Face Detector with Oriented Multiple Templates

Yea-Shuan Huang Wei-Cheng Liu

***Abstract*—This paper proposes a novel face detection algorithm which extracts a local image structure (LIS) feature and adopts a boosting approach to construct a cascaded face detector. Due to the locality of LIS, the extracted feature is not only robust to lighting variation but also is invariant to small degrees of rotation. With this robust property a multiple-template cascaded detection algorithm has been developed which can avoid rotating the image and also keep the ability to detect slanted faces. Because the multiple templates are constructed in the initialization stage, the proposed face detector can be executed rapidly. Experiments on the BioID face database have shown the efficiency of this method.**

***Index Terms*—Object Detection, Feature Extraction, Classifier, Local Image Structure**

I. Introduction

Due to the fast advancement of the computer-vision technology and the common usage of cameras (either still or mobile) installed in both our surrounding environment and electronic appliances, many vision-based applications such as video surveillance, man-machine interface, biometrics, and interactive game have been actively developed. Among these applications, face detection is a very essential preprocessing task that has attracted a lot of research attentions. Face detection locates the face candidates appeared in an image so that there are the right targets to perform further processing such as tracking and recognition. So far, many face detection algorithms [1, 2, 3, 4] have been proposed in the past decades. Among them, the boosting-based approaches in general have achieved the best performance on both detection accuracy and processing speed. Therefore, boosting-based face detectors have become the main research stream and indeed they can achieve a practical performance with a fast processor. But, almost all methods are still too slow to applications that use either a low-level processor for a single channel video signal or a high-level processor for multiple channel video signals. Therefore, it is crucial to further speedup the face detection process.

In general, a boosting-based face detector adopts a fixed size face template and is a multi-stage face filter. Each stage uses one or a few features to filter out the non-face regions. Only passing all stages successfully, a region is regarded as a face region. A processed image should be both scaled and rotated several times accordingly so that different orientations and sizes of faces can be detected. Unfortunately, rotation of the whole image is quite time-consuming. Also, for each rotated image, it is necessary to re-compute the image feature. Therefore, the current detectors tend to be slow. If a method can avoid the operation of image rotation and still keep the detection ability of rotated faces, it will certainly execute more efficiently. To achieve this goal, one possible way is to extract a rotation-invariant face feature. However, it is very difficult to design such an ideal feature which needs to satisfy two conditions (rotation invariance and good discrimination between face and non-face) at the same time. Alternatively, one may implement a feature which value is just computed once from the original image but the corresponding feature values of the rotated image can be derived directly from the already computed feature. The main contribution of this paper is to provide one such feature and to design an appropriate mean so that face detection can be executed fast.

In this paper, we propose a novel face detection algorithm which avoids the image rotation operation for any rotation angles between $\pm 22.5^{o}$. Section 2 describes the feature extraction method and shows its robustness to the lighting variation. Section 3 describes the used boosting algorithm to construct a face detector with a single template. Section 4 propose an improve face detector by avoiding rotating images with multiple templates. Section 5 designs a fast multi-template cascaded face detector which has no image rotation operations. Section 6 shows the experimental results on the famous BioID face database, and finally Section 7 concludes this paper.

II. Feature Extraction by Local Image Structure

Let x be an image pixel, $I(x)$, $R(x)$ and $L(x)$ be individually the image intensity, the reflectance vector and the illumination vector of x. From the image formation optics, it exists $I(x) = R(x).L(x)$. For another image pixel y, it becomes $I(y) = R(y).L(y)$. Suppose pixels x and y are neighbors, then $L(x)$ and $L(y)$ are assumed to be very similar or even the same. So

$$\frac{I(x)}{I(y)} \approx \frac{R(x)}{R(y)}. \qquad (2)$$

This equation shows that the ratio of two neighboring pixels is almost independent of illumination vector, and it approximately keeps constant if only the background illumination changes and all other factors remain unchanged. However, besides illumination there are other factors (such as surface normal and noises) which can slightly affect the image intensity. Therefore, it is not proper to directly take the image ratio as features. Instead the contrast relationship (larger than or not larger than) is a better and more stable feature description in representing the relation among image pixels in the neighborhood. For any two pixels x and y, a contrast relationship is defined as

Manuscript received Januarary 7, 2008. This work was supported by the Ministry of Economic Affairs, R.O.C. under the project **"**Construction of Vision-Based Intelligent Environment (VBIE)**",** the project code is 96-EC-17-A-02-S1-032.

Y.S. Huang is with the Computer Science & Information Engineering Department, Chung-Hua University, HsinChu, Taiwan (e-mail: yeashuan@chu.edu.tw).

$$\zeta(I(x), I(y)) = \begin{cases} 0, & \text{if } I(x) \geq I(y); \\ 1, & \text{otherwise.} \end{cases} \quad (3)$$

The relationship obtains a value 0 if pixel x is not darker than pixel y, otherwise it obtains a value 1. Let $\Phi(x) = \{P_0, P_1, \ldots, P_{N-1}\}$ be an N-element set and each element correspond to a chosen neighboring pixel of x. This set $\Phi(x)$ is called the interestedly selected pixel set, which specifies how many and what pixels are chosen to derive the local image structure. It is not difficult to realize that by integrating a set of relationship $\xi(I(x), I(p_n)), \quad 1 \leq n \leq N\text{-}1$ among pixel x and its selective neighboring pixels $(P_0,\ldots,P_{N-1})$ a structure-like information can be constructed. In fact, the structure-like information semantically represents one kind of image texture information. With pixel relationships, the local image structure $\Gamma(x)$ is designed as

$$\Gamma(x) = \sum_{i=0, p_i \in \Phi(x)}^{n-1} 2^i \times \zeta(I(\mathrm{p}_i), I(\mathrm{x})). \quad (4)$$

Figure 1 is an example of $\Phi(x)$ where the pixels marked I are the selected interested neighbor pixels. Practically, $\Phi(x)$ can be either assigned manually or determined by experiments. Figure 2 shows the adopted $\Phi(x)$ in this paper which uses only 8 nearest neighbors of x to construct its local structure. Such a 3×3 $\Phi(x)$ is also called the Local Binary Pattern (LBP). Obviously, with this configuration the corresponding $\Gamma(x)$ has in total 256 possible values ranging from 0 to 255, and each value corresponds to one specific image structure. For example, when $\Gamma(x)$ is 0, it means that x is the brightest pixel among its 8 neighbors. For another example, when $\Gamma(x)$ is 7, it means that among its 8 neighbors only P_0, P_1 and P_2 are darker than x. This structure representation is very robust to many kinds of variations, especially the illumination variation. In Figure 3, an illustrative example is shown. The local image structures of 5 different-illumination images are visualized as index images where the structure index determines the pixel intensity. The 4 right images in the first row of Figure 3 are generated from the first image of the same row by using an image processing software with the following parameters: light value −75, contrast value −50, gamma value 2.0, and gamma value 0.4, respectively. It reveals that the designed image structure is fairly robust to the illumination variation.

III. Construction of Cascaded Face Detector

The face detector analyzes image patches W of size 15×15 pixels. The whole classification procedure consists of a sequence of stage tests performed on the analyzed patch. For each test, the patch may be rejected as background or be accepted as a possible face region for further tests. Only passing all the stage tests, a patch is considered to be a face candidate. Let $\mathrm{H_j}(\Gamma)$ be the jth stage test which classifies the current patch with one or a few positions of the local image structure (LIS) of this patch. Figure 4 shows a constructed face detector of which the LIS of both positions (3, 5) and (4, 1) are tested in the first stage, and the final stage test takes the LIS of 128 positions. Each test consists of a threshold, and when the summation of the outputs of weak classifiers is larger than its threshold, the processed patch is considered to be background and is rejected for further test. The positions of LIS and their corresponding thresholds are obtained from a proposed boosting algorithm as described below:

Step 1: Given N_f face training images and N_{nf} non-face training images of which each image is normalized to be 15×15 pixels.

Step 2: Compute the LIS features of all face and non-face images. Let Γ_m^f and Γ_n^{nf} be the generated LIS of the face and non-face images, respectively, where $m = 1,\ldots,N_f$ and $n = 1,\ldots,N_{nf}$, and $\Gamma_m^f(\mathrm{p})$ is the LIS value of position p of the mth face image, where p belongs to the total position set A={0,…,224}.

Step 3: Initialization. Let the iteration index t=1, $D_1^f(m)$=0.5/N_f, $D_1^{nf}(n)$=0.5/N_{nf}, the set of the selective pixel positions S=Φ an empty set, where $D_1^f(m)$ and $D_1^{nf}(n)$ are individually the initial weight of the mth face image and the nth non-face image at the first iteration.

Step 4: Generate look-up tables of the total corresponding weights $g_t^f(\mathbf{p}, \gamma)$ and $g_t^{nf}(\mathbf{p}, \gamma)$ of the face and non-face images for each position p∈A and structure $\gamma \in \{0,\ldots,255\}$

$$g_t^f(\mathbf{p}, \gamma) = \sum_{m, \mathbf{p}, \gamma} D_t^f(m)\, F(\Gamma_m^f(\mathbf{p}) = \gamma)$$

and

$$g_t^{nf}(\mathbf{p}, \gamma) = \sum_{n, \mathbf{p}, \gamma} D_t^{nf}(n)\, F(\Gamma_n^{nf}(\mathbf{p}) = \gamma)$$

where $F()$ is the indicator function that takes 1 if the argument is true and 0 otherwise.

Step 5: Calculate error δ_t of position p from the computed lookup table:

$$\delta_t(\mathbf{p}) = \sum_{\gamma} \min\{g_t^f(\mathbf{p}, \gamma), g_t^{nf}(\mathbf{p}, \gamma)\}, \quad \forall \mathbf{p} \in \mathrm{A}.$$

Step 6: Select the best position $\mathbf{p}_t$ at iteration t and update the selected position location set S:

$$\mathbf{p}_t = \arg\min_{\mathbf{p} \in \mathrm{A}} \delta_t(\mathbf{p}), \quad \mathrm{S} = \mathrm{S} \cup \{\mathrm{p_t}\}$$

Step 7: Generate another lookup table for the weak classifier at iteration t and position $\mathbf{p}_t$:

$$w_t(\gamma) = \begin{cases} 0 & \text{if } g_t^f(\mathbf{p}_t, \gamma) > g_t^{nf}(\mathbf{p}_t, \gamma) \\ 1 & \text{else} \end{cases}, \quad \gamma = 0, \ldots 255.$$

Step 8: Obtain a value α_t from the error at iteration t:

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \delta_t}{\delta_t}\right).$$

Step 9: Assign the penalty of the current weak classifier:

$$h_{\mathbf{p}_t}(\gamma) = \alpha_t w_t(\gamma), \quad \gamma = 0, \ldots, 255.$$

Step 10: Update the distributions:

$$A(m) = D_t^f(m) \times \begin{cases} e^{-\alpha_t} & \text{if } w_t(\Gamma_m^f(\mathbf{p})) = 0 \\ e^{\alpha_t} & \text{else} \end{cases},$$

$$B(n) = D_t^{nf}(n) \times \begin{cases} e^{-\alpha_t} & \text{if } w_t(\Gamma_m^{nf}(\mathbf{p})) = 1 \\ e^{\alpha_t} & \text{else} \end{cases},$$

$$D_{t+1}^f(m) = \frac{A(m)}{Z_{t+1}},$$

$$D_{t+1}^{nf}(m) = \frac{B(m)}{Z_{t+1}}$$

where z_{t+1} is a normalization factor determined by the following equation:

$$z_{t+1} = \sum_m A(m) + \sum_n B(n).$$

Step 11: Increase the iteration index t with 1. If t is smaller than a pre-specified stopping value T, go to Step 4. Otherwise, go to Step 12.

Step 12: Obtain the final strong classifier at ith stage based on the final face model:

$$H_i(\Gamma) = \sum_{\mathbf{p} \in S} h_{\mathbf{p}}(\Gamma(\mathbf{p})).$$

Here, symbol A denotes the total pixel position, i.e., A={0,…,224}, symbol t denote the iteration index, $\Gamma(\mathrm{p}) = \gamma$ denotes that the local structure of pixel p is γ, $g_t^f(\mathbf{p}, \gamma)$ and $g_t^{nf}(\mathbf{p}, \gamma)$ are individually the total weights of face samples and non-face samples with $\Gamma(\mathrm{p}) = \gamma$ at iteration t. In step 5, the individual error of $\Gamma(\mathrm{p}) = \gamma$ is the smaller value between $g_t^f(\mathbf{p}, \gamma)$ and $g_t^{nf}(\mathbf{p}, \gamma)$ because this smaller value corresponds to the possible error if a decision is made to decide whether a patch is face or non-face when only observing the information that the local structure of pixel P is γ (i.e. $\Gamma(\mathrm{p}) = \gamma$), and the total error $\delta_t(\mathbf{p})$ of position p is the summation of the individual errors of $\Gamma(\mathrm{p}) = \gamma$. For the t-th iteration, a position $\mathrm{p_t}$ is chosen and is added into the selected position set S if it has the minimal error among the errors of all positions. Also, each structure γ at position $\mathrm{p_t}$ has a penalty value $h_{\mathbf{p}_t}(\gamma)$. Basically, $h_{\mathbf{p}_t}(\gamma)$ is determined by two factors derived from samples of $\Gamma(\mathrm{p}) = \gamma$. The first factor is whether the total population of non-face samples $g_{\mathrm{t}}^{nf}(\mathbf{p}_t, \gamma)$ is larger than that of face samples $g_{\mathrm{t}}^f(\mathbf{p}_t, \gamma)$ or not as checked in step 7, and the second factor is the penalty is proportional to α_t which in fact is computed from $\delta_t(\mathbf{p}_{\mathrm{t}})$ in step 8. The first factor tells us when there are more samples belonging to non face than belonging to face, a penalty value can be practically assigned. On the contrary, when there are more samples belonging to face than belonging to non face, the penalty should be quite conservative and even be 0 as assigned in this algorithm. Because α_t is computed from $\delta_t(\mathbf{p})$ and $0 \le \delta_t(\mathbf{p}) \le 0.5$, the second factor in fact means that a larger $\delta_t(\mathbf{p}_{\mathrm{t}})$ derives a smaller $h_{\mathbf{p}_t}(\gamma)$ and a smaller $\delta_t(\mathbf{p}_{\mathrm{t}})$ derives a larger $h_{\mathbf{p}_t}(\gamma)$. A large $\delta_t(\mathbf{p}_{\mathrm{t}})$ in fact denotes it is difficult to make a decision whether the current sample belongs to face or non face. By the proposed algorithm, a background patch will inherently obtain a large response value and therefore will be rejected in the early stage tests. But, a face patch will obtain a small response value and will pass all stage tests.

IV. Face Detection by Single Template

From Section 3, a cascaded face detector is constructed. The positions $\mathrm{p_t}$ and their corresponding penalties $h_{\mathbf{p}_t}(\gamma)$ are optimally derived by an algorithm proposed in [5]. Suppose there are N positions (i.e. S={$\mathrm{p_1}$,…,$\mathrm{p_N}$}) are selected in total which are accordingly formed into M stages with threshold {$\mathrm{h_1}$,…,$\mathrm{h_M}$}. Due to the limited space, the formation of stages is not explained in this paper. Let $\mathrm{S_i}$ denote the extra positions used in the i-th stage which are not used in the (i-1)-th stage , $\mathrm{p_j^i}$ be the j-th position of $\mathrm{S_i}$ and $\mathrm{n_i}$ be the total number of positions in $\mathrm{S_i}$. Then $\mathrm{S_i} = \{\mathrm{p_1^i}, \cdots, \mathrm{p_{n_i}^i}\}$ and $\mathrm{S=S_1 \cup S_2 \cup \ldots \cup S_M}$. In the first stage, $\mathrm{n_1}$ positions defined in $\mathrm{S_1}$ are used to compute the LIS of the current processed 15×15 patch W, and the computed LIS value of each position p becomes the index to retrieve its corresponding penalty value $h_{\mathbf{p}}(\Gamma(\mathbf{p}))$. Therefore, the penalty value of the first stage is $H_1(\Gamma) = \sum_{\mathbf{p} \in S_1} h_{\mathbf{p}}(\Gamma(\mathbf{p}))$. If $H_1(\Gamma)$ is larger than $\mathrm{h_1}$, W is regarded to be a non-face pattern and is rejected for any further check. Otherwise, $\mathrm{n_2}$ positions defined in $\mathrm{S_2}$ will be used to calculate their penalties in stage 2 and the total penalty until stage 2 becomes $H_2(\Gamma) = H_1(\Gamma) + \sum_{\mathbf{p} \in S_2} h_{\mathbf{p}}(\Gamma(\mathbf{p}))$. Again, if $H_2(\Gamma)$ is larger than $\mathrm{h_2}$, W is regarded as a non-face pattern and is rejected. Sequentially, the similar process can be performed for stage 3, stage 4,…, and stage M. Only passing all M stages successfully, a patch W is considered to be a face pattern.

In order to detect various sizes and orientations of face patterns, image transforms such as scaling and rotation should be performed on the processed image accordingly. Each image transform not only spends a lot of computation time but also needs to compute the LIS features of the transformed image. Therefore, the face detection becomes slow.

V. Fast Face Detection by Multiple Templates

For improving the detection speed, it is essential to totally avoid or significantly reduce the image transform and feature extraction. Fortunately, the 3×3 LIS as shown in Figure 2 computes the image structure by using only 8 nearest

neighboring pixels which are quite robust to image rotation. For a 3×3 region, the LIS features of any rotation angle between +22.5° and -22.5° in fact is inherently the same to each other. Therefore, it is unnecessary to re-compute the LIS feature of each pixel if the image is rotated under ±22.5°. However, the chosen positions from a 15×15 template can be used to detect only the upright front face. For detecting a certain angle of rotated faces, instead of rotating the whole image, we rotate the temple and select the corresponding positions of the rotated template. In Figure 5, we use a 4×4 template to simplify the explanation. Suppose the origin of the template is at the left bottom, the 4 selected positions of the cascaded face detector are marked by a white-gray color and their positions are (0,2), (1,3), (2,1), and (3,0). In fact, each position is described by two values relative to the origin of which the first value is the horizontal offset and the second value is the vertical offset. Therefore, position (2,1) denotes the grid which is two-pixel right and one-pixel above to the origin of the template. For a -15° rotated template, the corresponding grid offsets to the origin are (1,2), (2,3), (2,0), and (3,-1) respectively as shown in Figure 4. With this realization, it is easy to derive different templates to detect different angles of rotated faces. For example, 7 templates are constructed for detecting faces under ±15° rotation as shown in Figure 6 and each template supervises a range of 5° rotated faces.

For making a formal analysis of the proposed method, some symbols are defined here. Let Q be the input image consisting of M×N pixels, s be the time of resizing Q with ratio D, r be the time of rotating Q, f be the time of computing the LIS of Q, and d be the time of detecting faces of Q. Suppose Q is resized in total R times, D is the resize ratio of each time, and the overall operation time of each resized image including image scaling, image rotation, and LIS computation is approximately proportional to the pixel number of the resized image. Since both the resize ratios of width and length are D which is smaller than 1.0, the first resized image has $M\times N\times D^2$ pixels, the second resized image has $M\times N\times D^4$ pixels, and the *n*th resized image has $M\times N\times D^{2n}$ pixels. Traditionally, a boosting approach uses only one template to detect all faces of different sizes and different rotations by continuously resizing and rotating image. Let $T_{tradition}$ be the processing time of a traditional boosting method which resizes the original image *n* times, and for each resized image there are R times of both image rotations and LIS feature extractions. Therefore, to detect faces within ± 22.5°, $T_{tradition}$ requires

$$\begin{aligned} T_{tradition} &= D^2\times[s + R\times(r+f+d)]\ldots\text{(the 1st resized image)} \\ &\quad + D^4\times[s + R\times(r+f+d)\text{.....(the 2st resized image)} \\ &\quad + \ldots \\ &\quad + D^{2(n-1)}\times[s+R\times(r+f+d)]\text{...(the } n\text{th resized image)} \\ &= [s + R\times(r+f+d)]\,[D^2+\ldots.+D^{2(n-1)}] \\ &= [s + R\times(r+f+d)]\ \sum_{i=1}^{n-1} D^{2i} \end{aligned}$$

With the proposed method, an image still needs to be scaled and be re-computed the LIS of each scaled image, but there is no need to rotate an image and re-compute the LIS of an rotated image. Therefore, the total required time $T_{proposed}$ of this method becomes

$$\begin{aligned} T_{proposed} &= D^2\times[s + f + R\times d]\text{ .....(the 1st resized image)} \\ &\quad + \ldots \\ &\quad + D^{2(n-1)}\times[s+ f + R\times d]\text{ ...(the } n\text{th resized image)} \\ &= [s + f + R\times d]\,[D^2+\ldots.+D^{2(n-1)}] \\ &= [s + f + R\times d]\ \sum_{i=1}^{n-1} D^{2i} \end{aligned}$$

By dividing $T_{tradition}$ by $T_{proposed}$, the speedup ratio S becomes

$$S = \frac{T_{tradition}}{T_{proposed}} = \frac{[s + R\times(r + f + d)]}{[s + f + R\times d]}$$

For an example of a 320*240 image, suppose s=1ms, r=3ms, f=2ms, d=3ms, D=0.9, n=8 and R=5. Then $T_{tradition}$ is 82ms, $T_{proposed}$ is 34ms, and S is about 2.4.

VI. Experiments

In this section, we use the images in the BioID database, a well-known face database, to perform our experiments. BioID contains 1521 front–view gray level face images with a resolution of 384×286 pixels. The database features a large variety of illumination and face sizes, and the background of images is very complex. This database is believed to be more difficult than other commonly-used head-and-shoulder face database without complex background, e.g. the extended M2VTS database.

The essence of this paper is to propose a face detection method which adopts a multiple-template strategy so that the detection speed can be considerably improved. Therefore, we compare the face detection accuracies and speeds of two methods (single-template boosting and multiple-template boosting) on BioID, both methods use the LIS feature and the proposed boosting algorithm to train the cascaded detector. With the single template, images should be rotated several times in order to detect different slanted faces; however, with multiple templates no image rotation is required. In our experiments, different parameters are arranged to investigate the performance of the proposed algorithm. Table 1 shows the detection results where Min_R is the minimal rotation angle, Max_R is the maximal rotation angle, R_step is the increment angle of each rotation, and the image scaling ration D is set to 0.85. The experiments are performed on a PC with windows 2000/XP, Pentium IV 1.6 GHz CPU and 1G main memory. The experimental results clear show that the proposed algorithm can achieve a good face detection rate (≒ 90%) and the multiple-template strategy indeed considerably accelerate the detection speed.

VII. Conclusion

This paper proposes a novel face detection algorithm which extracts a local image structure (LIS) feature and adopts a boosting approach to construct a cascaded face detector. Due to the locality of LIS, the extracted feature is not only robust to lighting variation but also is invariant to small degrees of rotation. With this robust property a multiple-template cascaded detector has been developed which can avoid rotating the image and also keep the ability to detect slanted faces. Because the multiple templates are constructed in the initialization stage, the proposed algorithm

can be executed very fast. Experiments have shown the efficiency of this method.

Since LIS is invariant only under the rotation of ±22.5°, it is important to extend the proposed method so that a large rotated face can also be correctly detected. This will be our future research direction.

ACKNOWLEDGMENT

This paper is the research result sponsored by the MOEA project **"**Construction of Vision-Based Intelligent Environment (VBIE)**",** the project code is 96-EC-17-A-02-S1-032.

REFERENCES

[1] K.K. Sung and T. Poggio, "Example-Based Learning for View-Based Human Face Detection", in Proceedings IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 20(1), pp. 39-51, 1998.

[2] M.H. Yang, N. Ahuja and D. Kriegman, "Face Detection Using a Mixture of Factor Analyzers", In Proceedings of IEEE International Conference on Image Processing, Vol. 3, pp. 612-616, 1999.

[3] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascaded of Simple Features", in Proceedings IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 511-518, 2001.

[4] Yea-Shuan Huang, Hao-ying Cheng, Po-Feng Cheng and Cheng-Yuan Tang, " Face Detection with High Precision Based on Radial-Symmetry Transform and Eye-Pair Checking", IEEE Conference on video and signal based surveillance, 2006.

[5] B. Frba and A. Ernst, "Face Detection with the Modified Census Transform", Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition (AFGR), Seoul, 2004, pp. 91-96.

	I		I	
I		I		I
	I	X	I	
I		I		I
	I		I	

Figure 1: One example of the interestedly selected pixel set Φ(X), where X denotes a being processed pixel and the pixels marked I denotes the selected interested neighbor pixels.

P_3	P_2	P_1
P_4	X	P_0
P_5	P_6	P_7

Figure 2: One example of the interestedly selected pixel set Φ(X), where 8 neighbors of X, P_0, P_1, P_2, P_3, P_4, P_5, P_6 and P_7, are chosen to construct the local image structure of pixel X.

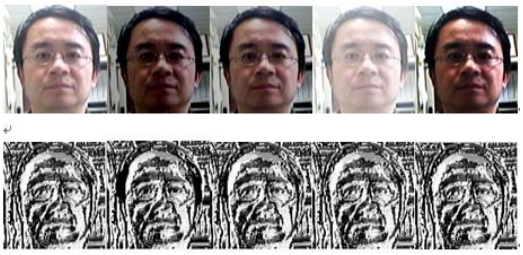

Figure 3: Images with different illumination are shown in the first row, and the corresponding images of derived local structures are shown in the second row.

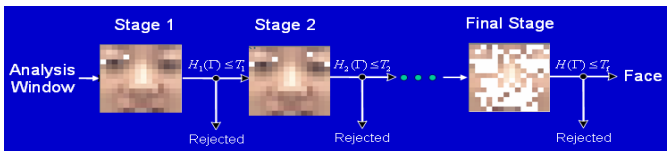

Figure 1: A cascaded face detector

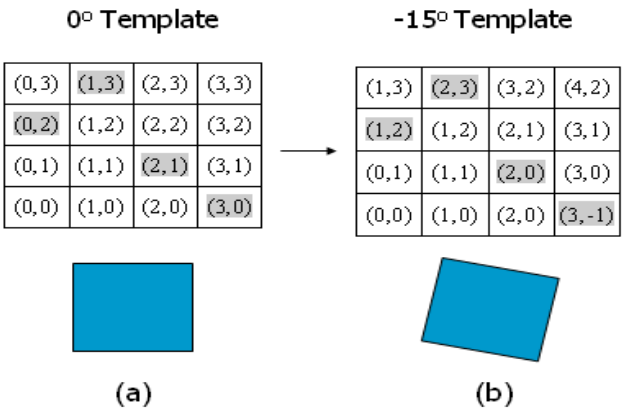

Figure 5: Two 4x4 illustrated templates. (a) the template of 0°-rotated faces, and (b) the template of 15°-rotated faces.

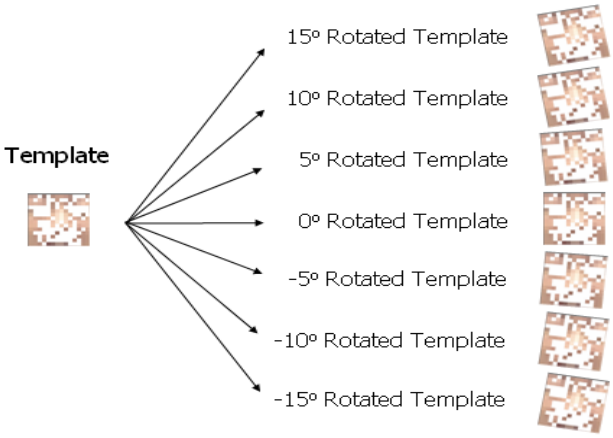

Figure 6: Multiple templates of which each corresponds to detect a range of 5° rotated faces.

Table 1: The detection results of single-template and multiple-template face detectors, where Min_R is the minimal rotation angle, Max_R is the maximal rotation angle, R_step is the increment angle.

method	Min-R	Max-R	R-step	Accuracy (%)	Speed (ms/image)
One template	-0°	0°	5°	84.0	180
Multiple templates	-0°	0°	5°	83.6	75
One template	-10°	10°	10°	86.5	223
Multiple templates	-10°	10°	10°	86.4	93
One template	-10°	10°	5°	89.4	230
Multiple templates	-10°	10°	5°	89.3	109

ISBN: 978-988-98671-8-8