Journal of Computers Vol. 30 No. 5, 2019, pp. 88-97
doi:10.3966/199115992019103005007

A Convenient Classification System for Face Orientations Recognition Based on Support Vector Machine

Yang Liu[1*], Yong-kui Shi[2], Ming-wei Xu[2]

[1] College of Management, Xi'an JiaoTong University, Xi'an 710049, Shanxi, China
fanxipingly@163.com

[2] College of Mining and Safety Engineering, Shandong University of Science and Technology, Tsingtao 266590, Shandong, China
shiyongkui@163.com, 903938350@qq.com

Received 2 November 2017; Revised 5 March 2018; Accepted 3 May 2018

Abstract. Face orientations recognition is the premise of face recognition. Many approaches have been proposed to recognize the human face orientations, but these methods usually rely on the assumption of a frontal view of human face. Besides this, the costs of different types of misclassifications are treated equally, without considering the number of orientations. This may lead to the inefficiency of face recognition system. Aiming at these problems, this paper aims to design a convenient classification system for recognizing face orientations. Firstly, the classification model is established based on PSO-SVM with a hybrid kernel. Secondly, maturity metric (M) is structured with precision metric (P) and cost metric (E) according to the classification results. It is used to judge whether the classification model meets the performance requirement. The experimental results show that the proposed classification system produces P=0.9815 and E=0.011, and the value of M equals 83.33. It indicates that the proposed classification system can be used for face orientations recognition in advance, which will contribute to the success of face recognition.

Keywords: classification system, face orientations recognition, kernel function, maturity metric

1 Introduction

Face recognition is a biometric identification technology based on facial features. There are a multitude of researchers devoting to face recognition due to its potential applications on pattern recognition, computer vision, and artificial intelligence, etc [1-3]. Quite a number of innovative face recognition methods were proposed and nowadays it has been applied in many fields. For example, Jack Ma demonstrated Angela Merkel and Ma Kai how to pay with his face at Centrum der Bro and Informations technik (CeBIT), held in March, 2015 in Hannover. Tim Cook tried to unlock iphone X by face recognition technology during the press conference, but his first try was a failure. Fortunately, he succeeded after a second attempt. It shows that face recognition has made significant space for development, but it still needs more improvements in the both recognition efficiency and reliability.

Face includes various orientations, which adds to the complexity of face recognition. The orientations of human face which are correctly determined in advance is the premise of face recognition. So in the face detection, face orientations can be divided into different classes. However, face orientations recognition (FOR) has been neglected by researchers for a long time. Only a small amount of studies have been conducted on the recognition of face orientations at present. And there are still some limitations in the existing methods: (1) the accuracy of FOR methods can not enjoy a satisfactory degree. (2) the costs of different types of misclassifications are treated equally. Its underwhelming status leads to the inefficiency of face recognition system. Aiming at these problems, this paper introduces machine

* Corresponding Author

learning algorithms for the recognition of face orientations since the nature of recognizing face orientations is a classification problem. The advantages of machine learning algorithms used in the classification have been proved effective. Therefore, this article applies them for the recognition of face orientations. Due to the highest accuracy and the least elapsed time, SVM optimized by particle swarm optimization (PSO) based on hybrid kernel is picked out to design a reliable classification system. If face orientations are captured correctly in advance, it will contribute to the success of face recognition.

The remainder of this study is organized as follows: Section 2 reviews the related literatures and works, and analyzes the feasibility of solving the existing problems. Three initial models are established in Section 3. Section 4 optimizes the initial models and proposes the optimal model. The classification system and case analysis are given in Section 5. Section 6 summarizes the conclusions.

2 Overview and Feasibility Analysis

The majority of the existing literatures about face recognition generally rely on the assumption of a frontal view of human face. In other words, these methods downplay the importance of face orientations. Only a small amount of studies have been conducted to recognize the non-frontal face orientations at the present stage. Zhou et al. [4] proposed a framework of orientation analysis for rotated human face detection. The experimental results showed the effectiveness of their framework. Yan et al. [5] proposed a kind of method on face orientation by combining Haar-like features and learning vector quantization (LVQ) classifier to detect face orientations. According to the experimental results, the proposed technique on face orientation detection is a kind of promising technique. Gao and Xu [6] harbored the idea that combining multiple feature triangles with BP neural network (BPNN) could improve the accuracy of face orientations detection. Nakazawa et al. [7] utilized the changes in the dark areas of nostrils for the recognition of face orientations. In his test, a high success percentage could be obtained.

Although many studies achieved high recognition accuracy of at least 90.00%, there are still some limitations in the recognition process. The detection of face orientations is the premise of face recognition, but it fails to draw due attention. The existing ways may achieve the highest accuracy, but the total cost is not the smallest. The costs of different misclassifications are treated equally, without considering the number of face orientations. For example, an unhealthy patient who is diagnosed as healthy may be a fatal mistake while a healthy person diagnosed of cancer is regarded as a non-fatal error. It is obvious that the costs of misclassifications are variously different. Last but not the least, human face covers lots of feature points, like eyes, nose and mouth. The dimension of input features is too big to result in the low learning ability and the optimal solution can not be obtained.

In view of the problems that lie ahead, we aim to design a convenient classification system for recognizing face orientations. Referring to Yan et al.'s research results [5], we observe that face orientations are obviously different along with the changes of facial features described by eyes, called Eigeneyes. Especially, the distance between two eyes becomes larger when the orientation turns from left semi-profile or right semi-profile to frontal. Conversely, the distance becomes smaller when the orientation turns from frontal to left semi-profile or right semi-profile. Therefore, we will make best use of the Eigeneyes and the Eigenvalues of Eigeneyes are taken as the key feature point. An optimal model is attained by comparing the experimental results. The performance metric of maximizing the precision and minimizing the cost is defined. The classification system is finally established.

3 Initial Models Establishment

The most frequently used machine learning algorithms, i.e. LVQ, BPNN, and support vector machine (SVM), are introduced to establish three initial models and we compare their recognition results.

3.1 Feature Extraction

The images used in our experiment are derived from Olivetti Research Laboratory (ORL). They are divided into three orientations: left semi-profile, frontal, and right semi-profile. Some typical images are shown in Fig. 1.

Fig. 1. Facial grey images

The goal of feature extraction is to search the key feature points. Fig. 2 shows the results of Eigeneyes detection located by Sobel edge operator in a range of head rotation.

Fig. 2. Geometric features of Eigeneyes

3.2 Initial Models

The extracted Eigeneyes can be expressed by an array, and the Eigenvalues indicated by one submatrix differ by orders of magnitude. Before building the initial models, the extracted Eigenvalues are normalized as:

$$x_j = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \tag{1}$$

where $x_{\min}$ is the minimum value; $x_{\max}$ is the maximum value; x_i is the input value; x_j is the output value. The normalized Eigenvalues are taken as the input of initial models and the output is the face orientations.

LVQ model. LVQ is an effective learning algorithm that trains the competition layer under supervision [8]. It consists of input layer, competition layer, and output layer. The steps of LVQ algorithm are as described below.

Step 1: Initialize the weights ω_{ij} and the learning rate η between input layer and competition layer.

Step 2: Import the input vector $x=(x_1, x_2,\ldots, x_m)^T$ to input layer and calculate the distance d_i as:

$$d_i = \sqrt{\sum_{j=1}^{R}(x_j - \omega_{ij})^2}\ . \tag{2}$$

Step 3: Choose the nearest neuron in the competition layer from input vectors and the label of output layer is C_i with the smallest d_i.

Step 4: Record the labels of input vectors as C_x. If C_i equals C_x, then adjust the weights ω_{ij_new} as:

$$\omega_{ij\_new} = \omega_{ij\_old} + \eta(x - \omega_{ij\_old})\ . \tag{3}$$

If not, then update the weights ω_{ij_new} as:

$$\omega_{ij\_new} = \omega_{ij\_old} - \eta(x - \omega_{ij\_old})\ . \tag{4}$$

By using newlvq( ) function, a LVQ model (net=newlvq( )) is established with η=0.1 and the number of neurons in competition layer S=20.

BPNN model. BPNN is a multilayer feed-forward neural network which contains input layer, hidden layer, and output layer [9]. The steps of BPNN algorithm are as described below.

Step 1: Initialize the threshold a and b of hidden layer and output layer, the learning rate η, and the activation function f.

Step 2: Compute the hidden layer output H as:

$$H_j = f(\sum_{i=1}^{m} \omega_{ij} x_i - a_j)\,, \tag{5}$$

where ω_{ij} is the weights between input layer and hidden layer.

Step 3: Calculate the output O of layer output as:

$$O_k = \sum_{j=1}^{m} H_j \omega_{jk} - b_k\,, \tag{6}$$

where ω_{jk} is the weights between hidden layer and output layer.

Step 4: Calculate the error e as:

$$e_k = Y_k - O_k\,, \tag{7}$$

where Y_k is the desired output.

Step 5: Update the weights ω_{ij} and ω_{jk} as:

$$\omega_{ij} = \omega_{ij} + \eta H_j (1 - H_j) x(i) \sum_{k=1}^{m} \omega_{jk} e_k\,, \tag{8}$$

$$\omega_{jk} = \omega_{jk} + \eta H_j e_k\,. \tag{9}$$

Step 6: Update the node threshold a and b as:

$$a_j = a_j + \eta H_j (1 - H_j) x(i) \sum_{k=1}^{m} \omega_{jk} e_k\,, \tag{10}$$

$$b_j = b_j + e_k\,. \tag{11}$$

By using newff( ) function, a BPNN model (net=newff( )) is established which owns 17 neurons in the hidden layer. The transfer function in the hidden layer is tansig function.

SVM model. SVM was used to pattern classification by using the principles of structural risk minimization [10]. The steps of SVM algorithm are listed below.

Step 1: Assume the given training set T as:

$$T = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}\,. \tag{12}$$

Step 2: Choose the suitable kernel function $K(x, x_i)$ and build the optimization model as:

$$\min \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(x, x_i) - \sum_{j=1}^{m} \alpha_j\,, \tag{13}$$

then gain the optimization solution $\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_m^*)$.

Step 3: Compute the threshold b^* as:

$$b^* = y_i - \sum_{i=1}^{m} y_i \alpha_i^* K(x - x_i)\,. \tag{14}$$

Step 4: Structure the decision function $f(x)$ as:

$$f(x)=\operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i^* K(x-x_i)+b^*\right). \tag{15}$$

The purpose of kernel function is to process the computation of inner product in the high dimensional space with simple function in the low dimensional space [11]. By default, radical basis function (RBF) kernel is used to construct $K(x, x_i)$ as:

$$K(x,x_i)=\exp(-\gamma\|x-x_i\|^2), \tag{16}$$

where γ is the slope of RBF kernel and the decision function is considered as:

$$f(x)=\operatorname{sgn}\left(\sum_{i=1}^{m} W_i \exp(-\gamma\|x-x_i\|^2+b\right). \tag{17}$$

By using svmtrain( ) function in libsvm toolbox developed by Chih-Jen Lin, a SVM model (net=svmtrain( )) is established based on RBF kernel.

3.3 Results Analysis

30 sets of the extracted Eigenvalues are supposed to the input of LVQ model, BPNN model, and SVM model respectively in order to train themselves. The rest are regarded as the testing set. To be clear, the details of operating system, processor and software version are as follows: (1) operating system: Windows 7; (2) computer processor: Intel (R) Core (TM) i5-3230M CPU@2.60GHz 2.60GHz; (3) matlab version: R2013a. After training, the average recognition accuracy (ARA) and the average elapsed time (AET) are listed in Table 1.

Table 1. Initial comparison results

Initial models	ARA	AET
LVQ model	81.48%	18.23s
BPNN model	72.22%	9.25s
SVM model	83.33%	3.15s

Both ARA and AET of three initial models achieve very good level, especially SVM model, but there is still potential to lift their performances higher because the parameters in the algorithms are fixed.

4 Preliminary Models Establishment

4.1 Preliminary Models

The default parameters in the initial models are designated to the fixed values but not the optimal values. Therefore, it is necessary to search the optimal parameters. In this section, the initial models are optimized by K-fold cross validation (K-CV), genetic algorithm (GA), and PSO. Then we get the preliminary optimization models and pick out a relatively satisfying model to be further optimized among them.

CV-LVQ model. The number of neurons in the competition layer in the LVQ model is supposed to a fixed value. In this subsection, K-CV algorithm is applied to pick out the best number of neurons, which are these: (1) set the range of neuron number from 10 to 30; (2) search the average accuracy of validation set. The error between expectation and classification is regarded as fitness function; (3) attain the best number of neurons and substitute it into LVQ model.

GA-BPNN model. The weights and thresholds in the BPNN model are selected by random selection. Due to the global search performance of GA, it is applied to adjust the connected weights and thresholds in input layer and hidden layer, hidden layer and output layer. The error generated by BPNN model with local search ability in the training set is considered as the fitness function of GA.

PSO-SVM model. The parameters in the SVM model including penalty parameter (c) and kernel parameter (g) are also selected arbitrarily. The supposed best c and best g give rise to different accuracies of SVM model. Compared with GA, PSO algorithm has no selection, crossover, and mutation in the iterative optimization. The residual error of PSO-SVM model should be as small as possible.

4.2 Results Comparison

In order to verify the effective of the preliminary models, we apply them to detect face orientations again and compare the recognition results in Table 2.

Table 2. Preliminary results

Models		ARA			AET	
Initial	LVQ	BPNN	SVM	LVQ	BPNN	SVM
models	84.18%	72.22%	83.33%	18.23s	9.25s	3.15s
Preliminary	CV-LVQ	GA-BPNN	PSO-SVM	CV-LVQ	GA-BPNN	PSO-SVM
models	87.04%	81.48%	92.59%	2593.85s	3651.43s	34.12s

Table 2 shows that PSO-SVM model not only reaches much higher ARA than CV-LVQ model and GA-BPNN model (92.59%>87.04%>81.48%), but also wastes less AET (34.12<2593.85<3651.43). It is for this reason that we pick out PSO-SVM model as the further optimized model.

4.3 PSO-SVM Model Optimization

Kernel Function. What calls for special attention is that kernel function in the PSO-SVM model is assumed to RBF kernel. Besides RBF kernel, the kernels yet consist of linear kernel, polynomial kernel, and Sigmoid kernel.

(1) Linear kernel is the simplest kernel function. It is given by the inner product plus an optional constant q as:

$$K(x,x_i) = x^T x_i + q \; . \tag{18}$$

(2) Polynomial kernel is a non-stationary kernel function, and is given by:

$$K(x,x_i) = (\beta x^T x_i + r)^d \; , \tag{19}$$

where β is the slope; r is the constant term; d is the polynomial degree.

(3) Sigmoid kernel comes from the neural networks field, where the bipolar sigmoid function is often used as an activation function for artificial neurons. It is given as:

$$K(x,x_i) = \tanh(\upsilon x^T x_i + h) \; , \tag{20}$$

where υ is the slope; h is the intercept constant.

Above kernel functions fall into two categories: global kernel and local kernel. Global kernel has higher generalization performance and weaker learning ability while local kernel owns stronger learning ability and inferior generalization performance. Different kernel functions will produce different SVMs.

PSO-SVM Model based on Hybrid Kernel. RBF kernel owns weak generalization performance. It is not appropriate as the kernel of PSO-SVM model. Therefore, it is of great importance to structure a new kernel function with stronger generalization performance and learning ability. We establish four PSO-SVM models based on linear kernel, polynomial kernel, RBF kernel, and Sigmoid kernel and apply them to detect face orientations. Table 3 shows the ARA and AET of four PSO-SVM models.

Table 3. Different PSO-SVM models recognition results

Kernels	ARA	AET
Linear kernel	88.89%	34.59s
Polynomial kernel	92.59%	38.48s
RBF kernel	96.51%	34.12s
Sigmoid kernel	90.74%	39.25s

Table 3 suggests that polynomial kernel has an ARA advantage over Sigmoid kernel. RBF kernel outperforms linear kernel in terms of ARA. There are no significant differences in the AET although they are different. Another reason why we choose PSO-SVM model is that a hybrid kernel (K_{hybrid}) can be determined. The hybrid kernel has stronger generalization performance and learning ability as:

$$K_{hybrid} = \lambda_1 K_1 + \lambda_2 K_2 \,, \tag{21}$$

where K_1 is polynomial kernel and λ_1 is its kernel coefficient; K_2 is RBF kernel and λ_2 is its kernel coefficient; $\lambda_1>0$, $\lambda_2>0$, $\lambda_1+\lambda_2=1$.

5 Classification System Construction and Case Application

In section 4, an improved PSO-SVM model based on hybrid kernel (IPSO-SVM model) has been proposed. In this section, we devise a convenient classification system and apply it for the recognition of face orientations. The classification system consists of the IPSO-SVM model and its performance metric.

5.1 Maturity Metric Definition

Precision metric. Face orientations consist of left semi-profile, frontal, and right semi-profile. The number of three orientations is usually imbalanced, which makes the ARA lack the adequacy and feasibility to evaluate the model performance. It is always likely to choose the most categories and get the higher ARA. On the base of Han and Kamber's previous studies on the imbalanced datasets, sensitivity function (S_e) and specificity function (S_p) can be used to measure the performance of imbalanced datasets. These two metrics can also be used for assessing the performance of IPSO-SVM model [11].

Assume that N_c is the number of positive cases classified correctly; N_e is the number of negative cases classified correctly; N_p is the totally number of positive cases. N_n is the totally number of negative cases. S_e is the ability of recognizing the positive cases:

$$S_e = \frac{N_c}{N_p} \,. \tag{22}$$

S_p is the ability of recognizing the negative cases:

$$S_p = \frac{N_e}{N_n} \,. \tag{23}$$

Based on the definitions of S_e and S_p, precision metric (P) is defined as [12]:

$$P = S_e \cdot \text{ARA} + S_p \cdot (1 - \text{ARA}) \,. \tag{24}$$

Cost metric. The ARA value considers each classification error to be equally significant. This may lead to an overwhelming ARA, but the total cost of classification errors is not the smallest. In order to weigh the different recognition failures, a cost matrix, which defines the costs incurred in three orientations, should be introduced, as shown in Table 4.

Table 4. Cost matrix

$Cost_{ij}$	Left semi-profile	Frontal	Right semi-profile
Left semi-profile	$Cost_{11}$ (0)	$Cost_{12}$ (0.2)	$Cost_{13}$ (0.5)
Frontal	$Cost_{21}$ (0.3)	$Cost_{22}$ (0)	$Cost_{23}$ (0.6)
Right semi-profile	$Cost_{31}$ (0.4)	$Cost_{32}$ (0.8)	$Cost_{33}$ (0)

$Cost_{ij}$ defines the cost of classifying orientation i for a sample when the "true" orientation is j (i, j=1, 2,…, k). In general, $Cost_{ij}$ is zero when i equals j, e.g. $Cost_{11}=0$, $Cost_{22}=0$, $Cost_{33}=0$. $Cost_{ij}$ is different from $Cost_{ji}$ ($i \neq j$), e.g. $Cost_{32}>Cost_{23}> Cost_{13}>Cost_{31}>Cost_{21}>Cost_{12}$.

Assume that N_{ij} is the number of classifying orientation i for j. Hence, cost metric (E) is defined as [13]:

$$E = \frac{\sum_{i,j=1}^{k} Cost_{ij} \times N_{ij}}{N_p + N_n}. \quad \textbf{(25)}$$

Maturity metric. The IPSO-SVM model should have the smallest cost and the highest precision. Therefore, maturity metric (M) is constructed with precision metric (P) and cost metric (E) as follows:

$$M = \begin{cases} \frac{P}{E}, & if\ E \neq 0 \\ +\infty, & if\ E = 0 \end{cases}. \quad \textbf{(26)}$$

5.2 Classification System

According to the IPSO-SVM model and its maturity metric, the classification system for the recognition of face orientations is established at last, as shown in Fig. 3. The value of M is used to judge whether the classification model meets the performance requirement. If it meets the demand, then add the correct classified sample to the training dataset. If not, it will feed back to the classification model for retraining until it fulfills the demand.

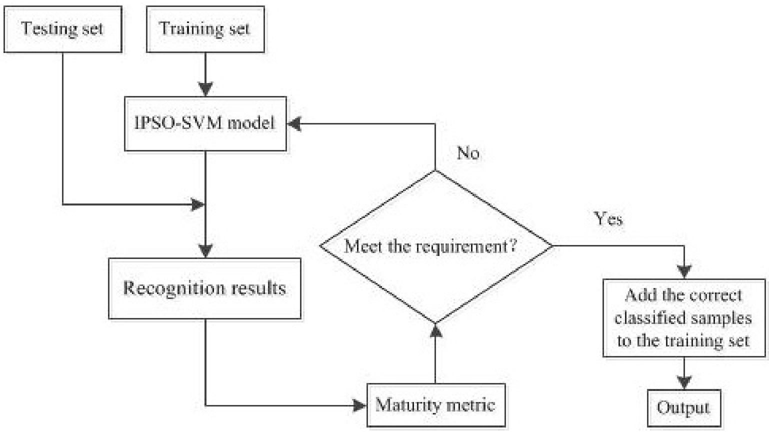

Fig. 3. Classification system

5.3 Case Analysis

As a matter of convenience, 1, 2, and 3 represent left semi-profile, frontal, and right semi-profile respectively. In order to verify the feasibility of the proposed classification system, we apply it to test the levels of the following 18 testing samples. Fig. 4 shows the recognition results. Only one frontal sample is identified as the right semi-profile one. From Eq. (26), the value of M equals 83.33.

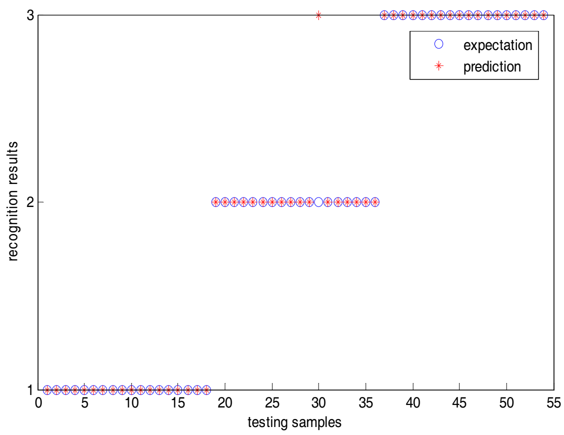

Fig. 4. Recognition results

5.4 Discussion

In order to illuminate the superiorities of the proposed classification system, we compare its recognition accuracy with other methods in Table 5.

Table 5. Comparison of different methods

Methods	Recognition accuracy
Classification system with Eigeneyes	98.15%
Orientation analysis framework [4]	98.00%
Haar features with LVQ [5]	95.00%
Nasal tip and corners of mouth with BPNN [6]	90.00%
Nostril features [7]	97.90%
Pulse coupled neural network [14]	96.00%

All approaches could achieve high recognition accuracy of at least 90.00%, but they do not take the different consequences of misjudgment costs into building models besides our system. The proposed classification system not only has the highest accuracy of 98.15%, but also achieves reliable maturity level (M = 83.33). It indicates that the proposed classification system can be used for face orientations recognition in advance, which will contribute to the success of face recognition.

6 Conclusion

According to the existing problems in the FOR process, this paper proposed a convenient classification system for recognizing face orientations. The IPSO-SVM model is established by taking the normalized Eigenvalues as the input, face orientations as the output. According to the recognition results, maturity metric ($M=P/E$, $E\neq 0$) is defined based on precision metric and cost metric. The innovation of this paper is that the costs incurred in three orientations are defined. The experimental results show that only one frontal sample is identified as the right semi-profile one. The proposed classification system produces M=88.33, which indicates that it achieves efficient recognition accuracy and reliable maturity.

Acknowledgements

The authors would like to thank professor Yong-kui Shi's support and also acknowledge the helpful comments and suggestions of the reviewers.

References

[1] K.S. Ali, U. Muhammad, R. Naveed, Face recognition via optimized features fusion, Journal of Intelligent and Fuzzy Systems 28(4)(2015) 1819-1828.

[2] A.P. Claudio, A.C. Leonardo, E.C. Luis, Methodological improvement on local Gabor face recognition based on feature selection and enhanced Borda count, Pattern Recognition 44(4)(2011) 951-963.

[3] H.S. Dadi, G.K.M. Pillutla, Improved face recognition rate using HOG features and SVM classifier, Journal of Electronics and Communication Engineering 11(4)(2016) 34-44.

[4] J. Zhou, X.-G. Lu, D. Zhang, C.-Y. Wu, Orientation analysis for rotated human face detection, Image & Vision Computing 20(4)(2002) 257-264.

[5] Z.-G. Yan, F. Yang, J. Wang, Y.-J. Shi, C. Lee, M.-X. Sun, Face orientation detection in video stream based on Harr-like feature and LQV classifier for civil video surveillance, in: Proc. 2014 International Conference on Smart and Sustainable City, 2014.

[6] L.-L. Gao, Y.-K. Xu, Face orientation recognition based on multiple facial feature triangles, in: Proc. 2012 International Conference on Control Engineering and Communication Technology, 2012.

[7] N. Nakazawa, T. Mori, A. Maeda, I.-H. Kim, T. Matsui, K. Yamada, Recognition of face orientations based on nostril feature, Journal of Robotics and Mechatronics 24(6)(2012) 1063-1070.

[8] T. Kohonen, Self-organizing and Associative Memory, Springer, New York, 1984.

[9] C.M. Carolyn, H. Stephan, Connectionism as metaphor: towards an integrated, unified conception of self-system and individual difference, Psychological Inquiry 18(2)(2007) 108-113.

[10] V.N. Vapnik, The nature of Statistical Learning Theory, Springer, New York, 1996.

[11] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, Morgan Kaufmann, Waltham, MA, 2001.

[12] L. Rokach, Pattern Classification Using Ensemble Methods, World Scientific Publishing, Singapore, 2015.

[13] Z.-H. Zhou, Machine Learning, Tsinghua University Press, Beijing, 2016.

[14] H.-Y. Zhang, Y. Wang, A PCNN based face orientation recognition scheme, International Review on Computers and Software 7(1)(2012) 387-391.