doubl
num_non_faces	0	doubl
number	5437	doubl
order	646	doubl
response	-2041	doubl
responses	1000x13921 dou...	doubl

```
        bootstrapDataSet(:,:,count+1) = examples(:,:,a);
        count = count + 1;

    end

end
```

```
acc = 82.8950
acc = 81.8834
acc = 93.0844
acc = 97.4986
acc = 84.3848
acc = 99.6138
acc = 86.5551
acc = 99.8713
acc = 85.5067
acc = 85.1573
acc = 98.9332
acc = 99.2091
acc = 81.5155
acc = 94.3351
acc = 99.9080
acc = 82.8950
acc = 81.8834
acc = 93.0844
acc = 97.4986
acc = 84.3848
acc = 99.6138
acc = 86.5551
acc = 99.8713
acc = 85.5067
acc = 85.1573
acc = 98.9332
```

boosted_classifier	*20x3 double*	double
bootstrapDataSet	*50x50x5449 dou...*	double
classification_accuracy	99.8707	double
classification_accuracyBT	*15x1 double*	double
classifier	*1x9 cell*	cell
classifier_number	1000	double
code_directory	'C:\Users\gauta\...	char
count	5449	double
count2	7252	double
data_directory	'C:\Users\gauta\...	char
example	13921	double
example_number	13921	double
examples	*50x50x13921 do...*	double
face_horizontal	50	double
face_size	[50,50]	double
face_vertical	50	double
faces	*50x50x3046 dou...*	double
faces_integrals	*50x50x3046 dou...*	double
feature	1000	double
i	15	double
index	3041	double
integral	*50x50 double*	double
labels	*13921x1 double*	double
max	99.9080	double
misclassified	0	double
misclassified2	0	double
misclassified3	110	double
misl	0	double
non_faces	*50x50x10875 do...*	double
non_faces_integrals	*50x50x10875 do...*	double
num_faces	0	double
num_non_faces	0	double
number	5437	double
order	646	double
response	-2041	double
responses	*1000x13921 dou...*	double