Robust and Computationally Efficient Face Detection Using Gaussian Derivative Features of Higher Orders

John A. Ruiz-Hernandez[1], James L. Crowley[2], Claudine Combe[2], Augustin Lux[2], and Matti Pietikäinen[1]

[1] Center for Machine Vision Research - University of Oulu, Finland
{john.ruiz,matti.pietikainen}@ee.oulu.fi
[2] INRIA Grenoble-Rhône-Alpes Research Center, France
{james.crowley,claudine.combe,augustin.lux}@inria.fr

Abstract. In this paper, we show that a cascade of classifiers using Gaussian derivatives features up to fourth order can be used efficiently to improve the detection performance and robustness as well when compared with the popular approaches using Haar-like features or using Gaussian derivatives of lower order. We also present a new training method that structures the cascade detection so as to use the least expensive derivatives in the initial stages, so as to reduce the overall computational cost of detection. We demonstrate these improvements with experiments using two publicly available datasets (MIT+CMU and FDDB), in the face detection problem, in addition we perform several experiment to show the robustness of Gaussian derivatives when several transformations are presented in the image.

Keywords: Higher-Order Gaussian Derivatives, Cascade of Classifiers, Face Detection, Half-Octave Gaussian Pyramid.

1 Introduction

Improvements in the cascades of classifiers to deal with the constraints of speed and robustness can be performed in two different ways, optimizing the cascade learning algorithm or finding a more robust set of features. In this paper we propose a combination of these two techniques. In a first time, to address this limitation, we have explored the use of a cascade detector using Gaussian derivative features of higher order, computed in real time with a half-octave Gaussian pyramid [1] as was proposed by Ruiz-Hernandez et al. [2]. We have found that including derivative features up to the fourth order in the cascade can reduce the overall computational cost, while providing improved robustness to image plane rotation and as well as extend detection to lower resolution images. Gaussian derivative features also make it possible to structure the detection cascade in a manner that further reduces computational cost by using lower cost features in the lower levels of the cascade where the great majority of empty face sub-windows are rejected. We will take profit of this property to propose a new

A. Fusiello et al. (Eds.): ECCV 2012 Ws/Demos, Part III, LNCS 7585, pp. 567–577, 2012.

training algorithm that takes into account the computational cost of each derivative order.

1.1 Principal Contributions

- We propose a speed-optimized cascade framework which takes into account the computational complexity of Gaussian derivatives and the local appearance information provided by different derivative orders to select its adequate position in the cascade.
- Use of Gaussian derivatives up to the fourth order are considered in a cascade of classifiers. Despite its high sensitivity to noise, experiments show that inclusion of higher order derivatives improves detection rates.
- We propose a new metric to compute computational load based on the number of requests to the image representation which is more suitable for evaluating feature performance in face detection.
- We perform several experiments for comparing the performance between Gaussian derivative features and Haar-like features when the input image is modified by different transformations such as contrast, noise and rotation.

To present and develop our hypothesis, this paper is organized as follows: Related works and theoretical background are reviewed in Section 1.2. A cascade framework for training speed-optimized cascades of Gaussian derivatives features is presented in Section 2 and experimental results are presented in Sections 3. Section 4 closes the paper with some concluding remarks.

1.2 Related Work and Theoretical Background

In this section we provide review of previous works related with these paper. For a more comprehensive summary of works in face detection we refer the reader to [3] and recently to [4].

A number of researchers have recently proposed methods to improve detection speed by using different feature types in the same cascade. Meynet et al. [5] proposed a cascade in which the first five nodes were composed of Haar-like features followed by nodes composed of anisotropic Gaussian filters. Xiaohua et al. [6] have explored the use of Haar-like features in the first nodes followed by nodes computed using an approximation of Gabor filters computed from an integral image. Roy and Marcel [7] proposed Haar Local Binary Patterns and Yan et al. [8] Locally assembled Binary features. Ruiz-Hernandez et al. [2] proposed to use Gaussian derivatives features up to the second order with some promising results, nevertheless the the paper does not propose to use higher derivatives orders, in addition, there is no intention to find an optimization method that takes into account the computational cost of each derivative order or a large experimental protocol to show the performance of their approach. Despite the detection speed and robustness improvements in the approaches above mentioned, the use of multiple image representations to perform a single task can pose serious problems in embedded systems because of tight constraints

on available memory and computing. Using a Gaussian Pyramid to compute Gaussian derivatives can avoid such problems by providing image features of increasing complexity from a single underlying image representation, besides Gaussian derivatives can be used as image representation in a complex processing pipeline for embedded facial analysis and biometrics systems.

1.3 Computational Cost of Cascade Classifiers

A quantitative measure for the run-time computational cost has been proposed by Brubaker et al. [9]. This measure, referred to as "Computation load", captures the computational cost for classifying a sub-window with the node, including the cost of features belonging to previous nodes.

$$E[T_i] = r_i \sum_{k=1}^{i} M_k \quad where \quad r_i = (i - p_i) \prod_{j=1}^{i-1} p_j \tag{1}$$

Where $E[T_i]$ is the expected computational load for an stage i in the cascade, M_k is the number of features in the node k, p_i is the fraction of sub-windows rejected by the node i and r_i is the fraction of the *decided sub-windows* in the node i.

Brubaker et al. [9] defined the *decided sub-windows* for a node i as the sub-windows that are not passed on to a following node, either because they have been rejected as non-face, or because they have been accepted as a face in the case of a terminal node.

1.4 Computational Cost with Different Types of Features

As shown in Equation 1 the computational load is calculated based in the number of features applied by layer. To extend this concept to compare cascades with different types of features, we propose to use the number of requests for features per mode R_k.

$$E[T_i] = r_i \sum_{k=1}^{i} R_k \tag{2}$$

A *request* is defined as a simple memory access made by the cascade to the image representation such as an integral image or a Half-Octave pyramid. Multiple requests may be necessaries to compute a simple feature in the node k of the cascade.

1.5 Gaussian Derivatives as a Feature Set

The choice of feature set has an important impact on detection rate as well as the scan speed of a cascade detector. We have explored a feature space composed by derivative orders up to four. Derivatives are computed at four different orientations $\theta = \{0, \pi/4, \pi/2, 3\pi/4\}$ in a 24×24 pixel window for all the real sample

positions available in a Gaussian pyramid of three levels $\sigma = \{\sqrt{2}, 2, 2\sqrt{2}\}$. In this way, a 24×24 pixel window gives rise to 8064 possible derivative features.

To test the performance of Gaussian derivatives features, we defined four different feature sets as shown in Table 1. We trained four cascades (one for each feature set) using the algorithms and the training set explained in preceding sections. Each cascade is composed of 21 nodes, except for the cascade trained with the feature set number 3 that has 22. For each trained cascade, we measure the node performance as the false negative rate in the validation set for each node in each cascade and we show the results in Figure 1, the experiment demonstrate that adding high-order Gaussian derivatives reduces the false negative rate for a given node during the training process. In this experiment, the false positive rate in the first three nodes is similar in all the feature sets and then for deeper nodes the inclusion of higher derivative orders dramatically improve the node-performances.

The node performance measure is useful because it directly compares the ability of each feature set to achieve the node-learning goal with a small number of features per node and number of nodes in the cascade.

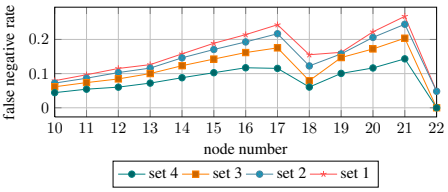


Fig. 1. Node performances for the four cascades trained with the feature sets shown in Table 1. The false negative error rate decreases as the derivative order rises, especially for deeper nodes in the cascade.

Table 1. Four different feature sets using different Gaussian derivative orders at pyramid levels of $\sigma = \{\sqrt{2}, 2, 2\sqrt{2}\}$ and orientations $\theta = \{0, \pi/4, \pi/2, 3\pi/4\}$

Feature Set ($\mathcal{F}_s$)	Derivative Orders	Total
$s = 1$	First Order	1792
$s = 2$	Up to the Second Order	4032
$s = 3$	Up to the Third Order	5824
$s = 4$	Up to the fourth Order	8064

2 Learning Speed-Optimized Cascades

In the preceding section, we have observed the effects of adding Gaussian derivative features up to fourth order in the cascade framework. We have observed that a strong improvement is obtained in the deeper nodes of the cascade. At the same time, higher order derivatives have a slightly higher computational cost. Thus lower computational cost, it appears reasonable to restrict higher order derivatives to deeper levels of the cascade. In this section we explore this hypothesis. Well structured derivatives with scale invariant impulse responses can be computed as sums and differences of adjacent pyramid samples [1]. In such a representation, first order features may be seen as responding to edge-like information, second order features to blob like structures, while higher order derivatives respond to more complex patterns of appearance. The cascade of

classifiers framework can exploit such a representation by using less expensive lower order derivative to reject the large majority of empty windows in the lower levels of the cascade, and applying more expensive higher order derivatives in the deeper nodes.

From this hypothesis, we should expect that Gaussian derivatives features of lower order, less expensive derivative features will perform well in the first nodes when not much information is necessary to discriminate a face from non-face while the more expensive higher order features would be useful in deeper nodes where the difference between a face and a background image becomes more difficult to discriminate. The following experiments confirm this hypothesis. Our proposed optimization-framework is summarized in Algorithm 1.

Algorithm 1. The speed-optimized cascade framework

Giving a set of positive examples $\mathcal{P}$, a set of initial negative examples $\mathcal{N}$, a set of positive validation examples $\mathcal{V}$, a set of bootstrapping negative examples $\mathcal{D}$, a training learn goal $\mathcal{G}$, a training learn goal per layer $\mathcal{G}_{\mathcal{L}}$ and an ensemble of s feature sets $\mathcal{F} = (F_1, F_2, F_3....F_s)$, a value p which represents the desired starting position in the ensemble of feature sets ($p \leq s$);
output: The output is a cascade $H = (H_1, H_2, H_3, ..., H_n)$
initialization: $i \leftarrow 0$,$H \leftarrow 0$;
repeat
  $i \leftarrow i + 1$;
  Node Learning { Learn H_i using $\mathcal{P}$, $\mathcal{F}_p$ and $\mathcal{N}$, add H_i to H};
  Run the current node H_i on $\mathcal{V}$ to compute d_i;
  while $(d_i < \mathcal{G}_{\mathcal{L}}) \wedge (p < s)$ **do**
    $p \leftarrow p + 1$;
    Node Learning { Learn H_i using $\mathcal{P}$, $\mathcal{F}_p$ and $\mathcal{N}$, add H_i to H};
    Run the current node H_i on $\mathcal{V}$ to compute d_i;
  end
  Remove correctly classified non-face patches from $\mathcal{N}$;
  Run the current cascade H on $\mathcal{D}$, add any false detection to $\mathcal{N}$ until $\mathcal{N}$ reaches the same size as the initial set.;
until *The learning goal $\mathcal{G}$ is satisfied*;

The Node-learning step is composed by two algorithms used in this paper to train the cascades :

- Adaboost [10] to find the best set of T features $\mathbf{h} = (h_1, h_2, \cdots, h_T)$ from a high dimensional feature set, giving this set of features, a feature vector for a training z sample can be build as $\mathbf{h}(\mathbf{z}) = (h_1(\mathbf{z}), h_2(\mathbf{z}), \cdots, h_T(\mathbf{z}))$
- LAC (Linear Asymmetric Classifier) [11] to provide an optimal linear strong classifier to accomplish node-learning, while providing the best trade-off between performance and computational cost, for further details please consult [11].

3 Experimental Results

3.1 Experimental Protocols

We performed several experiments to explore the performance of Gaussian derivatives for the face detection problem. Experiments were constructed to test sensitivity to image degradation, as well as performance with different data sets (MIT+CMU and FDDB) and variation in computational load of different feature sets.

Sensitivity Analysis: we constructed a data set to test sensitivity, where images from the LFW dataset [12] are degraded by rotation, blurring, additive Gaussian noise and contrast to evaluate the influence of such factors on detection rate. We applied all the cascades in our experiments to each image in the degraded dataset. For the transformation parameters, we record the detection rate over the set of images and the number of eventual false positives.

- **Rotation** Each image is rotated sequentially by an angle varying between -25 and +25 degrees with an steep of 3 degrees.
- **Blurring** : A Gaussian smoothing filter with scales ranging from 0 to 10 is applied to each image.
- **Noise**: Gaussian white noise with mean 0 and standard deviation between 0 and 0.2 is added to each image.
- **Contrast**: For each image, the pixel intensities I_p are modified as stated by $I_p = \alpha I_m + (1 - \alpha)I_p$, where I_m is the mean intensity of the image and α is a parameter varying from -2 to 1.0.

Comparative Results in Test Datasets: The performance of the cascade is commonly measured by a ROC (Receptive Operator Curve) calculated with an evaluation dataset. In all our experiments we resize the sliding window by a factor $s = 1.20$ which is a common value used in face detection benchmarks. Finally, we compare all our results with a cascade of Haar-like features to show the performance of our approach compared with the state-of-the-art methods using the FDDB [13] and the popular MIT+CMU [14] datasets.

The FDDB dataset is composed by 2845 images with a total of 5171 faces, which are organized in ten-fold testing sets. The implementation-software of the algorithms for matching detections and annotations in this dataset are publicly available[1]. From this software, a detection is scored taking into account $S(d_i, l_j) = {}^{area(l_j) \cap area(d_i)}/_{area(l_j) \cup area(d_i)}$ where d_i and l_j are the rectangle for the detected face and the rectangle of the ground truth respectively.

Two different types of ROC curves could be computed using the above mentioned score. The first one is the discrete score curve, where only the detection scores superior to 0.5 are used an the second is the continuous score curves where all the possible detections scores are included.

[1] `http://vis-www.cs.umass.edu/fddb`

3.2 Sensitivity Results

The results of experiments with the sensitivity test data set are shown in Figure 2.

The sensitivity to rotation can be observed in Figure 2a. In this case the cascade constructed with Gaussian derivatives outperforms that of Haar features. The detection rate for cascades of Gaussian derivatives is 100% for angles between -13 and +5 degrees and decreases significantly for larger rotations, while detection with Haar features is very sensitive to image plane rotation, maintaining a rate of 100% only for angles between -3 and +3 degrees. The number of false positives for this experiment was zero in all the cases.

The results of the influence of blurring are reported in Figure 2b. Notice that the detection rate remains 100% for blurring noise with a standard deviation of 8.3 and decreases slowly for larger standard deviations. The cascade of Haar features maintains a detection rate of 100% only out to standard deviations of 5 and then rapidly decreases. Thus we can conclude that Gaussian derivatives are less sensitive to image blur. No false positive were observed in this experiment.

Tolerance to contrast is reported in Figure 2c. Cascades of Gaussian derivatives and Haar-features cascade provide similar sensitivity, with a slight improvement for Haar-features. In this experiment, no normalisation to illumination was performed. However, subsequent experiments using illumination normalization showed no noticeable improvements.

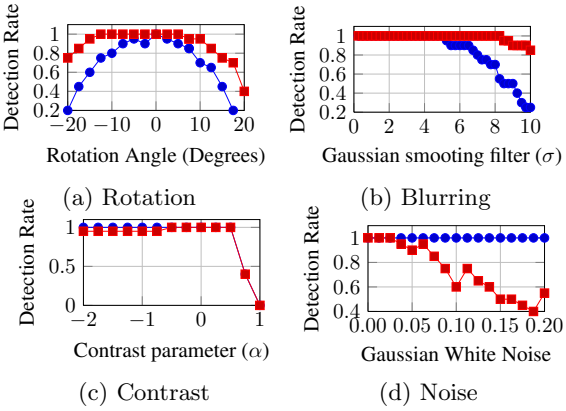
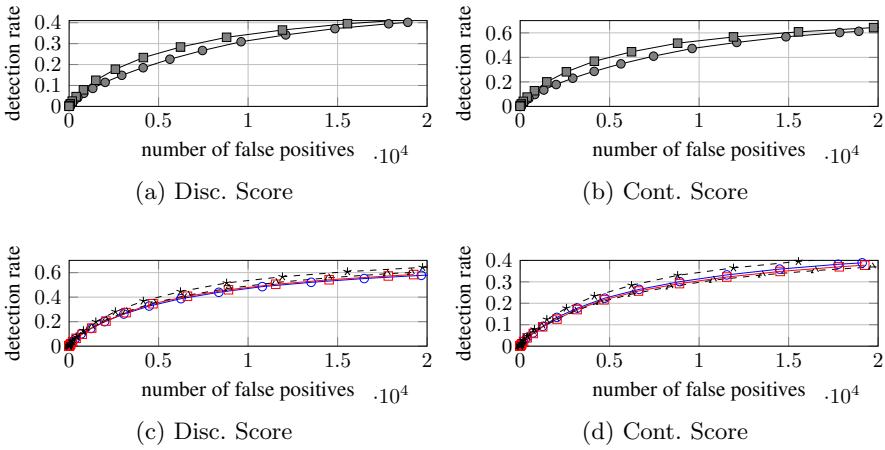


Fig. 2. Results of comparing a non-optimized cascade of Gaussian derivatives (squares) with a cascade of Haar features (circles) in the sensitivity testing dataset.

The influence of additive image noise is reported in Figure 2d. In this case, the cascade of Haar features outperforms Gaussian Derivatives, maintaining a 100% of detection rate (no false positive) for all the standard deviation values used in the Gaussian noise, compared with the cascade of Gaussian derivatives that maintains 100% of detection rate only for values lower than 0.028 (one false positive for a value of 0.06) and decreases slowly as the noise increases. This results can be explained by the sensitivity to noise of third and fourth order Gaussian derivatives.

For most of data sets tested, Gaussian derivatives out-perform Haar-like features in detection rate. For example, Figure 3 shows results of detection



(a) Disc. Score (b) Cont. Score

(c) Disc. Score (d) Cont. Score

Fig. 3. ((a) and (b)) Performance comparison of a non-optimized cascades of Gaussian derivatives features (squares) with a Haar-features cascade (circles) in the FDDB face dataset. ((c) and (d)) Performance comparison of speed-optimized cascades of Gaussian derivatives features in the FDDB face dataset ($p = 1$(squares), $p = 2$(circles), $p = 3$(dashed line with triangles) and non-optimized(dashed line with stars)).

performance for optimized cascades using the FDDB data set. Results are presented as continuous and discrete scores(see Figures 3c and 3d respectively). In both cases the optimized cascades work with a similar performance as non-optimized cascades. For this data set the evaluation was performed using the discrete and continuous scores as is shown in Figures 3a and 3b respectively. Cascade detectors constructed with Gaussian derivative features outperform the cascade of Haar features in detection rate (area under the ROC) by almost a 8%.

3.3 Detection Rates with Different Data Sets

Results using the MIT+CMU face data are shown in Table 2. In this case, the Haar-like features perform better than Gaussian derivatives. We believe that this reflects the conditions under which this data set has been constructed. The MIT+CMU face data set is composed of many images that were scanned from newspapers, thereby introducing high levels of additive noise from both the rendering process used in newspapers. In addition, aliasing is apparent in some images due to a low-quality scanning process. Such noise is not characteristic of that obtained with modern digital cameras. Aliasing is rarely seen with digital cameras, because digital cameras almost always use intentional blurring in front of the CCD to avoid aliasing, finally, some of the facial images in this dataset are draws of faces without any textural information, such kind of images could be considered as a false positive in any biometric or security system.

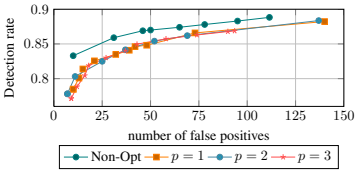


Fig. 4. ROC's of optimized-speed cascades of Gaussian derivatives features in the CMU+MIT face dataset

Table 2. A comparison of detection rates on the CMU+MIT data set for several standard detectors

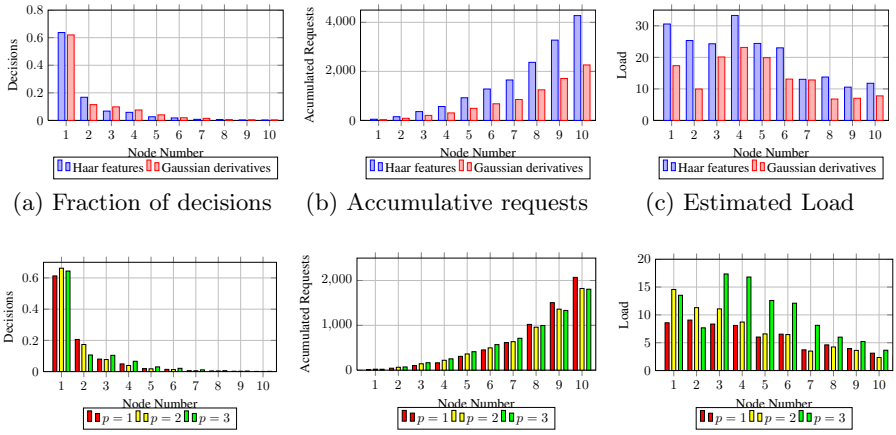
Method	False Positives								
	6	10	31	46	50	65	78	95	167
Garcia and Delakis [15]	-	0.905	0.915	-	-	0.923	-	-	0.931
Li and Zhang [16]	-	0.836	0.902	-	-	-	-	-	-
Luo [17]	0.866	0.874	0.903	-	0.911	-	-	-	-
Viola and Jones [18]	-	0.783	0.852	-	0.888	0.898	0.901	0.908	0.918
Wu et al [11]	-	-	**0.906**	**0.915**	**0.917**	**0.920**	**0.923**	**0.926**	**0.933**
Gaussian Derivatives	-	**0.833**	**0.859**	**0.869**	**0.870**	**0.874**	**0.878**	**0.883**	**0.906**

3.4 Results on Test Data Sets (Optimized)

In Figure 4, we present the results of performance using the MIT+CMU face data set, as we can see, the cascades trained using the speed optimized framework continues to operate satisfactorily compared with the non-optimized cascades, in terms of detection rate and false positive rate.

3.5 Results on Computational Load (Optimized)

Figure 5 shows the results of measurement of computational load for the speed-optimized cascades using the FDDB dataset. In all cases, despite the similar number of feature requests made to the pyramid (see Figure 5e), the optimized



(a) Fraction of decisions (b) Accumulative requests (c) Estimated Load

(d) Fraction of decisions (e) Accumulative requests (f) Estimated Load (optimized)

Fig. 5. (c) Computational Load comparison between a cascade of Haar features and a non-optimized cascade of Gaussian derivatives. The estimated load in a cascade with Gaussian is reduced, specially in the first nodes. (f) Computational Load comparisons in the optimized-cascade framework for different values of p

cascades increase the number of decisions taken (see Figure 5d), especially in earlier nodes where the number of sub-windows to visit is higher and the number of features is lower. This experiment also demonstrates that the computational load for the optimized cascades is decreased by almost half compared with the non-optimized cascades (see Figure 5f). Thus we can expect a gain of a factor of two in detection speed compared with the non-optimized cascades. We also note that a decrease in the number of nodes necessary to accomplish learning for the optimized cascades trained using $p = 2$.

4 Conclusions

In this paper, we have reported results with experiments with the use of Gaussian derivatives features to detect faces in images. We have shown that Gaussian derivatives outperform in more realistic data sets as the FDDB face data set where the images are similar to these produced by the digital cameras used in mobile telephones and devices. In addition, we have demonstrated the robustness of detection using Gaussian derivatives features to image variations as rotation, blurring, noise and contrast using the sensitivity test data set. We have compared all of our results with these obtained with a cascade of Haar features.

References

1. Crowley, J.L., Riff, O.: Fast Computation of Scale Normalised Gaussian Receptive Fields. In: Griffin, L.D., Lillholm, M. (eds.) Scale-Space 2003. LNCS, vol. 2695, pp. 584–598. Springer, Heidelberg (2003)
2. Ruiz-Hernandez, J.A., Lux, A., Crowley, J.: Face detection by cascade of gaussian derivates classifiers calculated with a half-octave pyramid. In: IEEE Face and Gesture Recognition, pp. 1–6 (2008)
3. Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: A survey. IEEE TPAMI 24, 34–58 (2002)
4. Zhang, C., Zhang, Z.: A survey of recent advances in face detection. Technical Report MSR-TR-2010-66, Microsoft Research (2010)
5. Meynet, J., Popovici, V., Thiran, J.P.: Face detection with boosted gaussian features. Pattern Recognition 40, 2283–2291 (2007)
6. Xiaohua, L., Lam, K.M., Lansun, S., Jiliu, Z.: Face detection using simplified gabor features and hierarchical regions in a cascade of classifiers. Pattern Recognition 30, 717–728 (2009)
7. Roy, A., Marcel, S.: Haar local binary pattern feature for fast illumination invariant face detection. In: BMVC (2009)
8. Yan, S., Shan, S., Chen, X., Gao, W.: Locally assembled binary (lab) feature with feature-centric cascade for fast and accurate face detection. In: IEEE CVPR, pp. 1–7 (2008)
9. Brubaker, S.C., Wu, J., Sun, J., Mullin, M.D., Rehg, J.M.: On the design of cascades of boosted ensembles for face detection. IJCV 77, 65–86 (2008)
10. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55, 119–139 (1997)

11. Wu, J., Brubaker, S., Mullin, M., Rehg, J.: Fast asymmetric learning for cascade face detection. IEEE TPAMI 30, 369–382 (2008)
12. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst (2007)
13. Jain, V., Learned-Miller, E.: Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst (2010)
14. Rowley, H., Baluja, S., Kanade, T.: Rotation invariant neural network-based face detection. In: IEEE CVPR (1998)
15. Garcia, C., Delakis, M.: Convolutional face finder: A neural architecture for fast and robust face detection. IEEE TPAMI 26, 1408–1423 (2004)
16. Li, S.Z., Zhang, Z.: Floatboost learning and statistical face detection. IEEE TPAMI 26, 1112–1123 (2004)
17. Luo, H.: Optimization design of cascaded classifiers. In: IEEE CVPR, pp. 480–485 (2005)
18. Viola, P., Jones, M.J.: Robust real-time face detection. IJCV 57, 137–154 (2004)