Face Detection in Crowd Images

Ashutosh Modi (modia), MSCS
Jeetendra Ahuja (jahuja), MSCS
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida 33620
Email: modia@mail.usf.edu; jahuja@mail.usf.edu

***Abstract*— Features from face can be obtained through various techniques such as pixel-based features [4], PCA-based features [5], Hog-based feature [1], [2], Eigen-based features, Haar-based features [3]. After extracting features, a classifier like cascade, SVM or other classification method can be used to train our model for detecting faces in an unknown crowd sample. Sometimes, we might need to detect crowd from an image and then detect faces from it [7]. This technique gives faster result by preventing classifier to look for faces in an extraneous area of an image. In this project, we worked on Haar feature-based cascade classifier and Histogram of Oriented Gradients (HoG) for face detection in crowd.**

Index Terms— Face detection, Histogram of Oriented Gradients, Haar features, Cascade classifier, Support Vector Machine.

1. Haar feature-based cascade classifier

This is known as Viola-Jones algorithm which is widely used for face detection. It comprises of following four steps:

1.1. Haar Feature Selection

It is based on some common properties for all human faces. A few common properties can be nose bridge region is brighter than the eyes, the eye region is darker than upper-cheeks, etc.

1.2. Creating an Integral Image

The Integral Image is used as a quick and effective way of calculating the sum of values (pixel values) in a given image – or a rectangular subset of a grid (the given image). It can also, or is mainly, used for calculating the average intensity within a given image.

1.3. Adaboost Training

It is used to improve the performance of a classifier. It selects only those features that improve the predictive power of the model by reducing dimensionality and potentially reducing execution time as irrelevant features don't need to be computed.

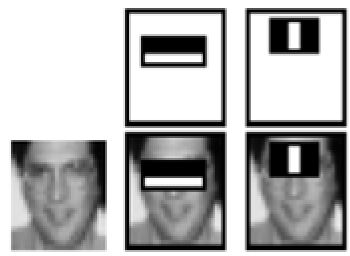

Figure 1. Haar feature-based

1.4. Cascading Classifiers

It is concatenation of several classifiers. Each stage consists of a strong classifier. 1st stage classifier may have just 2 features which is used to filter out most negative windows of image then 2nd stage classifier may have 10 features to tackle harder negative windows which were not detected by 1st stage classifier and it keep on increasing until it converges.

2. HOG descriptors

Image gradient is a directional change in the color of an image. Histogram of oriented gradients is a feature descriptor for object detection. Its base concept is that shape within an image can be described by distribution of intensity gradients or edge directions. The image is divided into small regions called cells and for pixels within each cell, a histogram of gradient directions is created.

Matrix of HoG descriptors is created where each row represents descriptors of an individual image (either positive or negative). Each row is appended with a label which is either 1 for positive sample or 0 for negative sample. HoG descriptor matrix is supplied as an input to SVM to train our model which finally detects a face in unknown crowd image.

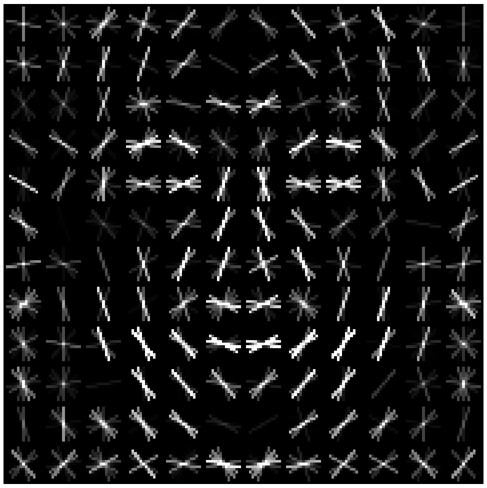

Figure 2. Histogram of Oriented Gradients

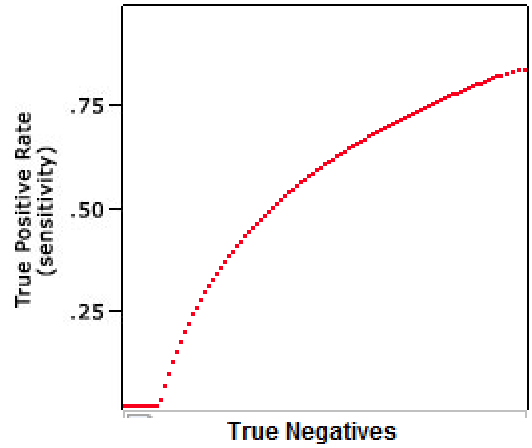

Figure 3. ROC

3. Dataset

3.1. Training

From given dataset for crowd images, we have used 83% for training the SVM for HoG. We used 99 positive cropped faces and 582 negative images for training for training Hog on SVM. Positive images have been extracted from the image dataset shared. Whereas, some negative images are cropped from crowd sample set and some are taken from abstract images.

3.2. Testing

From given dataset for crowd images, we have used 17% for testing.

4. Result

Figure 3 shows ROC curve for table1.

Scaling	Threshold	FP	TP	Image
1.1	1	1	0	1.png
1.2	1	12	5	1.png
1.2	1	7	55	1.png
1.15	1	1	0	1.png
1.3	1	3	14	1.png
1.18	1	1	0	1.png

TABLE 1. RESULT

5. Future Prospects

We have found that pretrained Haar performance is very good as compared to the HoG implemented. Poor performance may be due to:

- Small dataset
- Poor image quality
- Not enough training done on SVM

HoG performance can be enhanced by improving on these aspects. Also as discussed in this paper detecting crowd scene and then applying HoG on that area only would result in increased processing speed. Also GPU can be used to for training and testing to increase the parameter tuning speed.

References

[1] Dalal, Navneet, and Bill Triggs. *Histograms of oriented gradients for human detection.* Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.

[2] Dalal, Navneet, Bill Triggs, and Cordelia Schmid. *Human detection using oriented histograms of flow and appearance.* European conference on computer vision. Springer Berlin Heidelberg, 2006.

[3] Viola, Paul, and Michael Jones. *Rapid object detection using a boosted cascade of simple features.* Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001.

[4] Kienzle, Wolf, et al. *Face detection-efficient and rank deficient.* NIPS. Vol. 17. No. 2005. 2004.

[5] Menser, Bernd, and Frank Muller. Face detection in color images using principal components analysis. IEE conference publication. Vol. 2. No. 465. 1999.

[6] Yang, Ming-Hsuan, David J. Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. IEEE Transactions on pattern analysis and machine intelligence 24.1 (2002): 34-58.

[7] Arandjelovic, Ognjen. Crowd detection from still images. BMVC 2008: Proceedings of the British machine vision association conference 2008. BMVA Press, 2008.

[8] Barr, Jeremiah R., Kevin W. Bowyer, and Patrick J. Flynn. The effectiveness of face detection algorithms in unconstrained crowd scenes. Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on. IEEE, 2014.

[9] http://docs.opencv.org/3.0-beta/doc/tutorials/ml/ introduction_to_svm/introduction_to_svm.html

[10] HOG Image credit: http://blog.dlib.net201402dlib-186-released-make-your-own-object.html