Postprint: Improved Random Subspace LDA Combined with Multi-Patch Ensemble Learning for Robust Face Recognition Algorithm

Authors: Feng Daigao, Zhang Youjun

Date: 2018-05-24T00:00:00+00:00

Abstract

To address the negative impact of makeup on face recognition accuracy, an improved robust face recognition algorithm based on patch ensemble learning is proposed. First, each face image is embedded into patches and described by a set of feature descriptors, namely Local Gradient Gabor Pattern (LGP), Histogram of Gabor Space Order-Ratio Measurement (HGSFRM), and Densely Sampled Local Multi-value Pattern (DSLMP). Then, an improved Stochastic Subspace Linear Discriminant Analysis (SRS-LDA) method is employed to sample patches and establish multiple common subspaces between pre-makeup and post-makeup images for ensemble learning. Finally, collaborative and sparse representation classifiers are utilized to compare the feature vectors in these subspaces, while fusing the obtained scores through a sum rule. The proposed algorithm was evaluated and analyzed on multiple makeup datasets, and the results demonstrate that it achieves higher recognition accuracy compared to other algorithms specifically designed for post-makeup face recognition.

Full Text

Preamble

Robust Face Recognition Algorithm Based on Multiple Patch Integration Learning and Improved Random Subspace LDA

Feng Daigao, Zhang Youjun
(College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China)

Abstract: To address the negative effects of makeup on face recognition accuracy, this paper presents an improved robust face recognition algorithm based on patch integration learning. First, each face image is tessellated into patches, with each patch represented by a set of feature descriptors: Local Gabor Pattern (LGP), Histogram of Gabor Space Fixed Ratio Measures (HGS-FRM), and Densely Sampled Local Multi-valued Pattern (DSLMP). Then, an improved Random Subspace Linear Discriminant Analysis (SRS-LDA) method is used to sample patches and construct multiple common subspaces between before-makeup and after-makeup facial images for ensemble learning. Finally, Collaborative-based and Sparse-based Representation Classifiers compare feature vectors in these subspaces, with the resulting scores combined via sum-rule.

The proposed face matching algorithm is evaluated on multiple makeup datasets. Results demonstrate that this algorithm achieves higher recognition accuracy than other algorithms specifically designed for face recognition under makeup conditions.

Keywords: face recognition; robust evaluation; descriptor algorithm; patch integration; improved random subspace

0 Introduction

Automatic face recognition has been widely deployed in applications such as personal authentication, video surveillance, and human-computer interaction. These systems extract discriminative features from input facial images and compare them against stored templates in a database [1]. With advances in robust feature representation and matching technologies, the accuracy of face recognition systems has continuously improved [2], evidenced by significantly reduced error rates on public benchmark databases.

Nevertheless, numerous challenges persist in face recognition, particularly for heterogeneous face recognition. Facial makeup represents a very common form of facial heterogeneity, and recognition accuracy under makeup conditions is critically important. [Figure 1: see original paper] illustrates examples of makeup application effects. However, recent studies reveal substantial accuracy degradation for makeup-altered faces. According to [3], the recognition accuracy of both commercial and academic face recognition methods decreases dramatically due to makeup application, which can affect many existing matching procedures that rely on contrast and texture information cues. Moreover, recognizing faces with multiple combined cosmetic products poses greater challenges than recognizing faces with a single cosmetic product. Addressing the recognition problem under mixed makeup heterogeneity is key to developing robust face recognition systems.

To date, limited technical literature addresses the challenges introduced by makeup variations. Reference [4] utilized Canonical Correlation Analysis (CCA) together with Support Vector Machine (SVM) classifiers to improve matching between before-makeup and after-makeup images. To minimize the gap between matching images, [5] investigated mappings between features extracted from before-makeup and after-makeup images, studying these mappings using improved CCA and Partial Least Squares (PLS) methods. Although mapping-based approaches demonstrate effectiveness, they suffer from two limitations. First, the mapping between before-makeup and after-makeup facial images may be complex, spatially varying, and nonlinear; thus, investigating a single mapping is insufficient to describe the complex relationship between before-makeup and after-makeup samples. Second, CCA and PLS methods tend to overfit training data, yielding poor generalization to unseen subjects. References [6,7] proposed face detection methods based on integrated scoring of global and local information, which achieve high recognition accuracy on small sample sets but require further investigation for expanded sample ranges.

Therefore, this paper proposes a robust face recognition algorithm based on patch subspace ensemble learning to address heterogeneous face image recognition after makeup application. The method tessellates face images into patch sets, extracts feature vectors using multiple modalities, and obtains recognition results through classification comparators and sum-rule fusion. Simultaneously, overfitting is avoided by randomly selecting different patches as inputs for each subspace-based classifier. The algorithm is evaluated on multiple makeup datasets, with results demonstrating its effectiveness.

1 Robust Face Recognition Algorithm Based on Multi-Patch Ensemble Learning

The flowchart of the proposed multi-patch ensemble learning method for makeup-invariant face recognition is shown in [Figure 2: see original paper]. For a given facial image, the image is first tessellated into patches. Three feature descriptors are then applied to each patch: Local Gabor Pattern (LGP) [8], Gabor Space Fixed Ratio Measures (HGSFRM) [9], and Densely Sampled Local Multi-valued Pattern (DSLMP) [13]. These descriptors capture both global and local information. Next, a weight learning scheme based on Fisher' s separation criterion [10] ranks the importance of each patch. Then, based on patch-related weights, multiple patch sets are selected using a semi-random sampling method to create subspaces. Within these subspaces, Collaborative-based Representation Classifiers (CRC) [11] and Sparse-based Representation Classifiers (SRC) [12] are used to build ensemble classifiers that score the subspace patches. Finally, the scores generated by the classifiers are fused using sum-rule, which takes a weighted average of scores from multiple modalities. Through these steps, recognition of a facial image is completed.

This method provides an integrated framework for face matching that is robust to makeup application. The framework uses multiple subspaces corresponding to three different feature descriptors and multiple image patches. Sparse and collaborative classifiers are jointly applied in these subspaces, with selection guided by weight information from each patch rather than pure random sampling.

Descriptor selection is crucial in face recognition algorithms. LGP, HGSFRM, and DSLMP descriptors represent patches in each facial image. Mathematically, Gabor filters are described as [9]:

$$G_{\mu,v}(z) = I(z) * \varphi_{\mu,v}(z)$$

where z represents pixel location, μ and v denote orientation and scale of the Gabor filter, respectively. The complex Gabor response has two components: real part $r_{\mu,v}(z)$ and imaginary part $i_{\mu,v}(z)$. Therefore, Gabor magnitude $A_{\mu,v}(z)$ and phase $\theta_{\mu,v}(z)$ are calculated as:

$$A_{\mu,v}(z) = \sqrt{r^2_{\mu,v}(z) + i^2_{\mu,v}(z)}$$

$$\theta_{\mu,v}(z) = \arctan\left(\frac{i_{\mu,v}(z)}{r_{\mu,v}(z)}\right)$$

Here, $k_v = k_{\max}/s^v$ and $\phi_\mu = \pi\mu/8$, where $k_{\max}$ is the maximum frequency and s is the spacing factor between kernels in the frequency domain. The Gabor kernel function performs convolution on the input image to obtain the Gabor response.

1.1 Local Gabor Pattern (LGP)

To encode Gabor gradient responses, the gradient descriptor is defined as [10]:

$$\xi_v(z) = \arctan\left(\frac{c \cdot h_v(z)}{\beta + \lambda \cdot M_h(z)}\right)$$

where $h_v(z)$ and $M_h(z)$ represent image gradients calculated along vertical and horizontal directions, respectively. These two directions are orthogonal. The arctan function with parameters β and λ prevents input from increasing or decreasing too rapidly. Let x_0 denote the intensity value of the center pixel of a rectangle surrounded by neighboring points sampled from x_1 to x_{R-1}, where R is the neighborhood size. Its gradient is calculated as:

$$\gamma = \sum_{i=0}^{R-1} (x_i - x_{i+4}) \bmod R$$

where mod denotes the modulus operator. In our implementation, we use $R = 6$, $\beta = 8$, and $\lambda = 1 \times 10^{-6}$.

1.2 Histogram of Gabor Space Fixed Ratio Measures (HGSFRM)

To encode phase responses, we use Ordinal Measures (OM) [11]. OM compares two different regions to determine which has a larger value (e.g., mean). We use Multi-Lobe Differential Filters (MLDF) [12] to extract ordinal features. Mathematically, MLDF is expressed as:

$$\mathrm{MLDF}(z) = C_p \sum_{i=1}^{N_p} \exp\left(-\frac{\|z-\mu_{p,i}\|^2}{2\delta_{p,i}^2}\right) - C_n \sum_{j=1}^{N_n} \exp\left(-\frac{\|z-\mu_{n,j}\|^2}{2\delta_{n,j}^2}\right)$$

where z is the pixel location, $\mu_{p,i}$ and $\mu_{n,j}$ are the center positions of 2D Gaussian filters, and $\delta_{p,i}$ and $\delta_{n,j}$ are their scales. N_p is the number of positive lobes, N_n is the number of negative lobes, and C_p and C_n are constant coefficients to ensure MLDF output $\in [-1, 1]$. MLDF is a type of differential bandpass filter.

1.3 Densely Sampled Local Multi-valued Pattern (DSLMP)

To generate DSLMP features, each LBP-encoded image is divided into non-overlapping patches, with histogram information extracted from each patch [13]. The number of patches per image is 256, with each patch sized at 16×16. All three feature descriptors are summarized in .

1.4.1 Training Phase

Weight Learning: Before sampling patches, we assign weights to each extracted patch and rank them based on these weights. Weights are calculated based on Fisher' s separation criterion. Our hypothesis is that different facial regions may have varying impacts on face recognition under makeup because makeup information is not uniformly distributed. For each patch p and its associated feature vector $Y(p)$, the within-class mean distance is calculated as:

$$D_1(p) = \frac{1}{c} \sum_{i=1}^{c} \frac{1}{l_i} \sum_{j,k=1}^{l_i} \phi(Y_{ij}(p), Y_{ik}(p))$$

where ϕ denotes the chi-square distance between two feature vectors, c is the number of classes, and l_i is the number of samples in class i. The within-class distance variance is calculated as:

$$\mathrm{VAR}_1(p) = \frac{1}{c} \sum_{i=1}^{c} \frac{1}{l_i} \sum_{j,k=1}^{l_i} (\phi(Y_{ij}(p), Y_{ik}(p)) - D_1(p))^2$$

The between-class mean distance is calculated as:

$$D_2(p) = \frac{1}{\sum_{i=1}^{c-1}(c-i)} \sum_{i=1}^{c-1} \sum_{q=i+1}^{c} \frac{1}{l_i l_q} \sum_{j=1}^{l_i} \sum_{k=1}^{l_q} \phi(Y_{ij}(p), Y_{qk}(p))$$

The between-class distance variance is calculated as:

$$\mathrm{VAR}_2(p) = \frac{1}{\sum_{i=1}^{c-1}(c-i)} \sum_{i=1}^{c-1} \sum_{q=i+1}^{c} \frac{1}{l_i l_q} \sum_{j=1}^{l_i} \sum_{k=1}^{l_q} (\phi(Y_{ij}(p), Y_{qk}(p)) - D_2(p))^2$$

The learned weight for each patch p is calculated as:

$$W(p) = \frac{D_2(p) - D_1(p)}{\mathrm{VAR}_1(p) + \mathrm{VAR}_2(p)}$$

These learned weights determine the importance of different patches for recognition. Patches are then stored in descending order of weight to guide the sampling process for each feature vector.

Patch Sampling: Multiple subspaces are created to generate an ensemble of classifiers corresponding to each descriptor. Each subspace is constructed based on semi-random sampling of weighted patches. "Semi-random" means the probability of selecting a patch is related to its weight. For creating K subspaces, we sample α patches for a specific descriptor from P patches, where $\alpha \ll P$. By concatenating α feature vectors into a single vector representation, we obtain $Y_{B,ij} \in \mathbb{R}^{D \times d}$ and $Y_{A,ij} \in \mathbb{R}^{D \times d}$, where d is the feature dimension per patch (see). To ensure comprehensive patch utilization, 60% of α can be selected from the first half of patches and 40% from the remaining half. The values of α and K are determined empirically; recommended values are provided in later sections to reduce computational cost.

Subspace Construction [14]: Since the dimension of $Y_{B,ij}$ and $Y_{A,ij}$ is typically higher than the number of samples, feature dimensionality reduction is performed to reduce computation time and avoid small sample size problems. A common method for reducing feature dimensionality is Principal Component Analysis (PCA). PCA finds a projection space that best reconstructs the original vectors.

To find this subspace, mean vectors η_B and η_A are calculated from feature vectors sampled from before-makeup and after-makeup images, respectively:

$$\eta_B = \frac{1}{N} \sum_{i=1}^{c} \sum_{j=1}^{n_i} Y_{B,ij}$$

$$\eta_A = \frac{1}{M}\sum_{i=1}^{c}\sum_{j=1}^{m_i} Y_{A,ij}$$

where $N = \sum_{i=1}^{c} n_i$ and $M = \sum_{i=1}^{c} m_i$. The total covariance matrix is calculated as:

$$S_T = \frac{1}{N+M}\sum_{i=1}^{c}\sum_{j=1}^{n_i+m_i}(Y_{ij}-\eta)(Y_{ij}-\eta)^T$$

where η is the overall mean. Eigenvectors E and eigenvalues W can be computed from the covariance matrix S_T via $EW = S_T EW\lambda$. After generating E, before-makeup and after-makeup samples can be projected into the new subspace:

$$y_{B,ij} = E^T(Y_{B,ij} - \eta_B)$$

$$y_{A,ij} = E^T(Y_{A,ij} - \eta_A)$$

The feature vectors after PCA are used to compute between-class and within-class scatter matrices, ensuring that the learned feature representation is less sensitive to makeup variations.

When constructing the k-th subspace for a feature descriptor, feature vectors from before-makeup and after-makeup images are used together. This approach serves two purposes: (a) using both before-makeup and after-makeup patches jointly in the weight learning process, and (b) generating corresponding subspaces. The before-makeup and after-makeup feature vectors are used to learn within-class and between-class scatter matrices for LDA. These repeatedly constructed subspaces for each descriptor are also called common subspaces.

1.4.2 Testing Phase

In the testing phase, after-makeup images from the isolated test set are used as probes and compared against before-makeup images serving as gallery images. Let Y_{ij} denote a set of feature vectors extracted from image I_{ij}, where I_{ij} is either a before-makeup or after-makeup image from the test sample. The same set of patches is selected from Y_{ij} and concatenated into a feature vector $y_{B,ij}$ or $y_{A,ij}$. The positions and order of patches are identical in both training and test sets.

Subspace Projection: For each derived subspace $k \in \{1, 2, \dots, K\}$, the test feature vectors are projected:

$$y_{B,ij}^{(k)} = F_k^T(y_{B,ij} - \mu_B)$$

$$y_{A,ij}^{(k)} = F_k^T(y_{A,ij} - \mu_A)$$

where μ_B or μ_A are the final projection feature vectors for before-makeup and after-makeup test samples, respectively. If makeup information is unknown, a makeup detection scheme [15] can be used for discrimination. The total mean vector μ can be used for projection, resulting in only a slight reduction in matching accuracy (<1% recognition rate).

Next, the between-class scatter matrix is calculated as:

$$S_B^{(k)} = \sum_{i=1}^{c}(\eta_i^{(k)} - \eta^{(k)})(\eta_i^{(k)} - \eta^{(k)})^T$$

where $\eta_i^{(k)}$ is the mean class vector for the i-th subject computed from projected vectors. The within-class scatter matrix is calculated as:

$$S_W^{(k)} = \sum_{i=1}^{c}\sum_{j=1}^{n_i}(y_{B,ij}^{(k)} - \eta_i^{(k)})(y_{B,ij}^{(k)} - \eta_i^{(k)})^T + \sum_{i=1}^{c}\sum_{j=1}^{m_i}(y_{A,ij}^{(k)} - \eta_i^{(k)})(y_{A,ij}^{(k)} - \eta_i^{(k)})^T$$

LDA aims to find the optimal projection ω_k that maximizes the ratio of the determinant of the between-class scatter matrix to the determinant of the within-class scatter matrix. The optimization problem is defined as:

$$\omega_k = \arg\max \frac{|\omega_k^T S_B^{(k)} \omega_k|}{|\omega_k^T S_W^{(k)} \omega_k|}$$

This is equivalent to solving the generalized eigenvalue problem $S_B^{(k)}\psi_k = \lambda_k S_W^{(k)}\psi_k$. For each subspace, the training phase outputs F_k, μ_B, μ_A, and ω_k.

SRC and CRC Classification: As previously described, before-makeup samples are used as gallery G and after-makeup samples as probe P. Let $Y_{B,ij}$ denote gallery feature vectors from before-makeup samples, where c is the number of subjects in the gallery and n_i is the number of samples for the i-th subject. Let $Y_{A,ij}$ denote probe samples from after-makeup images. Distance scores are computed by replacing the gallery with the probe to obtain similarity scores between before-makeup and after-makeup samples. This is calculated using sparse and collaborative representation principles [16].

SRC and CRC each have their own advantages and provide complementary information for classification. Therefore, we develop two classifiers for each subspace: one based on SRC and another based on CRC, combining their outputs for classification.

2 Experimental Simulation and Analysis

To verify the effectiveness of the proposed face recognition algorithm based on multi-patch ensemble learning for recognizing before-makeup and after-makeup face samples, relevant facial data from the YMU dataset is used for the following experiments.

2.1 Makeup Dataset

Experiments are conducted using the YMU dataset. To reduce variations caused by scaling and pose, faces are geometrically normalized using affine transformations based on eye landmarks. All normalized face images are cropped and resized to dimensions of 128×128 and converted from RGB to grayscale.

For training the face matching procedure, another dataset is assembled. This training dataset consists of a subset of female subjects from the FAM dataset, Asian Female Makeup dataset, and entire MIAA dataset. The total number of samples in the training dataset is 796, corresponding to 398 subjects. Each subject has one before-makeup and one after-makeup sample. This dataset is referred to as "Makeup Training Data (MTD)." Due to the limited number of samples per subject in MTD, facial symmetry is exploited to generate mirrored face samples. This helps construct more robust subspaces. It should be noted that training with MTD does not cause over-optimization of experimental results because it is a distinct database from YMU with no overlapping subjects between training and testing.

The following parameter values are used in experiments: number of subspaces $K = 120$, $\lambda = 0.15$, $\lambda_1 = 0.1$, $\alpha = 180$ for each descriptor; $\alpha = 45$ for HGSFRM and LGP; and $\alpha = 80$ for LBP. The dimension of SRS-LDA feature vectors is 256.

2.2 Experiments on YMU Dataset

To evaluate the performance of the proposed face matching procedure, three categories of recognition experiments are conducted:

a) Comparing B vs. B ($B \to B$): Both images being compared are before-makeup images.

b) Comparing A vs. A ($A \to A$): Both images being compared are after-makeup images.

c) Comparing A vs. B ($A \to B$): One image is after-makeup while the other is before-makeup.

The Equal Error Rates (EER) for different matching scenarios considered in the YMU dataset are summarized in [Figure 4: see original paper]. COTS-1, COTS-2, and COTS-3 are commercial face recognition software algorithms [15] representing state-of-the-art performance in face recognition tasks. These evaluated algorithms exhibit significantly higher EER when matching after-makeup against before-makeup samples. For the matching scenario $A \to B$, we substantially reduce EER from nearly 20% to 6.36%, achieving the lowest equal error rate.

Compared to random sampling in $A \to B$ matching, the semi-random sampling scheme generated by weight learning improves GAR by approximately 4%. To verify the stability of SRS-LDA, we repeat the experiments 3.1 and report the distribution of EER and GAR. As shown in [Figure 6: see original paper], at 0.1% FAR, EER values range from 6.13% to 6.69%, and GAR varies from 69.18% to 70.6%. These results demonstrate that the semi-random scheme does not cause instability in the ensemble algorithm.

2.3 Algorithm Fusion

We also consider fusing the proposed method with COTS to further improve matching performance. Individual matching scores generated by different matching procedures are normalized using min-max rule and simple sum-rule. As shown in [Figure 5: see original paper], the fused matching procedure significantly improves face matching performance in terms of both EER and Genuine Accept Rate (GAR). Clearly, the proposed method and COTS provide complementary information. COTS-1, COTS-2, and COTS-3 achieve EERs of 13.04%, 7.79%, and 9.33%, respectively. The GARs at 0.1% FAR are 49.36%, 77.15%, and 58.97%, respectively. The proposed method achieves an EER of 6.41% and GAR of 69.89%. When fusing our method with COTS-2 and COTS-3, we obtain the best results: 84.11% GAR and 5.08% EER, which outperforms the single-algorithm results of COTS-2 and COTS-3. This demonstrates that the proposed method can effectively improve COTS algorithms.

It should be noted that we focus only on robust feature extraction and matching, which is opposite to end-to-end COTS matching procedures that benefit from years of research and may incorporate advanced pre-processing and post-processing. Nevertheless, the proposed method remains highly competitive in recognition rate and fusion algorithm improvement.

2.4 Individual Descriptor Analysis

Three types of feature descriptors (LGP, HGSFRM, and DSLMP) are used in the multi-patch ensemble learning algorithm. However, the performance of individual descriptors varies. If only descriptor-based methods are used, matching performance for $A \to B$ is very low. As shown in [Figure 7: see original paper], LGP, HGSFRM, and DSLMP achieve EERs of 20.79%, 20.54%, and 19.93%, respectively. This indicates that the ensemble learning framework further improves algorithm performance. After applying the SRS-LDA method, matching performance increases significantly. Extensive recognition experiments demonstrate that the fusion of LGP, HGSFRM, and DSLMP yields the best results, confirming the rationale for using these features in the proposed framework. HGSFRM achieves the best overall performance among individual descriptors.

2.5 Number of Subspaces

An important parameter in SRS-LDA is the number of subspaces K used. To analyze the convergence of the proposed method, we construct an experiment gradually increasing the number of subspaces and computing the corresponding Rank-1 accuracy. As shown in [Figure 8: see original paper], the performance of the proposed method initially increases with the number of subspaces and then stabilizes. HGSFRM peaks at 60 iterations, while DSLMP and LGP peak after 50 and 80 iterations, respectively. In our experiments, the number of subspaces is set to 95, where algorithm performance is stable.

2.6 Computational Complexity

Since the training phase is performed offline, only feature extraction and classification of test samples affect computation time during real-time operation. The feature extraction process involves simple linear operations with limited resource consumption. Both DSLMP and CRC use pure random sampling. Computation times for different methods are shown in . The SRS-LDA computation time is lower than other methods because although the random subspaces generated by the ensemble learning algorithm are independent, they can be processed in parallel, effectively meeting the requirements of practical face recognition systems.

2.7 Large-Scale Recognition Experiment

To demonstrate practical face retrieval, we increase the number of before-makeup samples using image subsets from FRGC and MIW datasets. In this experiment, after-makeup samples from the YMU dataset are used as probes. A subset of 10,000 photo images is selected from the FRGC dataset (10K: 4574 females + 5426 males). Additionally, 112 face images from the MIW dataset are added to the gallery, resulting in a total of 10,414 images (302 before-makeup YMU images, 10K FRGC images, and 112 MIW images). Face matching accuracy curves are shown in [Figure 9: see original paper]. Experimental results demonstrate only a slight decrease in large-scale sample recognition accuracy when using the exact same algorithm, indicating that the proposed method can be applied to large datasets.

3 Conclusion

This paper proposes a method for matching after-makeup images against before-makeup images. The method is based on patch ensemble learning, generating multiple subspaces for three different descriptors. Fisher' s separation criterion is then used to guide the patch sampling process before subspace generation. Within the generated semi-random subspaces, both SRC and CRC are used for classification. The final output consists of matching scores from individual descriptors. Experimental results on the YMU dataset and larger-scale datasets demonstrate the effectiveness of the proposed method. The proposed approach can be fused with COTS algorithms to further improve matching accuracy.

References

[1] Liu Feng, Meng Fanrong, Liang Zhizhen. Face recognition based on first and two order information image representation [J]. Application Research of Computers, 2017, 34 (2): 603-606.

[2] Zou Guofeng, Fu Guixia, Li Haitao, et al. Overview of multi pose face recognition [J]. Pattern Recognition and Artificial Intelligence, 2015, 28 (7): 613-625.

[3] Ueda S, Koyama T. Influence of make-up on facial recognition [J]. Perception, 2010, 39 (2): 260-264.

[4] Chen Xi, Zhang Ge. Biometric extraction algorithm based on non Gauss two-dimensional gabor filter [J]. Application Research of Computers, 2017, 53 (18): 170-175.

[5] Chen Jing, Zhang su. Robust face recognition method based on entropy weighted Gabor feature [J]. Control Engineering, 2017, 24 (8): 1623-1629.

[6] Fan Shouke, Zhu Ming. Illumination robust face recognition based on weighted block sparse representation [J]. Application Research of Computers, 2015 (5): 1563-1567.

[7] Su Yu, Shan Shi Guang, Chen Xilin, et al. Face recognition based on global and local feature integration [J]. Journal of Software, 2010, 21 (8): 1849-1862.

[8] Liu B, Yan X U, Qiang M I, et al. Face recognition based on LBP feature and improved Fisher criterion [J]. Computer Engineering & Applications, 2012, 48 (18): 170-175.

[9] Min Yongzhi, Cheng Tiandong, Yin Chao, et al. Texture image segmentation based on GLCM and adaptive Gabor filter banks [J]. Computer Engineering, 2017, 43 (1): 280-286.

[10] Xiang Zheng, Tan Hengliang, Ma Zhengming. Performance Comparison of Improved HOG, Gabor and LBP [J]. Journal of Computer Aided Design and Graphics, 2012, 24 (6): 787-792.

[11] Sun Z, Tan T. Ordinal Measures for Iris Recognition [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2009, 31 (12): 2211-2226.

[12] Nguyen H V, Bai Li, Shen Linlin. Local gabor binary pattern whitened PCA: a novel approach for face recognition from single image per person [C]// Advances in Biometrics. Berlin: Springer, 2009: 269-278.

[13] Shao Zhuhong, Ouyang Junlin, Liao Fan, et al. Robust color face recognition algorithm based on local features and integrated learning [J]. Journal of Southeast University: Natural Science Edition, 2015, 45 (2): 251-255.

[14] Juefeixu F, Pal D K, Singh K, et al. A preliminary investigation on the sensitivity of COTS face recognition systems to forensic analyst-style face processing for occlusions [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2015: 25-33.

[15] Proença H, Neves J C. Deep-PRWIS: periocular recognition without the iris and sclera using deep learning frameworks [J]. IEEE Trans on Information Forensics & Security, 2017, 34 (9): 34-42.

[16] Huang Chao, Gao Li Ping. Facial expression independent discriminating cooperative expression recognition algorithm [J]. Application Research of Computers, 2017, 34 (9): 2858-2862.

Note: Figure translations are in progress. See original paper for figures.