Journal of Computer and Communications, 2016, 4, 22-29
Published Online August 2016 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2016.410003

Fast Face Detection with Multi-Scale Window Search Free from Image Resizing Using SGI Features

Masayuki Miyama

Faculty of Electrical and Computer Engineering, Institute of Science and Engineering, Kanazawa University, Kanazawa, Japan
Email: miyama@se.kanazawa-u.ac.jp

Received 27 June 2016; accepted 7 August 2016; published 10 August 2016

Abstract

Face detection is applied to many tasks such as auto focus control, surveillance, user interface, and face recognition. Processing speed and detection accuracy of the face detection have been improved continuously. This paper describes a novel method of fast face detection with multi-scale window search free from image resizing. We adopt statistics of gradient images (SGI) as image features and append an overlapping cell array to improve detection accuracy. The SGI feature is scale invariant and insensitive to small difference of pixel value. These characteristics enable the multi-scale window search without image resizing. Experimental results show that processing speed of our method is 3.66 times faster than a conventional method, adopting HOG features combined to an SVM classifier, without accuracy degradation.

Keywords

Face Detection, Multi-Scale Window Search, Resizing Free, SGI Feature

1. Introduction

Face detection is applied to many tasks such as auto focus control, surveillance, user interface, and face recognition. Image features and machine learning methods to detect faces have been studied for many years. Processing speed and accuracy of the face detection have been improved continuously [1] [2].

Object detection including face detection uses multi-scale window search to detect objects with various sizes. For example, a popular face detector using Haar-like features and a cascade classifier provided by OpenCV adopts the multi-scale window search [3]. In general, the multi-scale window search requires image resizing to

How to cite this paper: Miyama, M. (2016) Fast Face Detection with Multi-Scale Window Search Free from Image Resizing Using SGI Features. *Journal of Computer and Communications*, **4**, 22-29. http://dx.doi.org/10.4236/jcc.2016.410003

calculate image features for each scale. If the feature calculation does not need the image resizing, the multi-scale search will become faster.

This paper proposes a novel method of fast face detection with the multi-scale window search free from image resizing. We adopt statistics of gradient images (SGI) as image features [4], and append an overlapping cell array to improve detection accuracy. The SGI feature is scale invariant and insensitive to small difference of pixel value. These characteristics enable the multi-scale window search without image resizing. Experimental results show that processing speed of our method is 3.66 times faster than a conventional method, adopting HOG features combined to an SVM classifier, without accuracy degradation.

This paper is organized as follows. Next section describes the conventional methods of face detection. Section three describes the novel method of face detection using the SGI feature and the multi-scale windows search free from image resizing. Experimental results show the proposed method is faster than the conventional methods with the similar detection accuracy in section four. Section five concludes this paper.

2. Conventional Methods of Face Detection

Many kinds of image feature using image gradient have been proposed. Haar-like features are frequently used for face detection [5]. To obtain a Haar-like feature, a mask is set on a target image firstly. The mask consists of some neighboring rectangles. A sum of pixel values for each rectangle is calculated next. A difference among the sums for each rectangle is obtained as an image feature lastly. There are many combinations of kind, size, position, and form of the mask. There for many features can be defined in a target image. Effective features for detection are selected using a boosting technique, and a cascade detector is constructed.

A HOG descriptor is based on a set of histograms [6]. The histogram expresses a distribution of a sum of gradient intensities with respect to gradient directions divided by nine levels. In order to obtain the HOG descriptor, a target image is firstly divided into cells in a grid. The grid is a two dimensional array of square cells. A histogram is created for each cell by seeking a sum of intensities for each direction of the image gradient. Neighboring cells construct a square block. The histogram for each cell is normalized using the sum of all bins in a block. The HOG descriptor is a feature vector composed of elements which are all bins of all normalized histograms in the target image. The HOG descriptor is used for people detection combined to an SVM detector.

SVM (Support Vector Machine) is a method for finding a hyperplane that separates learning samples in a feature space to positive samples and negative samples [7]. A distance between the nearest sample and the hyperplane is called a margin. The hyperplane is determined so that the margin is maximized. Samples to decide the hyperplane are called support vectors. If the original feature space is linear inseparable, it is separated in the space after the non-linear mapping, which is called the kernel trick. SVM classifies test data according to the following equation:

$$y(x) = sign\left(\sum_{i=1}^{N} a_i y_i K(x, x_i) + b\right). \tag{1}$$

Here, $y(x)$ is a sign of a class which a test datum x belongs to, x_i is a support vector, y_i is a sign of a class x_i belongs to, a_i is a Lagrange multiplier obtained by training, $K(x, x_i)$ is a kernel function, b is a bias obtained by training. A linear kernel ($x \cdot x_i$ inner product of x and x_i) or an RBF kernel $\left(\exp\left(-\gamma \|x - x_i\|^2\right)\right)$ are often used as a kernel function.

Fast and accurate object detection method was proposed in [8] recently. The method computes multiple registered image channels using various transformations of the input image. Next, features such as local sums, histograms, and Haar wavelets are computed efficiently using integral images. A cascade detector is used for classification.

3. Fast Object Detection Using SGI Features

This section describes statistics of gradient image (SGI) we adopt as image features [4]. Then we explain a novel method of multi-scale object detection free from image resizing.

3.1. SGI Feature

Image gradients are frequently used for image features. The HOG descriptor is based on histograms showing distribution of the intensity with respect to the direction of the gradient. The SGI features also express both intensity and direction of the gradient statistically. Equations below show the SGI features:

$$\begin{cases} \bar{I}_x = \frac{1}{N}\sum_{i\in R} I_x(i) \\ \sigma_x = \sqrt{\sum_{i\in R}\frac{\left(I_x - \bar{I}_x\right)}{N}} \\ \bar{I}_y = \frac{1}{N}\sum_{i\in R}\frac{\left(I_y - \bar{I}_y\right)}{N} \\ \sigma_y = \sqrt{\sum_{i\in R}\frac{\left(I_y - \bar{I}_y\right)}{N}} \\ cc = \frac{\sum_{i\in R}\left(I_x - \bar{I}_y\right)/N}{\sigma_x\sigma_y}. \end{cases} \quad (2)$$

Here, *R* shows a target region and *N* shows the number of pixels in the region. I_x is a gradient image, derivative in x direction of image *I*. $\bar{I}_x$ is a gradient image, derivative in *x* direction of image *I*. $\bar{I}_x$ is an average of $I_x(i)$ in the region R and σ_x is the standard deviation. $\bar{I}_v$ is an average of $I_y(i)$ in the region R, and σ_y is the standard deviation. cc is a correlation coefficient between $I_x(i)$ and $I_v(i)$ in the region R. All features except cc are normalized using an average and a variance for each feature, obtained from training images. All features are scale invariant, and each value range is from −1 to +1. $\bar{I}_x$, σ_x, $\bar{I}_y$, and σ_y are related to the intensity. *cc* is a feature related to the direction.

The equations of σ_x, σ_y and cc can be transformed as follows:

$$\begin{cases} \sigma_x = \sqrt{\overline{I_x^2} - \bar{I}_x^2} \\ \sigma_y = \sqrt{\overline{I_y^2} - \bar{I}_y^2} \\ cc = \frac{\overline{I_x I_y} - \bar{I}_x \times \bar{I}_y}{\sigma_x\sigma_y}. \end{cases} \quad (3)$$

Above equations show that averages of I_x, I_y, I_x^2, I_y^2 and I_xI_y produce these three features. On the other hand, it is well known that the integral image accelerates to calculate a sum and an average of a rectangle region in an image. Thus, the proposed image features for rectangle regions can be quickly generated by integral images of I_x, I_y, I_x^2, I_y^2 and I_xI_y.

To detect objects using the SGI features, a target image is divided into cells in a grid. In this work, we newly append an overlapping cell array, constructed inside of a cell array originally defined on a target image as shown in **Figure 1**, to improve detection performance. The above five features are calculated for each cell. A feature vector, which is an SGI descriptor corresponding to the target image, is generated by collecting all features of all cells. In the example of **Figure 1**, the number of cells is $6\times6+5\times5=61$ and the number of features for each cell is 5, then the number of elements in the SGI descriptor is 305. The SGI descriptor does neither calculate the gradient direction explicitly nor require the histogram normalization for each block, unlike the HOG descriptor. Compared to the HOG descriptor, computation complexity of the SGI descriptor is low.

3.2. Multi-Scale Window Search Free from Image Resizing

Multi-scale window search is necessary for object detection with various sizes. **Figure 2** shows the multi-scale window search. As shown in the left side of the figure, the multi-scale window search repeats window search inside of the image and image resizing. The windows search repeats object detection in a window and window sliding in the image. The window size is always the integer multiple of the cell size, and the window moves by the cell unit. After the window search, the image size is reduced, but the window size and the cell size don't change at the next scale. All cells on all scales have the same size and form, and the cell array for each scale is a regular grid. The image reduction ratio is a decimal in general. To obtain the reduced image, pixel values with decimal coordinates of the original image are required. At the window search of the next scale, integer pixels are read from the reduced image, which is consisted of the decimal pixels of the original image.

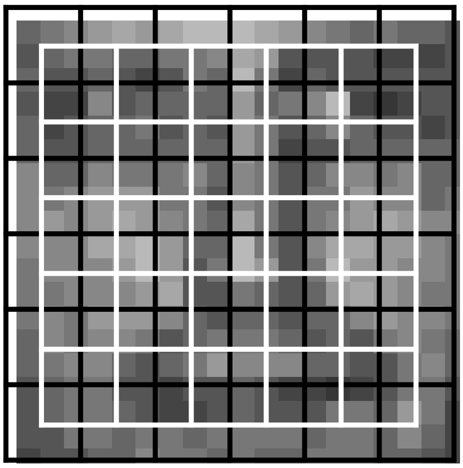


Figure 1. SGI descriptor.

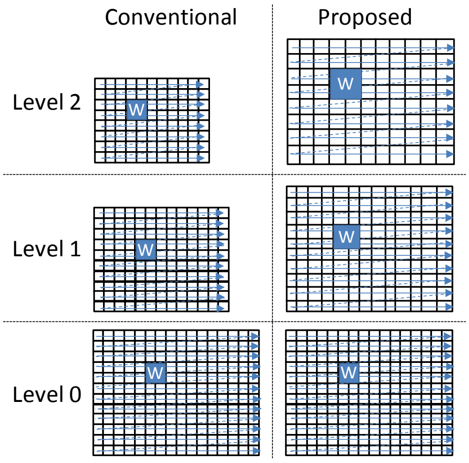


Figure 2. Multi-scale object detection.

The right side of **Figure 2** shows the proposed method. Thanks to the scale invariant characteristic of the SGI feature, the proposed method changes the cell size and the window size instead of the image size. On the upper level of the hierarchy, the decimal coordinates are rounded to the integer coordinates. Then integer pixels are read from the image of the original resolution. As a result of the rounding, every cell size and form on the upper level are various, and the cell array is an irregular grid. Slight difference of the cell size and form does not have a severe influence on the object detection because the SGI features are statistics. The SGI features of all scales can be calculated with the integral image of the original resolution in this way. The image resizing and the integral image calculation for each scale are unnecessary at all by the proposed method.

4. Experimental Results

This section describes experimental results of face classifier generation and face detection.

4.1. Classifier Training and Test

4.1.1. Experimental Setup

We used MIT CBCL Face Database [9] to create training images. The number of positive training images is 2428. The number of negative training images is 4458. The number of positive test images is 2430. The number of negative test images is 4638. An original resolution of the examples was 19×19 pixels. **Figure 3** shows examples of the training image. **Figure 4** shows the average and gradient images of the positive training samples.

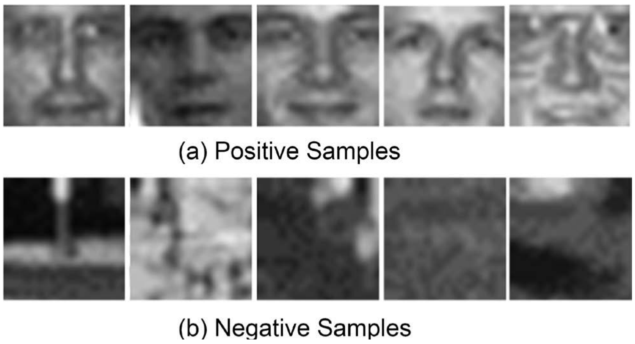


Figure 3. Training images.

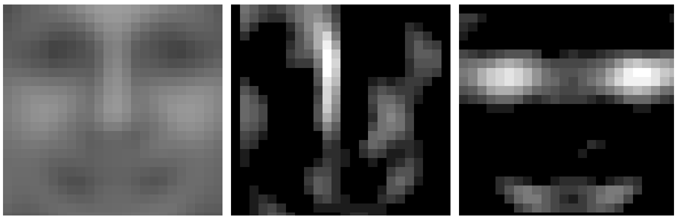


Figure 4. Average image of training samples and its gradient images.

We created a classifier with the HOG descriptor. The size of a grid was 6×6 cells. A function compute () in a class HOGDescriptor of Open CV [3] was used to calculate the HOG descriptor. The size of a cell was 4×4 pixels. The size of a block was 2×2 cells. A function train_auto () in CvSVM class of OpenCV was used to create the classifiers. We used EPS_SVC as an SVM formulation and LINEAR as a kernel.

We also created a classifier with the SGI descriptor. The size of a grid was $6\times6+5\times5$ cells for the descriptor with the overlapping cell array, and 6×6 cells for the descriptor without the overlapping cell array. The size of a cell was 4×4 pixels. A function train_auto () in CvSVM class of OpenCV was used to create the classifiers. We used the EPS_SVR as an SVM formulation and LINEAR as a kernel.

A five points search was executed changing a parameter ε as 0.025, 0.05, 0.1, 0.2, 0.4, and a parameter C was 0.01. The parameter ε is an allowable range of the regression error. The parameter C controls a trade-off between margin size and penalty of misclassification. As C increases, the margin decreases and the penalty increases. Parameters giving the smallest error were obtained by cross-validation of 10 data sets.

The classifiers trained with these parameters were applied to the test images. Pairs of an FPPW (False Positive Per Window) and a miss rate were obtained by changing b in the Equation (1) from −0.8 to +0.8 with 0.02 step, and DET (Detection Error Trade-off) curve was drawn. The FPPW and the miss rate are defined as follows:

$$\mathrm{FPPW}=\frac{\mathrm{FP}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}},\mathrm{Recall}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}},\mathrm{MissRate}=1-\mathrm{Recall}. \tag{4}$$

Here, TP is the number of true positive samples. TN is the number of true negative samples. FP is the number of false positive samples. FN is the number of false negative samples.

4.1.2. Results

Training results are shown in **Table 1**. The descriptor SGI means that with the overlapping cell array. The descriptor SGI_old means that without the overlapping cell array. The ε had the optimal value obtained by the training. The # el is the number of elements in a descriptor.

Figure 5 shows the DET curve of the test. The miss rate of the HOG method was 1.1% with the FPPW of 0.001. The miss rate of the SGI method was 0.3% with the FPPW of 0.001.The miss rate of the SGI_old method was 1.3% with the FPPW of 0.001. The proposed SGI method was higher in accuracy than the HOG method in this experiment. Addition of the overlapping cell array improves accuracy of the proposed method.

Figure 6 shows the SGI features of the positive average images, the SVM coefficients obtained by the training, and the products of the SGI features and the SVM coefficients. The feature images of σ_x, $\bar{I}_y$, and σ_y for the average face are symmetric. The feature images of $\bar{I}_x$, and cc for the average face are antisymmetric. The coefficient image of cc is also antisymmetric. Then the product image of cc becomes almost symmetric. The product image of cc for the average face has many bright cells. This suggests that product of cc has large influence on a judgment whether a face exists in a window. Table 2 shows all kinds of products averaged over all positive samples and all negative samples independently. Every average of positive samples are large, compared to those of negative samples. The product of σ_x has large influence on a judgment of negative samples because the average of negative samples is the smallest. The product of σ_y has large influence on judgments of both positive samples and negative samples because the average of positive samples is the largest and that of negative samples is the second smallest. These suggest variances of gradient strengths are effective for face detection. On the other hand, the HOG features cannot express the variance of gradient strengths. This may give accuracy difference between the HOG method and the SGI method in this experiment.

Table 1. Experimental results of classifier generation.

Descriptor	ε	# el	# TP	# TN	# FP	# FN
HOG	0.4	900	2421	4451	7	7
SGI	0.4	305	2421	4453	5	7
SGI_old	0.4	180	2406	4441	17	22

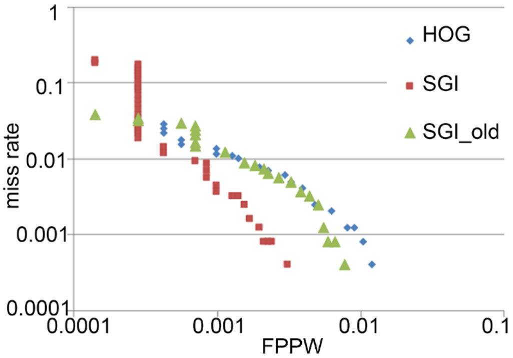


Figure 5. Detection error trade-off curve.

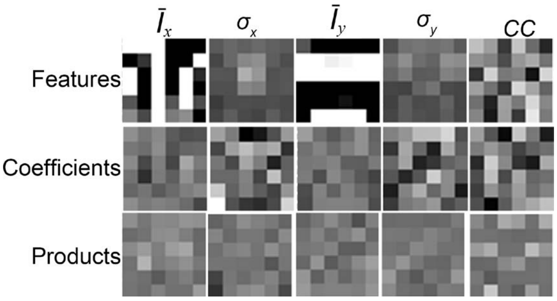


Figure 6. Image of features, coefficients, and products.

Table 2. All kinds of products averaged over positive samples and negative samples.

	$\bar{I}_x$	σ_x	$\bar{I}_y$	σ_y	CC
Average of positive samples	7.25e−03	4.91e−03	4.49e−03	7.91e−03	7.03e−03
Average of negative samples	6.33e−04	−7.14e−05	1.68e−04	−2.30e−05	2.88e−04

4.2. Face Detection

4.2.1. Experimental Setup

This experiment compared four methods below.

- SGI + SVM: calculates SGI features without image resizing and detects faces with an SVM classifier.
- SGI − R + SVM: calculates SGI features with image resizing and detects faces with an SVM classifier.
- HOG + SVM: calculates HOG features with image resizing and detects faces with an SVM classifier.
- HAAR + CAS: calculates Haar-like features and detects faces with a cascade classifier.

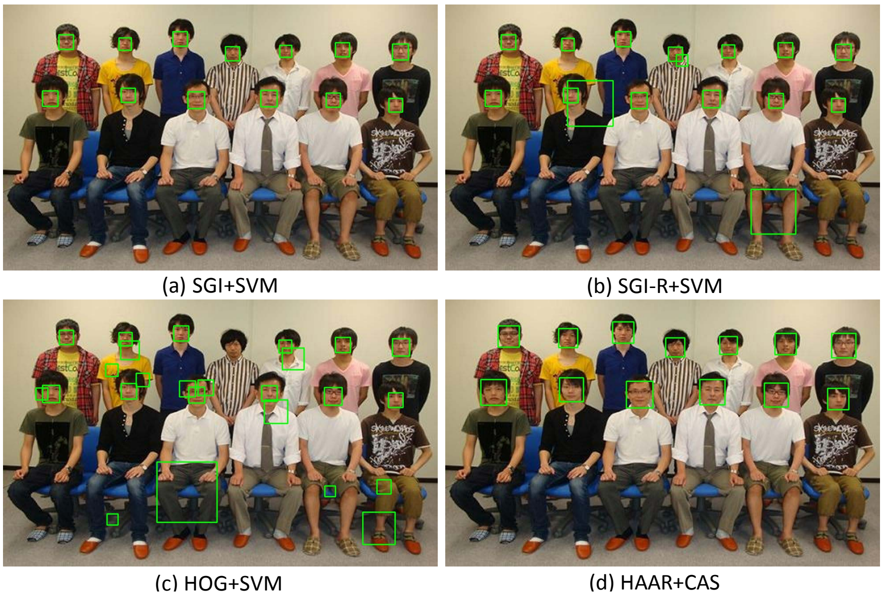
The identical SVM classifier obtained as the result of the previous section was used for both SGI + SVM and SGI − R + SVM. HOG + SVM also adopts the SVM classifier of the previous section. The cascade classifier provided by OpenCV (haarcascade_frontalface_alt. xml) was used for HAAR + CAS. In the multi-scale window search, the smallest windows size was 24×24 pixels and the number of scales was 32. The image resizing ratio of 1.1 was used for HAAR + CAS, and 1.05 was used for otherwise. They were the default values of OpenCV.

The image resolution used for this experiment was 984×602 pixels. The PC used Intel Core-i5 2320@3GHz CPU with 4 GB memory. The OS was Microsoft Windows 10 64 bit. The compiler was Microsoft Visual Studio 2013. The version of OpenCV was 2.4.11.

4.2.2. Results

Experimental results are shown in **Figure 7**. SGI + SVM detected all faces without error detections. SGI − R + SVM also detected all faces, but included error detections. Though they used the same classifier, they produced different results because the SGI features calculated by the two methods were slightly different. HOG + SVM had both miss detections and error detections. HAAR + CAS detected all faces without error detections. The size of the face detected by HAAR + CAS was different because the classifier attached to OpenCV was trained with different images.

Table 3 shows the processing time. SGI + SVM was 1.98 times faster than SGI − R + SVM. The proposed


(a) SGI+SVM

(b) SGI-R+SVM

(c) HOG+SVM

(d) HAAR+CAS

Figure 7. Images of face detection.

Table 3. Experimental results of face detection.

	SGI + SVM	SGI − R + SVM	HOG + SVM	HAAR + CAS
Time (ms)	88.89	176.41	325.09	285.00
Ratio	1.00	1.98	3.66	3.21

method doesn't need image resizing and integral image calculation. SGI + SVM was 3.66 times faster than HOG + SVM. In addition to unnecessity of image resizing, both the feature calculation and the SVM calculation of the proposed method are fast because the number of elements in a feature vector (descriptor) is small. SGI + SVM was 3.21 times faster than HAAR + CAS. If we combine a fast classifier like the cascade classifier with early termination to the SGI feature, the proposed method will be faster.

The proposed method uses integral images. It is similar to the method of [8], which proposes a fast method to calculate image features. Image features can be calculated using another image features at the corresponding point on the neighboring level. The method saves calculations of integral images from resized images on most levels, but those on some levels still remains. In contrast, the proposed method does not need them at all. The SGI feature is scale invariant and insensitive to small difference of pixel value. These characteristics enable to calculate all features on all levels only by the integral images of the original resolution.

5. Conclusion

This paper described a novel method of fast face detection with multi-scale window search free from image resizing. We adopted statistics of gradient images (SGI) as image features, and appended an overlapping cell array to improve detection accuracy. The SGI feature is scale invariant and insensitive to small difference of pixel value. These characteristics enable the multi-scale window search without image resizing. The method of integral channel features [8], which is the state of the art in fast object detection, still requires image resizing instead. Experimental results showed that processing speed increased by 3.66 times without accuracy degradation, compared to the conventional method adopting HOG features combined to an SVM classifier. The proposed method was 3.21 times faster than the most popular method using Haar-like features and a cascade classifier, with similar accuracy. Adoption of a fast classifier such as a cascade classifier is a future work.

Acknowledgements

This study received the furtherance of Grant in Aid for Scientific Research (22560325).

References

[1] Zhang, C. and Zhang, Z.Y. (2010) A Survey of Recent Advances in Face Detection. Technical Report MSR-TR-2010-66.

[2] Degtyarev, N. and Seredin, O. (2010) Comparative Testing of Face Detection Algorithms. *Proceedings of the* 4*th International Conference on Image and Signal Processing*, Trois-Rivières, QC, Canada, 30 June-2 July 2010, 200-209.

[3] http://opencv.org

[4] Miyama, M. (2015) Fast Vehicle Detection System Using Statistics of Gradient Images as Image Features. *IIEEJ Transactions of Image Electronics and Visual Computing*, **3**, 206-214.

[5] Viola, P. and Jones, M.M. (2004) Robust Real-Time Face Detection. *International Journal of Computer Vision*, **57**, 137-154. http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb

[6] Dalal, N. and Triggs, B. (2005) Histograms of Oriented Gradients for Human Detection. 2005 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, 25-25 June 2005, 886-893. http://dx.doi.org/10.1109/cvpr.2005.177

[7] Vapnik, V.N. (1995) The Nature of Statistical Learning Theory. Springer, New York. http://dx.doi.org/10.1007/978-1-4757-2440-0

[8] Dollar, P., Tu, Z., Perona, P. and Belongie, S. (2009) Integral Channel Features. *Proceedings of the British Machine Conference*, September 2009, 91.1-91.11. http://dx.doi.org/10.5244/c.23.91

[9] http://cbcl.mit.edu/software-datasets/FaceData2.html