Face Detection: Histogram of Oriented Gradients and Bag of Feature Method

L. R. Cerna, G. Cámara-Chávez, D. Menotti
Computer Science Department, Federal University of Ouro Preto
Ouro Preto, MG, Brazil

Abstract— *Face detection has been one of the most studied topics in computer vision literature; so many algorithms have been developed with different approaches to overcome some detection problems such as occlusion, illumination condition, scale, among others. Histograms of Oriented Gradients are an effective descriptor for object recognition and detection. These descriptors are powerful to detect faces with occlusions, pose and illumination changes because they are extracted in a regular grid. We calculate and vector quantizes into different codewords each descriptor and then we construct histograms of this codeword distribution that represent the face image. Finally, a set of experiments are presented to analyze the performance of this method.*

Keywords: Face Detection, Histogram of Oriented Gradients, descriptor, codeword, Bag of features.

1. Introduction

Actually, many applications and technologies inventions use computers because of their rapid increase of computational powers and the capability to interact with humans in a natural way, for example understanding what people says or reacting to them in a friendly manner, so through years they become more intelligent like humans. One technique that enable such natural human-computer interaction is face detection [17].

Face detection is a very important task to recognize a person by using a computer. Actually, many algorithms have been developed to make this detection task more easy but in real world scenario it is very difficult due to complex background, variations in scale, pose, color, illumination and among others. Because of its popularity many applications use it such as surveillance systems, digital camera, access control, human-computer interaction and so on.

How we can detect faces into a given arbitrary image? A possible solution is to segment this image into interest regions based on some homogeneity criterion, and then search and locate in all image regions where a face is. Methods in the literature have many restrictions because they do not vary pose and only work with frontal faces, constants lighting conditions, *etc*, (as seen in Figure 1) and when we evaluate with faces in real world scenarios their performance decrease and do not present good generalization and accuracy.

Fig. 1: Example of face images with huge variations in pose, facial expression, color, lighting conditions, *etc*.

There have been hundreds of face detection approaches in the literature. In order to simplify our study, we can group them into four categories: knowledge-based methods, feature invariant approaches, template matching methods and appearance-based methods [17].

The publication of Viola-Jones work increased the progress of the face detection area [15]. This framework presents problems when detects faces in complex backgrounds. Moreover, the processing time to extract and select features is very long due to the feature dimensions and the training time is very slow, demanding a great computer effort. On the other hand, the detection time is very fast since it uses a set of strong features selected.

Histograms of Oriented Gradient (HOG) are descriptors rotationally invariant which have been used in optimization problems as well as in computer vision [13], [6]. In our case, we apply in the face detection problem.

In this paper, we explore the representational power of HOG descriptors for face detection with Bag of features. We propose a simple but powerful approach to detect faces: (1) extract HOG descriptors using a regular grid, (2) vector quantization into different codewords each descriptor, (3) apply a support vector machine to learn a model for classifying an image as face or non-face based on codeword histograms.

The remainder of this paper is organized as follows. In Section 2, we present our proposed face detection method. Details of implementation are described in Section 3, and in Section 4, we present a set experiments. Finally the conclusion is presented in Section 5.

2. Face Detection Method

Histograms of Oriented Gradients are generally used in computer vision, pattern recognition and image processing to detect and recognize visual objects (i.e. faces). We propose to use HOG descriptors because we need a robust feature set to discriminate and find faces under difficult illumination backgrounds, wide range of poses, *etc*, by using feature sets that overcome the existing ones for face detection.

HOG is reminiscent of edge orientation histogram, SIFT descriptor and shape context. They are computed on a dense grid of cells that overlap local contrast histogram normalizations of image gradient orientations to improve the detector performance [5]. So that, this feature set performs very well for other shape based object classes (i.e. face detection) because of the distribution of local intensity gradients, even not precising any knowledge of the corresponding gradient [4].

To extract HOG descriptors, first count the occurrences of edge orientations in a local neighborhood of an image. This means the image is divided into small connected regions, called cells (*e.g.*, size 9) and the histogram of edge orientations is computed for each one. Depending on whether the gradient is unsigned or signed, the histogram channels are spread over $0^\circ - 180^\circ$ or $0^\circ - 360^\circ$.

To compensate the illumination, histogram counts are normalized by accumulating a measure of local histogram energy over the connected regions, then use the results obtained to normalize all cells in the block (*e.g.*, size 2) and finally, the combination of these histograms represents the HOG descriptor (see Figure 2).

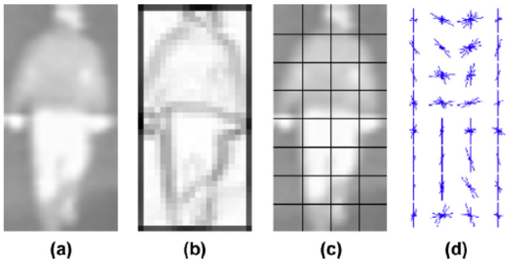

Fig. 2: Images from the various stages of generating a Histogram of Oriented Gradients feature vector. (a) Original pedestrian image, scaled to 20x40 pixels, (b) gradient image, (c) image divided into cells of 5x5 pixels, resulting in 4x8 cells, (d) resulting HOG descriptor for the image showing the gradient orientation histograms in each cell [11].

To make invariant the Hog descriptor in scale and rotation, extract descriptors from salient points by using a rotation normalization in the scale space of the image [5]. The steps are:

- Scale-space extrema detection: intends to achieve scale invariance.
- Orientation assignment: finds the dominant gradient orientation.
- Descriptor extraction.

Figure 3 shows an example patch with their corresponding HOGs.

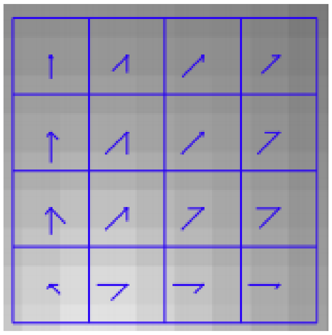

Fig. 3: Example HOG descriptors, patch size=8x8. Each cell of the patch shows the orientation of the gradients.

Orientation histograms have been used in many other methods, so they work really well when they are combined with local spatial histogramming and normalization in Lowe's Scale Invariant Feature Transformation approach [10]. In the case of Shape Context, it studies the cell and block shapes; initially used edge pixel counts without the orientation histogramming.

Advantages of HOG/SIFT representation are: it works with local shapes because it captures edge structure with a controllable degree of invariance to local geometric and photometric transformations (i.e. if translations or rotations are much smaller than the local spatial or orientation bin size, they are little different).

2.1 Bag of Features

Actually, Bag of words method overcomes the other methods for object detection. It represents an image as an orderless collection of local features [7] (i.e. in face representations local features can be an eye, ear, mouth, etc).

However, in face detection, object images belong to the same category (face images), histograms of orderless local features from the whole face do not have large enough between class variations [9].

In Bag of Words [7], orderless local features are extracted from images of different categories (face or non-face) as candidates for basic elements, *i.e.*, "words". Feature descriptors are represented like numerical vectors. By clustering methods, they convert numerical vectors to "codewords" (cluster center) to produce a "codebook". The number of total clusters is the codebook size. So each feature in an image is mapped to a codeword through the clustering process and they are used to represent the histogram (see Figure 4).

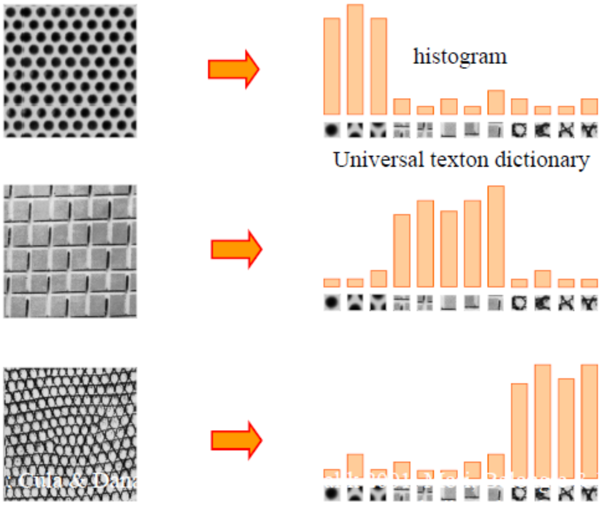

Fig. 4: Bag of features

In our work, we first extract the HOG descriptor of each image (face and non-face) and then we apply the clustering process to these features to obtain clusters with different sizes and each cluster center is the codeword that we used to construct histograms based on the frequency of their appearance in the image. The class of each feature is chosen using the minimum distance to the cluster center. And therefore we build groups of features in each cluster. Finally, we used these histograms to train our SVM classifier to detect faces in an input image. Figure 5, shows our model to extract HOG features and to construct histograms based on codewords.

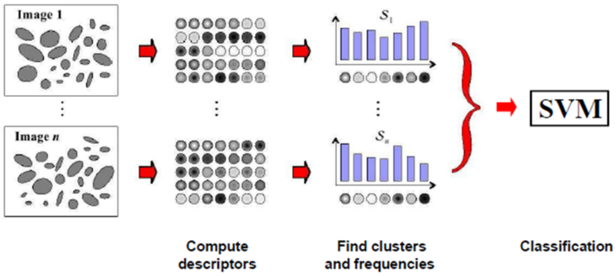

Fig. 5: Proposed Model for Face Detection

3. Implementation

This section describes the implementation of our method. This discussion includes details about the structure of training and building the detector.

3.1 Training dataset

The face training set consists of 2385 faces and 7025 non-faces images/patches of 50 × 50 pixels. The faces were extracted from two different face databases:

- AT&T Database contains ten different images for each of 40 distinct subjects, the images were taken at different times, varying the lighting and facial expressions. All the images were taken against a dark homogeneous background with the subjects in an upright frontal position [1].
- FEI Face Recognition Database is a Brazilian face database that contains a set of face images taken at the Artificial Intelligence Laboratory of FEI in São Bernardo do Campo, São Paulo, Brazil. There are 14 images for each of 200 individuals, adding up 2800 images. All images are colorful and taken against a white homogeneous background in an upright frontal position with profile rotation up to 180 degrees. [14].

We formed our database by choosing some faces of different datasets because we use frontal faces, faces with profile rotation and faces with illumination changes. Non-faces were extracted from images available in [16]. These non-faces have different sizes so we cut each image into sub images in a base resolution of 50 × 50pixels. From this process we obtained 7025 non-face images.

To evaluate our algorithms, we used the Label Faces on Wild Dataset that contains 2845 grayscale and color images with differents sizes, a wide range of difficulties including occlusions, difficult poses, and low resolution and out-of-focus faces, [8].

3.2 SVM Training

For training our models we use the Support Vector Machines algorithm [3], [12] since we need to learn a model to discriminate faces from non-faces samples. The linear kernel was chosen due to its capability to work with high dimensional features. We use the Libsvm library in Matlab for training our algorithms [2]. In contrast to the Viola-Jones algorithm which takes days for training the cascade, the training time was 2 to 4 minutes depending on the amount of training data (images features).

3.3 Classification and Detection

For classifying faces and non-faces we used the feature vector obtained by the histograms of the codewords. Our face detection method receive an input image, extract the feature vector of each candidate image subwindow and then classify as face or non-face by the trained model (see Figure 7).

4. Face Detection Experiments

This section describes experiments for validating the proposed face detector method. The SVM model is built using the entire training set described inSection 3;

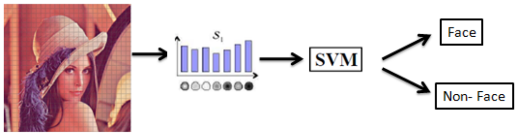

Fig. 6: Face Detection Metohd

First, we presented a set of experiments over the Training Dataset:

We use the standard image databases available on the internet described in section 3. So we have 2385 faces and 7025 non-faces samples to train. We divide the training database (faces+non-faces) because we only train with 30%.

We train our classifier with different codebook size because we wanted to see which one presents better results in our training dataset. First, we train our classifier with a codebook size equal to 10, the Accuracy over 6588 test samples was 99.71% and over 2822 train samples and the Accuracy was 100%.

		Output Class	
		Faces	Non-Faces
Target Class	Faces	711	4
	Non-Faces	4	2103

Table 1: Training Dataset Confusion Matrix, k=10.

We train our classifier with a codebook size equal to 100, the Accuracy over 1911 test samples was 84.51% and over 7499 train samples and the Accuracy was 85.71%.

		Output Class	
		Faces	Non-Faces
Target Class	Faces	307	170
	Non-Faces	0	1434

Table 2: Training Dataset Confusion Matrix, k=100.

For testing our method we used Label Faces on Wild Dataset. False Positive, True Positive and Accuracy are presented below. In this case, we annotated faces per images because when we will pass the detector it will return one rectangle per face and then we will use it to obtain detecion rates.

Figure 8 and Figure 9 presents the face detection of a random image selected.

5. Conclusion

In this paper, we proposed, implemented and tested our Face Detection Method by using the SVM classifier. From experiments, we concluded that we can improve our results by using the Elastic Bunch Graph Matching Method to extract the most important parts in the face (eyes, nose, etc) and from them we can obtain HOG descriptors without using the entire image, so we reduce the number of operations. Moreover, we plan to perform more tests on other databases in order to verify how robust is the proposed method.

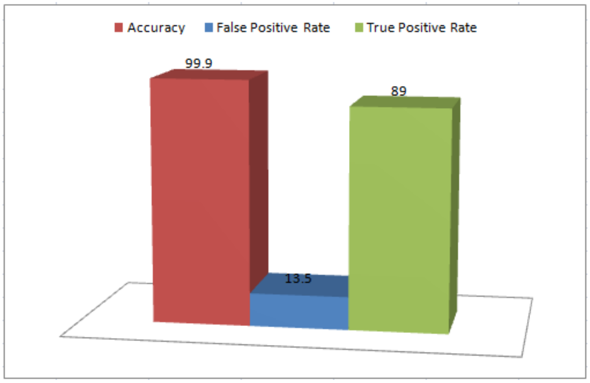

Fig. 7: Detection Rate

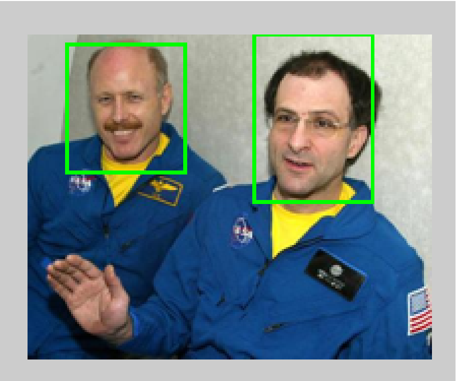

Fig. 8: Face Detection example 1

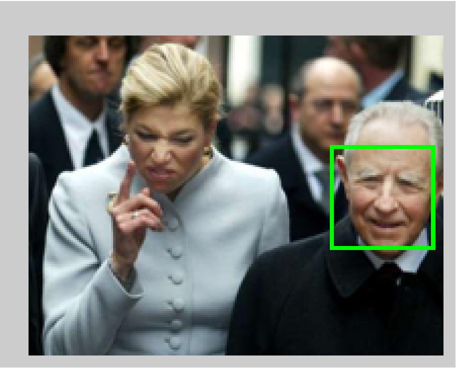

Fig. 9: Face Detection example 2

References

[1] A. L. Cambridge. The database of faces. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html, Apr. 2002.

[2] C.-C. Chang and C.-J. Lin. Libsvm : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:1–27, 2011.

[3] C. Cortes and V. N. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[5] O. Déniz, G. Bueno, J. Salido, and F. De la Torre. Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12):1598–1603, 2011.

[6] P. Dollár, S. Belongie, and P. Perona. The Fastest Pedestrian Detector in the West. 2010.

[7] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE, 2005.

[8] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. *University of Massachusetts, Amherst*, 2010.

[9] Z. Li, J.-i. Imai, and M. Kaneko. Robust face recognition using block-based bag of words. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1285–1288. IEEE, 2010.

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[11] R. O′Malley, E. Jones, and M. Glavin. Detection of pedestrians in far-infrared automotive night vision using region-growing and clothing distortion compensation. *Infrared Physics & Technology*, 53(6):439–449, 2010.

[12] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002.

[13] W. Schwartz, A. Kembhavi, D. Harwood, and L. Davis. Human Detection Using Partial Least Squares Analysis. 2009.

[14] C. L. Thomaz. Fei face database. http://fei.edu.br/ cet/facedatabase.html, 2012.

[15] P. Viola and M. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[16] J. Wu, S. C. Brubaker, M. D. Mullin, and J. M. Rehg. Fast and robust face, rare event detection. http://c2inet.sce.ntu.edu.sg/Jianxin/RareEvent/rare$_event.htm$, 2008.

[17] C. Zhang and Z. Zhang. A survey of recent advances in face detection. Technical report, Tech. rep., Microsoft Research, 2010.