Boosted Gaussian Classifier with Integral Histogram for Face Detection

Haijing Wang
School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang, China
haijing.wang@yahoo.com

Peihua Li
College of Computer Science and Technology, Heilongjiang University, Harbin, Heilongjiang, China
peihualj@hotmail.com

Tianwen Zhang
School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang, China

Novel features and weak classifiers are proposed for face detection within the AdaBoost learning framework. Features are histograms computed from a set of spatial templates in filtered images. The filter banks consist of Intensity, Laplacian of Gaussian (Difference of Gaussians), and Gabor filters, aiming at capturing spatial and frequency properties of faces at different scales and orientations. Features selected by AdaBoost learning, each of which is corresponding to a histogram with a pair of filter and template, can thus be interpreted as boosted marginal distributions of faces. We fit the Gaussian distribution of each histogram feature only for positives (faces) in the sample set as the weak classifier. The results of experiment demonstrate that classifiers with corresponding features are more powerful to describe the face pattern than haar-like rectangle features introduced by Viola and Jones.

Keywords: Face detection; integral histogram; Gaussian; filter; AdaBoost.

1. Introduction

Face detection, which refers to detect and localize an unknown number of faces in the given image, has been extensively studied for its various applications in many fields such as security, multimedia retrieval, and human computer interaction. Viola and Jones 1 present a well-known framework dealing with face detection rapidly, whereas maintaining comparable performance with the state of art face detection algorithms. There are three key contributions of their work 1.

1) The first is the haar-like rectangle feature with the introduction of a new image representation called “integral image”, which allows features used by the detector to be computed very quickly.
2) The second is the AdaBoost learning algorithm, which selects a small number of critical visual features and yields extremely efficient classifiers.
3) The third contribution is a method for combining classifiers in a “cascade” architecture, which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions.

1.1. Related Work

Many researchers present their work following Viola-Jones’ from three-folds: features set, AdaBoost algorithm, and the cascade classifier architecture.

From the view of feature set, Lienhart et al. extend Viola-Jones’ features definition from 4 to 15, which include edge, line, center-surround, and special diagonal line features. Mita et al. introduce joint Haar-like feature. Murphy et al. use a set of filters, including edge filters, corner detection filters, and the Laplacian filter, to convolve the image. And the second and the fourth moments are utilized to construct weak learners of one dimensional histogram from the special patch on the filtered images. Recursive nonparametric discriminant analysis (RND) features are learned and combined with AdaBoost to form a multi-view face detector.

Wu et al. describe two variations of Viola-Jones’ work, a symmetric method which weights false positives and false negatives equally, and an asymmetric approach which is based on a Bayes risk criterion that assigns more weight to false negatives. Xue and Ding propose a multi-modal boosting algorithm to integrate 3D (range) and 2D (intensity) information provided by a facial scan to detect the face and feature points. Other variations of AdaBoost for face detection include FloatBoost, vector Boosting, asymmetric AdaBoost algorithm, multi-class boosting, and so on.

To improve the cascade classifier structure, Ichikawa et al. propose a component-based face detection approach using AdaBoost and decision tree. Huang et al. take the magnitude and phase of Gabor filter response as features, and a polynomial neural network (PNN) as the classifier. Tu proposes probabilistic boosting-tree (PBT) for learning two-class and multi-class discriminative models.

1.2. Proposed Framework of Face Detection

In this paper, we present novel features and classifiers for face detection in the paradigm of AdaBoost algorithm. Features are histograms computed from a set of spatial templates in filtered images. Our filter bank is similar to the one presented by Zhu et al., in which a set of filters are selected from the filter bank based on histogram characterizing distribution of texture according to Maximal Entropy. Briefly speaking, the steps of the proposed algorithm include image preprocessing, filter selection, histogram statistics, fit Gaussian distribution, and feature selection based on AdaBoost learning, as listed below (see Fig. 1).

1) Image preprocessing. We take histogram equalization for both training sam-
ples and test samples to make each image with equally distributed brightness levels over the whole brightness scale.

2) **Filter bank for face detection.** We convolve each image patch with three kinds of filters: intensity, Laplacian of Gaussian (Difference of Gaussians), and Gabor filters, to capture spatial (frequency) properties of faces at different scales and orientations.

3) **Histogram statistics as the feature.** We summarize the responses of the patch convolved with filters using histograms, which represent marginal distributions of these patches.

4) **Fit Gaussian distribution as the proposal of weak learner.** We make a
very simple assumption by fitting the Gaussian distribution for each histogram feature only to positives (faces) in the sample set. Then for each weak learner we determine the best threshold to separate face and non-faces examples in accordance with this Gaussian.

5) **Feature selection using AdaBoost learning.** To decide which features describe the face pattern best, features are selected by AdaBoost learning. The features, each of which is corresponding to a histogram with a pair of filter and template, can thus be interpreted as boosted marginal distributions of faces.

This paper is arranged as follows. The bank of filters is described in Section 2. In Section 3, we present the histogram feature set. Weak learner based on Gaussian assumption is shown in Section 4. In Section 5, the AdaBoost training of our detector is described. Experiments are presented in Section 6. Finally, conclusions are given.

### 2. Filter Bank for Face Detection

Transform domain features can exhibit high "information packing" properties compared with the original input samples. The original sample (gray image) by filtering operation can capture spatial and frequency properties of faces at different scales and different orientations. Our filter bank includes three kinds of filters: 1) the intensity filter \( \delta(\cdot) \), which captures the DC component; 2) the isotropic center-surround filters, i.e., the Laplacian of Gaussian (LoG)/ Difference of Gaussians (DoG) filters; and 3) the Gabor filters \(^1\) with both sine and cosine components.

#### 2.1. Intensity Filter

The ideal impulse in the image plane is defined by Dirac distribution \( \delta(\cdot) \), which captures the DC component. We may express the image function as a linear combination of Dirac pulses located at the points \((a, b)\) that cover the whole image plane

\[
\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b) \delta(a - x, b - y) dadb = f(x, y) \tag{1}
\]

where samples are weighted by the image function \( f(x, y) \). In essence, the response of the intensity filter is the image signal itself.

#### 2.2. Laplacian of Gaussian / Difference of Gaussians Filter

Considering the Laplacian operator of an image smoothed by a 2D Gaussian smoothing, we get a convolution mask of a Laplacian of Gaussian (LoG) operator as shown in Eq. (2):

\[
\text{LoG}(x, y) = \frac{1}{\pi \sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) e^{-\frac{x^2 + y^2}{2\sigma^2}} \tag{2}
\]
where the standard deviation $\sigma$ is proportional to the size of the neighborhood on which the filter operates. It can be shown that LoG is the derivative with respect to $2\sigma^2$ of a Gaussian. In order to avoid the large computation of the LoG operator, the DoG operator (Difference of Gaussians) (see Eq.(3)) can be used as an approximation to the LoG by taking the difference of two Gaussians having different standard deviations.

$$\text{DoG}(x, y) = \frac{1}{2\pi \sigma_1^2} e^{-\frac{x^2 + y^2}{2\sigma_1^2}} - \frac{1}{2\pi \sigma_2^2} e^{-\frac{x^2 + y^2}{2\sigma_2^2}}$$ (3)

The ratio $\sigma_1/\sigma_2 = 1.6$ results in a good approximation of the LoG.

2.3. Gabor Filter

In the spatial domain, a Gabor wavelet \(^1\) is a complex exponential modulated by a Gaussian function. Its kernels are similar to the 2D receptive field profiles of the mammalian cortical simple cells, exhibiting desirable characteristics of spatial locality and orientation selectivity.

The Gabor filters can be defined as follows, assuming that $\sigma_x = \sigma_y = \sigma^{10}$:

$$\psi_{\mu, \nu}(\vec{z}) = \frac{|\vec{k}_{\mu, \nu}|^2}{2\pi \sigma^2} e^{-\frac{|\vec{k}_{\mu, \nu}|^2}{2\pi \sigma^2}} [e^{i\vec{k}_{\mu, \nu} \cdot \vec{z}} - e^{-\frac{|\vec{k}_{\mu, \nu}|^2}{2\pi \sigma^2}}]$$ (4)

In Eq.(4), $\mu$ and $\nu$ stand for the orientation and scale of the Gabor kernels, $\vec{z} = (x, y)$ is a given pixel, $\|\cdot\|$ denotes the norm operator, and the wave vector $\vec{k}_{\mu, \nu}$, restricted by a Gaussian envelope function, is defined as follows:

$$\vec{k}_{\mu, \nu} = k_\nu e^{i\phi_\mu} = \begin{pmatrix} k_\nu \cos \phi_\mu \\ k_\nu \sin \phi_\mu \end{pmatrix}$$

$$k_\nu = a^{-\nu} f_{\text{max}}$$

$$\phi_\mu = \mu \frac{2\pi}{n}, \mu = \{0, \ldots, n-1\}$$ (5)

In Eq.(5), $k_\nu$ is the $\nu$-th frequency, and let $f_{\text{max}} = \pi/2$ be the highest frequency desired, and $a$ is the frequency scaling factor ($a > 1$). Useful values for $a$ include $a = 2$ for octave spacing and $a = \sqrt{2}$ for half-octave spacing. The width $\sigma/k_\nu$ of the Gaussian is controlled by the parameter $\sigma = 2\pi$. $\phi_\mu$ is the $\mu$-th orientation and $n$ is the number of orientations to be used. However, the computation can often be reduced to half since responses on angles $[\pi, 2\pi]$ are the phase shifted from responses on $[0, \pi]$ in a case of a real valued input.

3. Feature Generation with Integral Histogram

Image filters remove information redundancies. We assume that a set of reference patterns (templates) are available in this section. To seek statistical models that avoid making strong assumptions about distributional structure while still retaining good properties for estimation, the best compromise we found was histograms.
3.1. Template Set

Given a $p \times q$ image, any rectangle template $t$ is a tetrad noted as $t = (x, y, w, h)$, where $x$ and $y$ are the location of horizontal and vertical coordinate of template $t$, respectively, and $w$ and $h$ represent width and height of template $t$, respectively. The rules are listed as follow.

1) Both width and height of each template are no less than eight pixels. And the step is eight pixels. It means $w \in \{8i | i = 1, 2, \ldots, \lfloor \frac{q}{8} \rfloor \}$ and $h \in \{8i | i = 1, 2, \ldots, \lfloor \frac{p}{8} \rfloor \}$.

2) The rectangle templates are created in a step of eight pixels along both horizontal and orthogonal direction. There are $x \in \{8i | i = 0, 1, \ldots, \lfloor \frac{q}{8} \rfloor - 1 \}$ and $y \in \{8i | i = 0, 1, \ldots, \lfloor \frac{p}{8} \rfloor - 1 \}$.

3) Each template should satisfy that $x + w \leq q$ and $y + h \leq p$.

4) Each template includes $p \times q$ pixels at least.

According to the rule above, there are 1,024 templates for $64 \times 64$ image (see Fig. 2). In repeated experiments, we find that 132 templates are less chosen during the AdaBoost training for intensity filter. Therefore, we choose 892 different rectangle templates from the total 1,024 ones to form the template set for intensity filter. We take 40 and 15 templates for LoG/DoG filters and Gabor filters, respectively. In Fig. 2, the gray rectangles are the masks used to calculate the histogram feature. To speed up the calculation of histogram, we propose “integral histogram”.

3.2. Integral Histogram

Inspired by the work of Viola and Jones \cite{14}, our histogram features can be computed rapidly using an intermediate representation of the image which is called the “integral histogram image”.

Given an $n \times m$ image, we need to create $(n+1) \times (m+1)$ arrays of length $L$ (the number of possible gray levels), noted as $H_{x,y}[p]$. The integral histogram $H_{x,y}[p]$ corresponds to the number of pixels falling into the $p$th bin, the spatial location of which is above and to the left of $(x, y)$. 

$$H_{x,y}[p] = \sum_{x' \leq x, y' \leq y} \delta(x', y')$$  \hspace{1cm} (6)

where $\delta(x', y') = 1$ if the intensity of pixel $(x, y)$ belongs to the $p$th bin of histogram; otherwise $\delta(x', y') = 0$. The steps of integral histogram algorithm are listed as follows.

1) Initialize $H_{x,0}[p] = 0$, $H_{0,y}[p] = 0$, and integral row histogram $h_{x,y}[p] = 0$, where $x = 0, \ldots, m$; $y = 0, \ldots, n$, and $p = 1, \ldots, L$.

2) Using the following pair of recurrences:

$$h_{x,y}[p] = h_{x,y-1}[p] + \delta(x, y)$$
$$H_{x,y}[p] = H_{x-1,y}[p] + h_{x,y}[p], \quad p = 1, \ldots, L$$  \hspace{1cm} (7)

the integral histogram can be computed in one pass over the original image.
Thus, the histogram $h_r[p](p = 1, \ldots, L)$ of any rectangle region $r(x, y, w, h)$ can be determined in $(4 \times L)$ array references by the integral histogram image:

$$h_r[p] = H_{x+w,y+h}[p] - H_{x+w,y}[p] - H_{x,y+h}[p] + H_{x,y}[p], \quad p = 1, \ldots, L$$

where $w$ and $h$ are the width and height of rectangle $r$, respectively.

4. Proposal of Weak Learner under Gaussian Assumption

Building a model for the face detection task is challenging because of the difficulty in characterizing prototypical “non-face” images. Instead we make a very simple assumption by fitting the Gaussian distribution for each histogram feature only to positives (faces) in the sample set. Then for each weak learner we determine
the best threshold to separate face and non-faces examples in accordance with this Gaussian.

Assume now that the likelihood function of feature \( \lambda_i \), with respect to histogram feature of sample \( x \) in the \( d \)-dimensional feature space, follow the general multivariate Gaussian density \( \mathcal{N}(\mu, \Sigma) \):

\[
p(x|\lambda_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1}(x - \mu_i)\right), i = 1, \ldots, M \tag{9}
\]

where \( d \) is according to the dimensions of the histogram, and \( M \) is the feature count. The \( d \)-component mean vector of the feature \( \lambda_i \) is described as

\[
\mu_i = \mathbb{E}[x] = [v_1, v_2, \ldots, v_d]^T \tag{10}
\]

The \( d \times d \) covariance matrix \( \Sigma_i \) is defined in Eq.(11), whose determinant and inverse are denoted as \( |\Sigma_i| \) and \( \Sigma_i^{-1} \), respectively.

\[
\Sigma_i = \mathbb{E}[(x - \mu_i)(x - \mu_i)^T] = [\sigma_{pq}]_{d \times d}, \quad p, q = 1, \ldots, d \tag{11}
\]

where \( (x - \mu_i)^T \) is the transpose of \( (x - \mu_i) \).

Given histogram features

\[
X = (x_1, x_2, \ldots, x_n), \quad x_k = [x_{1k}, x_{2k}, \ldots, x_{dk}]^T \in X \tag{12}
\]

the \( j \)-th component of \( \mu_i \) is described as

\[
v_j = \sum_{k=1}^n x_{jk}\omega_k \tag{13}
\]

where \( \omega_k \) is the weight of \( x_k \), and \( \sum_{i=1}^n \omega_i = 1 \). In Eq.(11), \( \sigma_{pq} \) is defined as

\[
\sigma_{pq} = \sum_{k=1}^n \omega_k(x_{pk} - \nu_p)(x_{qk} - \nu_q) \tag{14}
\]

The feature value of all samples (both positives and negatives) can now be calculated via Eq.(9). Next, features are selected by AdaBoost learning.

5. Learning Classification Functions by AdaBoost

Utilizing AdaBoost \(^2\), each trained classifier produces a weak classification rule with one feature. The weight distribution is updated at each round of learning. The threshold of the final strong classifier is decided by the prescribed hit ratio of the strong classifier to positives. The construction of the final cascade detector depends on the ratio of false positives for the training set. Features used are the histogram feature with corresponding Gaussian fitting described in the previous section. To speed up the process of detection, only the intensity filter is adopted at the first few stages. Then Difference of Gaussians and Gabor filters are added to the feature set used for training.
6. Experimental Results

In this section, we first introduce the training data set, filters, and feature set. Then training results and the detection performance are described.

6.1. Training Data Set

We crop 10,000 frontal face images as training samples. The negative samples are collected by selecting random sub-windows from a set of 24,621 images which do not contain faces. For each layer, the maximum size of the negative set is 10,000. Each sample is scaled to 64 by 64 pixels. We take histogram equalization for both training samples and test samples to make each image with equally distributed brightness levels over the whole brightness scale.

6.2. Filters and Feature Set

To LoG/DoG filter, nine groups of $(\sigma_1, \sigma_2)$ are taken into account. We choose two scales and five orientations $(0, \pi/8, 3\pi/8, 5\pi/8, 7\pi/8)$ for Gabor filter. Thus, our feature set includes 1,402 histogram features, which is far less than the size of Viola and Jones’ feature set.

6.3. Training Results

The final detector of Viola and Jones is a 38 layer cascade of classifiers which included a total of 6,060 features. However, our cascade detector only includes 11 layers with 428 features. It is trained following two steps: “coarse” learning, which speeds up both training and detection, and “fine” learning, which picks up more “meaningful” features. The coarse learning gets 280 intensity histogram features with six layers. Fig. 3 shows the first two selected features.
To demonstrate that our whole feature set is powerful to describe the face pattern, we continue to train the 7th and 8th layer by two ways. When the training features are only based on intensity filter, the detector holds 137 and 198 features for the 7th and 8th layer, respectively. However, when the whole feature set based on intensity, LoG/DoG, and Gabor filters is used, there are only four features and nine features at the 7th and 8th layer while achieving the same performance. Thus the whole feature set is used to "fine" learning. Finally, we get five layers holding 148 features by fine learning, which include 49 features based on the intensity filter, 45 features based on DoG filters, and 54 features based on Gabor filters. Our approach is promising to decrease the number of features, which is supported by the experimental result, therefore, the proposed features contain rich information for face detection.

6.4. Detector Performance

To evaluate the performance of our detector, we train a cascaded classifier containing six 20-feature classifiers according to what Viola and Jones have done in their experiment \(^{14}\). The first stage classifier in the cascade is trained using 5,000 faces and 10,000 non-face sub-windows randomly chosen from non-face images. The second stage classifier is trained on the same 5,000 faces plus 1,094 false positives of the first classifier. This process continues, that is, subsequent stages are trained using the false positives of the previous stage. ROC curves comparing the performance of our detector and Viola and Jones’ are given in Fig. 4.

![ROC curves](image.png)

**Fig. 4.** ROC curves comparing a cascaded classifier containing ten 20-feature classifiers (by Viola and Jones) with our cascaded classifier containing six 20-feature classifiers.

The detector scans across the image at multiple scales and locations. Faces of different size are detected by scaling the image. And the test set includes 178 images with 651 faces under different light and resolution. The corresponding ROC curve
is shown in Fig. 5. Here, the detector of Viola and Jones is a 25-layer detector from OpenCV. The result of experiments demonstrates that the selected features are more powerful to describe the face pattern than the haar-like rectangle features used by Viola and Jones.

7. Conclusion

This paper presents novel histogram features for face detection in the paradigm of AdaBoost learning. First, intensity, Laplacian of Gaussian (Difference of Gaussians), and Gabor filters are used to capture spatial and frequency properties of human faces at different scales and different orientations. Then, the responses of the patch convolved with filters are summarized with histograms. For simplicity and efficiency, we fit Gaussian distribution to histogram features only based on positives. Finally, features, each of which is corresponding to a histogram with a pair of filter and template, are selected with AdaBoost learning. The experiment results demonstrate that the selected features are powerful to describe the face pattern.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (NSFC) under Grant Number 60505006 and 60673110 and Natural Science Foundation of Heilongjiang Province (F200512), supported in part by Science and Technology Research Project of Educational Bureau of Heilongjiang Province (1151G033), the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, the Excellent Scientific Research Project for the Returned Overseas Scholars, Ministry of Personnel of China, Science and Technology Innovation Research Project (2006RFLXG030) of Harbin Sci. & Tech. Bureau, Postdoctoral Fund for Scientific Research of Heilongjiang Province (LHK-04093).
References

17. F. Xue and X. Ding, “3D+2D Face localization using boosting in multi-modal feature space,” *18th Int. Conf. on Pattern Recognition,* Hong Kong, China, (2006) 499–502
Biographical Sketch and Photo