

CIS 6930 (Biometrics) Project - 1

Crowd Faces Detection - Readme

Ashutosh Modi (modia), Jeetendra Ahuja (jahuja) - MSCS

March 7, 2017

1 Following names are the code files for project-1.

1. hog.cpp
2. haar.cpp

2 Windows Enviornment

To develop the algorithm following system configuration/enviornment has been used:

- Microsoft Visual Studio Enterprise 2015
- Compiler VC14
- OpenCV 3.1

2.1 Create VS Project

Need to create a Visual Studio console project with standard process for CPP enviornment. This enviornment and process is common for both algorithm implementations. Projects need to be created separately.

2.2 HOG specific enviornment

2.2.1 Train and Test data set path

Two folders with positive and negative training data needs to be placed under one directory. Folder names for positive and negative directories are unique and should not used other than as mentioned below respectively:

- pveimages
- nveimages

Parent directory containing these directories would be a console input.

2.2.2 Image naming convention while training

It is important to use the code specific naming conventions for training images. All training images used must be in “.png” format. Following conventions should be used for image names

- Positive dataset: All images must have numeric names starting from 1 i.e. “1.png”, “2.png”, ...
- Negative dataset: All images must have numeric names with the format as mentioned. “TrainNeg (1).png”, “TrainNeg (2).png”, “TrainNeg (1).png”,...

2.2.3 Test image configuration

Image which to be tested for face detection needs to be placed in the same directory where code file is present. Test image can have any naming convention.

2.3 Haar based environment

For code to run following XML file needs to be kept in the directory where binary resides.

- haarcascade_frontalface_alt.xml

3 Running project

3.1 Code change

3.1.1 HOG specific

As of now command line input accepted is only train directory name. Hence following small changes needs to be done before building and running the code.

1. Provide the number of positive and negative image count. Variables used are posCount and negCount as shown in [figure 1a](#).
2. Provide test image name for the variable as shown in the [figure 1b](#).

```
//Positive image count  
posCount = 99;  
//Negative image count  
negCount = 582;
```

(a) Image Count

```
//Test image name to enter  
Mat testImg = imread("Crowd8.jpg",0);
```

(b) Test Image

Figure 1: Code Changes

3.1.2 Haar specific

Test image needs to be updated in the code similar to shown in the [figure 1b on the preceding page](#).

3.2 Running Code

3.2.1 HOG

Run HOG project code with VS standard steps. Supply the directory path once prompted on console. After training the model detected faces will be shown on supplied test image in a window name "testImage".

3.2.2 Haar based

Run HOG project code with VS standard steps. Output image will get stored in the project directory with name "HaarDetected.png".

4 Attachments

1. hog.cpp
2. haar.cpp
3. haarcascade_frontalface_alt.xml
4. hogSVMFaces.xml¹
5. Project1-Report.pdf

¹Trained on our image dataset