Efficient eye detection using HOG-PCA descriptor

Article *in* Proceedings of SPIE - The International Society for Optical Engineering · February 2014

DOI: 10.1117/12.2036824

Andreas Savakis
Rochester Institute of Technology

Efficient Eye Detection using HOG-PCA Descriptor

Andreas Savakis[1], Riti Sharma[1] and Mrityunjay Kumar[2]
[1]Department of Computer Engineering, Rochester Institute of Technology, Rochester, New York
[2]FiveFocal, Boulder, Colorado, USA

ABSTRACT

Eye detection is becoming increasingly important for mobile interfaces and human computer interaction. In this paper, we present an efficient eye detector based on HOG-PCA features obtained by performing Principal Component Analysis (PCA) on Histogram of Oriented Gradients (HOG). The Histogram of Oriented Gradients is a dense descriptor computed on overlapping blocks along a grid of cells over regions of interest. The HOG-PCA offers an efficient feature for eye detection by applying PCA on the HOG vectors extracted from image patches corresponding to a sliding window. The HOG-PCA descriptor significantly reduces feature dimensionality compared to the dimensionality of the original HOG feature or the eye image region. Additionally, we introduce the HOG-RP descriptor by utilizing Random Projections as an alternative to PCA for reducing the dimensionality of HOG features. We develop robust eye detectors by utilizing HOG-PCA and HOG-RP features of image patches to train a Support Vector Machine (SVM) classifier. Testing is performed on eye images extracted from the FERET and BioID databases.

Keywords: Eye Detection, Histogram of Oriented Gradients, HOG PCA, Random Projections

1. INTRODUCTION

With the proliferation of webcams and mobile devices, eye detection is becoming increasingly important for human computer interaction and augmented reality applications. In the context of mobile interfaces, eye detection provides cues on whether or not the user is looking at the screen [1]. Furthermore, eye detection is the starting point for gaze estimation and eye tracking. Following detection, eye locations can be used for face alignment, normalization, pose estimation, or initial placement of Active Shape/Active Appearance Models for efficient convergence. Several eye detection systems are relying on infrared light emitting diodes, but in general it cannot be assumed that such devices are available. This paper presents a robust and efficient eye detector based on images or video obtained from a single image or video input.

Eyes are the most prominent facial feature and their localization has received considerable attention in the computer vision literature. Eye detection methods include active approaches that require infrared illumination [2], active shape models [3], methods based on geometrical features, such as eye corners or iris detection [4], and methods based on feature classification [5], for example using Support Vector Machine (SVM) classifiers. While infrared illumination provides many benefits, we cannot assume it would always be available, especially when dealing with mobile devices. Furthermore, methods relying on corners or circles are susceptible to noise, low resolution or poor illumination. Active Shape models require good initialization and may be computationally intensive.

In this paper, we take an approach that is based on feature classification using Support Vector Machines. Specifically, we utilize Histogram of Oriented Gradient (HOG) features, which have gained popularity in recent years for a variety of classification problems. The HOG feature was introduced in [6] where it was combined with an SVM classifier to perform people detection. The HOG is a dense descriptor computed on overlapping blocks along a grid of cells. Following its initial success on human detection, the HOG has been adopted for face recognition and smile detection [7], [8]. HOG-based eye detectors were combined with circular Hough transform [9] and used with adaboost for faster performance [10].

In addition to good classification performance, we are interested in computationally efficient processing. To that end, we employ Principal Component Analysis (PCA) for feature dimensionality reduction. We obtain a HOG-PCA descriptor for efficient representation of HOG features before classifier processing. Our HOG-PCA features are computed using Principal Component Analysis on the HOG vector, which significantly reduces feature dimensionality compared to the original HOG size or the eye image patch size. The HOG-PCA descriptor effectively generates Eigen-Eyes in HOG space and offers an efficient feature representation for classifier training and testing. We also consider Random Projections (RPs) as an alternative to the PCA transformation that does not require knowledge of the data statistics.

After computing the HOG-PCA features at a local window, we perform eye detection using a Support Vector Machine (SVM) classifier. Our approach facilitates real time implementation, as the reduced dimensionality of HOG-PCA allows for faster computation during the eye detection classifier stage. The rest of the paper is organized as follows. Following the Introduction, Sections 2 discusses the HOG features Section 3 introduces the HOG-PCA descriptor obtained by dimensionality reduction. Section 4 discusses the experimental setup and classifier training and Section 5 presents the results of our experiments. The paper concludes in Section 6.

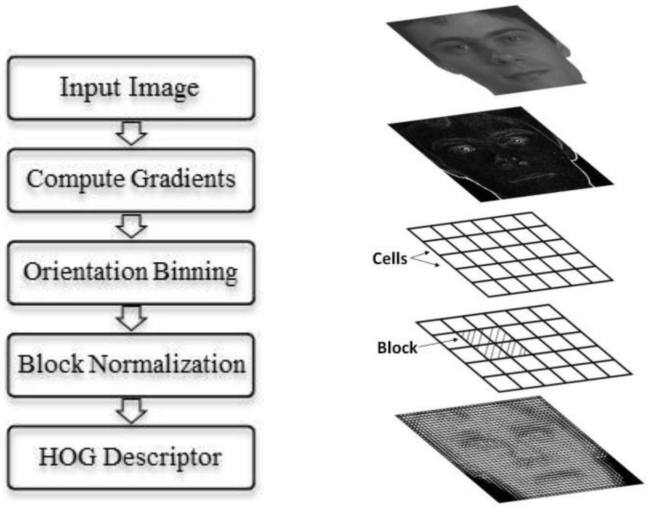


Figure 1. Process of formation of Histogram of Oriented Gradients feature descriptor

2. HOG FEATURES

The success and broad application of the SIFT keypoint descriptors [11] has motivated the development of various local descriptors based on edge orientations. The Histogram of Oriented Gradients is a popular descriptor that was initially proposed for pedestrian detection by Dalal and Triggs [6]. Since then it has been adopted in the context of various classification problems. The process of generating HOG Descriptor for an image is shown in Figure 1. To obtain the HOG features, the gradient of the image is computed by applying an edge mask. A simple 1-D mask of the form [-1, 0, 1] works well. More complex masks, such as Sobel operator and others, were tested in [6], but the simple, centered, 1-D derivative mask works the best. The resulting gradient image is divided into smaller nonoverlapping spatial regions, called cells. These cells can be rectangular, which results in the R-HOG descriptor, or circular, which results in the C-HOG descriptor, as shown in Figure 2.

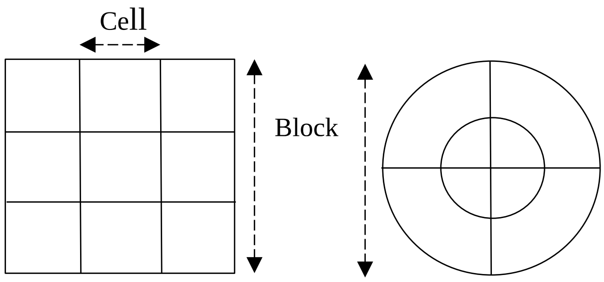


Figure 2. Rectangular and Circular Cells, used in Histogram of Oriented Gradients

Each pixel in the cell casts a vote that is weighted by its gradient magnitude, and contributes to an orientation aligned with the closest bin in the range 0-180 (unsigned) or 0-360 (signed). The orientations of the bins are evenly spaced and generate an orientation histogram. This step is known as Orientation Binning. After Orientation Binning, the cell histograms are normalized, for better invariance to illumination and contrast. The cells are grouped together in 2x2 overlapping blocks and each of these blocks is normalized individually. The overlapping of blocks ensures that there is no loss of local variations and there is consistency across the whole image.

The HOG descriptor has proven effective for classification purposes and has been applied on a variety of problems. Its main disadvantage is the associated computational load, and we utilize dimensionality reduction to ease the computational burden.

3. DIMENSIONALITY REDUCTION

The objective of dimensionality reduction is to project the data on a lower dimensional space while retaining the important properties of the data for the problem under consideration. Working with high dimensional data can be computationally intensive and slow. In cases where the feature space is extremely large, the effect known as the Curse of Dimensionality may result in lower performance compared to that of a lower dimensional feature space. Dimensionality reduction offers computational benefits obtained by processing a smaller amount of data. A secondary benefit of dimensionality reduction is that some noise and redundancies in the features are eliminated after reducing the feature space. As a result, the data can be represented in a more compact and effective way, which can improve classifier performance. Techniques such as Principal Component Analysis (PCA), Linear Discriminant Analysis, and Manifold Learning have been used for subspace learning and dimensionality reduction [12], [13]. Principal Component Analysis is the most widely used approach.

3.1 Principal Component Analysis

Principal component analysis (PCA) transforms the data dimensions by retaining the first few principal components, which account for most of the variation in the original data. Principal components are obtained as linear combinations of the original variables. Let $x_1, x_2, \dots, x_n$ be the original set of data in D dimensional space. We represent this data into some subspace W which has dimension d <*D*. Let y_i; $i = 1, \dots n$ be the linear combination of these variables such that

$$y = A^T(x - m_x) \tag{3}$$

where, $A = [\,\alpha_1 | \dots | \alpha_n]$ is the matrix whose columns are the eigenvectors of Σ, the covariance of the original data, and m_x is the mean of the original data.

Dimensionality reduction using PCA is a well-established and effective technique, but it has the limitation of requiring knowledge of the data statistics. When the mean and covariance of the testing samples do not accurately match these of the training data, this approach could lead to errors. We propose the use of Random Projections as an alternative to PCA for performing dimensionality reduction without requiring knowledge of the data statistics.

3.2 Random Projections

Random Projections (RPs) is a linear dimensionality reduction technique based on the Johnson-Lindenstrauss (JL) lemma [14]. The JL lemma states that for every set *P* of *n* points in R^D there exists a random matrix transformation $y = \boldsymbol{R}x$ from *D*-dimensional space to *k*-dimensional space, such that for given $1>\varepsilon>0$, integer *n*, and selected *k* such that $k \geq O(\log(n)/\varepsilon^2)$, then for all *x, y* in *P*:

$$(1-\varepsilon)\sqrt{k/D} \leq \frac{\|\boldsymbol{R}x - \boldsymbol{R}y\|_2}{\|x-y\|_2} \leq (1+\varepsilon)\sqrt{k/D} \tag{8}$$

The random matrix $\boldsymbol{R} \in \mathbb{R}^{k \times D}$ has elements that are drawn as independent, identically distributed (i.i.d) samples from a zero mean distribution with bounded variance.

The significance of this result is that a Random Projection transformation preserves distances, within some bounds, in the reduced dimensional space. As a result, the complexity of algorithms that depend on the dimensions of their input data, such as a classifier with HOG features, can be significantly decreased without diminishing the ability to perform classification such as k-NN, SVM, etc., in the RP transformed space.

From the implementation perspective, there exists a computationally efficient method for computing the projection matrices of RP transformation. In [15] it was shown that the elements r_{ij} of $\boldsymbol{R}$ can be drawn i.i.d from the following distribution

$$r_{ij} = \sqrt{3}\begin{cases} 1 \text{ with probability } 1/6 \\ 0 \text{ with probability } 2/3 \\ -1 \text{ with probability } 1/6 \end{cases} \quad (9)$$

The transformation based on an RP matrix with coefficients following the above distribution is particularly efficient, since it discards 2/3 of the data that correspond to multiplication by zero. Furthermore, if the scaling coefficient is factored out, the transformation can be implemented using fixed point arithmetic operations consisting of only additions and subtractions.

There are some notable advantages to using RP transformations for dimensionality reduction. Both PCA and RPs involve linear transformations and thus offer fast application through matrix multiplications. The nature of the RP projection matrix is such that the computations are more efficient. Furthermore, unlike PCA, RPs is data-independent, i.e. it is does not require training on a particular dataset and thus allows better generalization. Random projections in combination with SVM classification were used for face detection in [16].

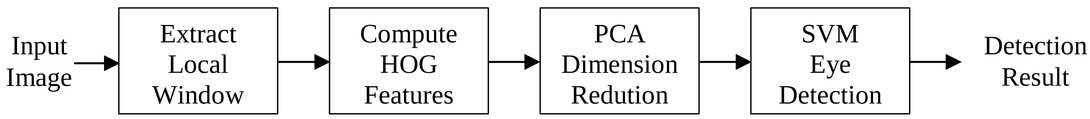
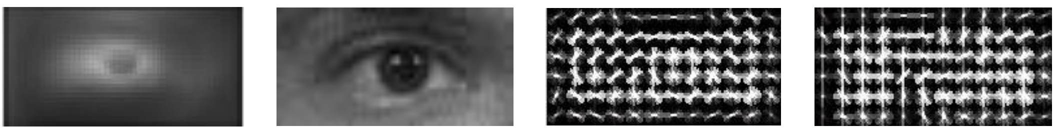


Figure 3. Eye detection using HOG-PCA features.

4. METHODOLOGY AND EXPERIMENTAL SETUP

The overall eye detection system is shown in Figure 3. A sliding window of varying size is necessary for extracting the appropriate window size for eye detection. This process can be computationally intensive and it is combined with face detection to reduce the search area for faster processing. A face detector based on the Viola-Jones detector is used to identify the face region. After face detection, the face region is normalized to a standard size and the local windows for eye detection are extracted from the upper portion of the face region. The HOG-PCA descriptor is computed for each local window and a linear SVM classifier is used to



(a) (b) (c) (d)

Figure 4. (a) Average gradient image eye training examples. (b) Eye Image. (c) HOG Descriptor for Eye. (d) HOG Descriptor for Non Eye.

4.1 Feature Extraction

The SVM classifier training was based on eye and non-eye patches from the FERET and BioID databases. The size of the patches was fixed to 25x50, and all patches were extracted at orientations of 0, ±5, ±10, ±15, ±20 degrees. The extracted eye and non-eye patches were normalized to compensate for variations in illumination. The HOG descriptor for each of these images was found using cell sizes of 8 and 9 bins. The size of the HOG descriptor was 1x648 and was reduced to 1x130 using PCA. An example of the average image gradient is show in Fig. 4(a), an eye patch and its HOG descriptor are shown in Fig. 4(b) and (c) and for comparison purposes, an HOG descriptor example of a non-eye patch is shown in Fig. 4(d).

4.2 SVM Classification

Support vector machine (SVM) classification is the most popular classification method in computer vision. We trained a linear SVM using image patches of eyes and non-eyes. To improve classification performance, we used the bootstrap technique to reduce the number of false positives. The bootstrap process is as follows.

a) Take a subset of samples from the original training set and train the classifier on this subset.

b) Test this classifier on a random subset of samples.

c) If the number of false positives is less than the specified threshold, stop the algorithm.

d) Otherwise, add the false positives to the training set and train a new classifier. Continue with step b).

To test a new face image, sliding window approach is used. The sliding window is of the same size as that of the patches used for training. Each of the sliding windows is separately classified using the trained SVM. After identifying the top 5 patches that are classified as eyes based on the confidence level returned by the classifier, we cluster them to determine the location of the eye.

5. RESULTS

5.1 Datasets

The results were evaluated using the BioID Face Database and the FERET Database. The BioID database consists of 1521 frontal images of 23 different persons. The images are grayscale and have size 384x286 pixels. The database includes annotations of 20 locations of facial feature points for each image.

The FERET database contains both frontal as well as posed faces. The database contains 1564 sets of images with 365 duplicate sets. The duplicate sets contain images of a person already in the database, but taken on a different day. For some individuals, the time elapsed between their first and last sitting was over a year in order to accommodate the changes in appearance over time.

5.2 Classification of Image Patches

The training set consisted of 31,735 eye samples and 28,214 non eye samples extracted from the BioID and FERET databases. The classification performance was tested using 10-fold cross validation technique. The results with HOG descriptor are as shown in Table 1. Table 2 and 3 show the results with HOG-PCA and HOG-RP respectively.

Table 1. Confusion Matrix for classification of 25x50 patches using the HOG descriptor and SVM.

	Eyes	***Non Eyes***
Eyes	98.27	1.73
Non Eyes	1.74	98.6

Table 2. Confusion Matrix for classification of 25x50 patches using the HOG-PCA descriptor and SVM.

	Eyes	***Non Eyes***
Eyes	98.16	1.84
Non Eyes	1.91	98.09

Table 3. Confusion Matrix for classification of 25x50 patches using the HOG-RP descriptor and SVM.

	Eyes	***Non Eyes***
Eyes	98.11	1.89
Non Eyes	2.03	97.97

These results illustrate that the HOG descriptor can be effectively used for eye detection on image patches. Dimensionality reduction of features using HOG-PCA or HOG-RP resulted in comparable results and did not compromise classification performance.

5.3 Eye Detection in Images

To demonstrate the effectiveness of the HOG-PCA eye detector on full images, we used a sliding window approach along with the SVM classification. The results for the BioID database are shown in Table 4. Representative images for correct detection and false positives are shown in Figure 5.

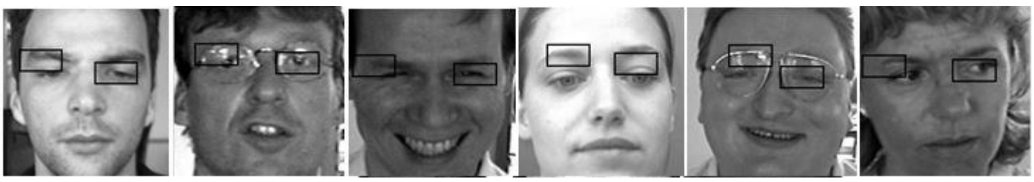
Table 4. Eye Detection Results on BioID Database

Method	Percent Correct		
	Both Eyes	***Right Eye***	***Left Eye***
HOG	96.25	96.32	97.14
HOG-PCA	96.04	96.25	96.79
HOG-RP	95.63	96.05	96.32

(a) HOG descriptor results

(b) HOG-PCA descriptor results

(c) HOG-RP descriptor results



(d) Examples of false positives

Figure 5. Eye Detector examples in BioID database using (a) HOG Descriptor, (b) HOG-PCA Descriptor, (c) HOG-RP Descriptor, and (d) examples of false positives.

The eye detection results in the BioID database demonstrate that the HOG, HOG-PCA and HOG-RP descriptors perform well and their results are comparable within one percent. The overall performance is a bit lower than the detector performance for patch classification due to wider variability in the appearance of patches in full images.

The results for the FERET database are summarized in Table 5. Representative images for correct detection and false positives are shown in Figure 6. The observations again point to comparable performance between the three descriptors, and in this case there is not significant reduction in performance when using the HOG-RP descriptor.

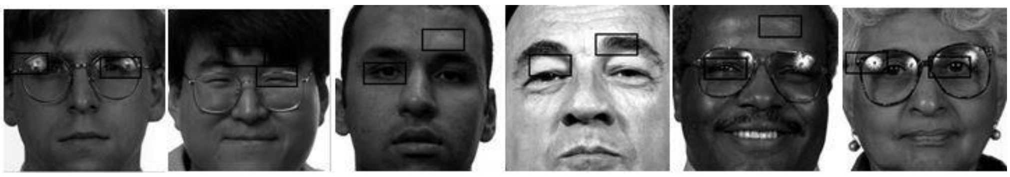
Table 5. Eye Detection Results on FERET Database

Method	Percent Correct		
	Both Eyes	***Right Eye***	***Left Eye***
HOG	98.06	98.33	98.95
HOG-PCA	97.45	97.89	98.68
HOG-RP	97.62	98.07	98.68

(a) HOG descriptor results

(b) HOG-PCA descriptor results

(c) HOG-RP descriptor results



(d) Examples of false positives

Figure 6. Eye Detector examples in FERET database using (a) HOG Descriptor, (b) HOG-PCA Descriptor, (c) HOG-RP Descriptor, and (d) examples of false positives.

6. CONCLUSIONS

In this paper, we presented an eye detector based on HOG-PCA or HOG-RP features and SVM classification. We tested the HOG-PCA and HOG-RP eye detectors on the FERET and BioID databases with good success. Our results demonstrate that eye detection using HOG-PCA and HOG-RP is comparable to that of the HOG descriptor while the feature dimensionality is significantly reduced. The HOG-PCA and HOG-RP features are both discriminative and efficient, which makes it good candidates for real time object detection.

7. ACKNOWLEDGMENTS

This research was supported in part by RealD, Boulder, Colorado. Mrityunjay Kumar was with RealD when this work was performed.

REFERENCES

[1] Miluzzo, E., Wang, T., and Campbell, A. T., "EyePhone: Activating Mobile Phones With Your Eyes," ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds MobiHeld'10 (2010)

[2] Zhu, Z., and Ji, Q., "Robust Real-time Eye Detection and Tracking Under Variable Lighting Conditions and Various Face Orientations," Computer Vision and Image Understanding, 124-154 (2005).

[3] Cootes, T., and Taylor, C., "Active shape model search using local grey-level models: A quantitative evaluation," British Machine Vision Conference, BMVC (1993).

[4] Valenti, R., and Gevers, T., "Accurate Eye Center Location through Invariant Isocentric Patterns," IEEE Trans. on Pattern Analysis and Machine Intelligence, 1785 - 1798 (2012).

[5] Campadelli, P., Lanzarotti, R., and Lipori, G., "Precise eye localization through a general-to-specific model definition," British Machine Vision Conference, BMVC (2006).

[6] Dalal, N., and Triggs, B., "Histograms of Oriented Gradients for Human Detection," IEEE Computer Vision and Pattern Recognition Conf., CVPR (2005).

[7] Déniz, O., Bueno, G., Salido, J., and De la Torre, F., "Face recognition using Histograms of Oriented Gradients," Pattern Recognition Letters, 1598–1603 (2011).

[8] Bai, Y., Guo, L., Jin, L., and Huang, Q., "A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition," IEEE Int. Conf. Image Processing, ICIP (2009).

[9] Ito, Y., Ohyama, W., Wakabayashi, T., and Kimura, F., "Detection of Eyes by Circular Hough Transform and Histogram of Gradient," Int. Conf. Pattern Recognition, ICPR (2012).

[10] Monzo, D., Albiol, A., Sastre, J. and Albiol, A., "Precise eye localization using HOG descriptors," Machine Vision and Applications, pp. 471–480 (2011).

[11] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 91-110 (2004).

[12] Duda, R. O., Hart, P. E., and Stork, D. G., [Pattern Classification], Wiley, (2000).

[13] Ghodsi, A., "Dimensionality Reduction: A Short Tutorial," University of Waterloo Technical Report, (2006).

[14] Bingham, E., Mannila, H., "Random projection in dimensionality reduction: Applications to image and text data," ACM (2001).

[15] Achlioptas, D., "Database-friendly random projections: Johnson-Lindenstrauss with binary coins," Journal of Computer and System Sciences, 671–687 (2003).

[16] Tsagkatakis, G., and Savakis, A., "Random Projections for Face Detection under Resource Constraints," Int. Conf. on Image Processing, ICIP (2009).