Signal Processing and Renewable Energy
June 2017, (pp.1-10)
ISSN: 2588-7327

Improvement of Face Recognition Approach through Fuzzy-Based SVM

Leila Yar Mohammady[1], Amir Hoshang Mazinan [2*]
[1,2] Department of Electrical Engineering, Islamic Azad University, South Tehran Branch

Abstract
In this investigation, automatic face recognition algorithms are discussed. For this purpose, a combination of learning algorithms with supervision are realized; in this way, the classification is first designed by the fuzzy-based support vector machine and then the AdaBoost meta-algorithm is applied to the designed classification to reach more accuracy and overfitting control. In the research proposed here, in order to address the effects of asymmetric classes, the adaptive coefficients are employed. In addition, to reduce the data size, the principal components analysis is also applied to the raw data. It is to note that the proposed approach is carried out in a set of images extracted from Yale University data set and its accuracy of the proposed one is verified.

Keywords: Face recognition, classification, fuzzy-based support vector machine, AdaBoost.

1. INTRODUCTION

Face recognition is an action which is usually done unconsciously without any effort and by humans, but in the field of computer vision, even after 30 years of research, it is still a difficult problem and at the beginning of the way to achieve a technological and effective solution. Automatic Face Detection as a biometric technology has the desirable features that conduct research into practical techniques. Over the past decade, face recognition and detection has become one of the most popular fields of research and application in the field of system diagnostics and analysis. One of the goals of such researches is developing user-friendly security systems. Although reliable biometric methods such as fingerprint analysis and retinal or iris scan were successful, they are based on participant's cooperation while image-based systems are usually effective even without participant's cooperation or awareness. Reduction in the price of cameras and increase in operational capacity has led to the development and production of new algorithms that make face recognition systems more functional. These systems are increasingly being used in a wide range of practical applications, and future enhancements will promise greater use of face recognition.

The automatic human's face detection involves a wide range of technologies. At the highest level, technologies are distinguished by the type of input data, such as visible light, infrared light or 3D data. However, more focus is on static black and white images which are captured in visible light, although face recognition in color video images has recently been taken into consideration [1].

*Corresponding Author's Email: mazinan@azad.ac.ir

Each data which is used to detect faces, resistant it to certain conditions, for example, infrared imaging is practically resistant to change in light [2] or in the theory, the 3D data are immutable regardless of how the head is placed [3]. Hence, imaging in the visible light spectrum is still prominent in research and applied fields due to the large amount of legacy data, the simple and continual presence of imaging and inexpensive imaging equipment.

Investigations in the field of face recognition have not mostly been done according to modes, appearance, intensity of light, and facial expressions at the same time. This led to a comprehensive approach for face recognition. Therefore, the fuzzy-based support vector machine model with adaptive coefficients was adopted in combination with Adabobst algorithm.

This method is one of the supervisory learning methods which the data is divided into two parts of the training data and testing data. Using training data, the classifier is designed. Then, the accuracy and precision of the classifier is investigated by testing data. Meanwhile, using dimensional diminishing methods was necessary for reducing dimensions of the data. In the next section, we will look at literature of research. In section 3, used methods are going to be introduced. Section 4 includs numerical results and finally, section 5 is devoted to a general conclusion.

2. BACKGROUND

Due to the fact that facial images are used as data in face recognition process, it is natural that the dimensions of raw data be become numerically large. Hence, in feature based face recognition, use of feature extraction methods is common. Lots of dimension reduction methods are proposed, including those which are based on: special spaces [4]-[6], function core[7]-[9] and Fisher core [10] while owing to low processing speed and the hardness of setting core parameters in nonlinear methods, linear PCA method is used in this research. Although feature extraction techniques greatly facilitate the diagnostic problem, there is a severe drop in the performance when faced with new inputs that are not considered in the training process; for example, when the test face viewing angle is not in front of the camera. That's while training faces had been front faced. Feature extraction with PCA and FLDA varies greatly with such big changes, because these methods are basically based on appearance. As authors suggest in [11], more complex classification than the closest neighborhood is needed owing to inseparable distribution of the face pattern in the FLDA-based and PCA-based subsystem. In other words, classification with appropriate generalization and minimum empirical risk is needed to eliminate the weakness of appearance-based feature extraction. Accordingly, the support vector machine can be a suitable candidate for the classifier.

The support vector machine, which has been successfully applied in many applications, was firstly introduced by Vapnick [12]. In the last decade, lots of efforts have been done in the field of face recognition by using SVM, which has contained desirable results [13], [14].

The first problem of SVM is its sensitivity to inappropriate data because the weight of the penalty is the same for all data [15], [16]. The second problem is that when the SVM is applied to a problem with an unbalanced data set, the class margin skews [17], [18]. In other words, when the number of negative-class data is much higher than the positive-class, the class boundary, or the optimal separating super-plate, obtained using SVM, is skewed toward positive class.

In this case, the fuzzy-based support vector machine and comparative penalty factor are used for solving the distributing data and class margin skewness problems, respectively. The fuzzy-based support vector machine was firstly introduced by Lin and Wang [16] in 2002. In FSVM, a membership grade is assigned to each training data.

Different membership grades lead to different approaches for learning the super-plate. In this way, when the super-splitter plate is found, the effect of distortion and noisy data decreases. Wang [19] introduced a double-faced FSVM by representing two membership functions for each sample, which was later extended to vague collections in [20]. EBA and Aino proposed FSVM for multi-class problems [21], which was a generalization of the binary classification problem, and then applied it to the multi-class categorization [22]. Fuzzy-based support vector regression was also introduced in [23]. While a lot of work has been done in choosing the appropriate fuzzy membership grade, it is still an open problem [16]. The fuzzy membership function is defined based on the Euclidean distance between the samples and the center of classes in the original space while [24] defines these intervals in the high dimension property space. However, both of these fuzzy membership functions are based on the distance between each sample and the center of its class.

In [25], [26], fuzzy membership functions are calculated based on the distance between the samples and the centers of their classes and the dependency of the samples. Using the decision values which are generated by the SVM [27],[28], other membership functions are also obtained. In [29], initial knowledge about the importance of the data which is used to determine the membership grade of the samples is obtained by using the internal class structure of samples (Weraidi). In another approach [30], appropriate centers of each class are selected by clustering methods. Then, appropriate fuzzy membership grade is given to the samples by modifying the margin of the clusters [31].

In the unbalanced data scenario, many traditional classifications usually fail in classification procedure, so it is necessary to use a bias correction technique before or after constructing a classifier. One of these techniques is re-sampling the main training data sets to make the classes approximately equal. Sampling can be sample increment [31] - [33] sample reduction [34], [35] or combined sampling [36]. In this article, samples have been increased.

Several articles have been published for comparing training techniques with monitoring. One of most important training techniques is Adaboost which was firstly introduced by Frédér and Schepier in 1997, but after proposing reforms by Viala and Jones and mixing with classifiers in series structure, it became popular in 2004. This algorithm improves the classifier function by combining several weak classifier functions and creating a stronger classifier. In order to significantly reduce the computational complexity, the complex classifiers are combined in series structure. I this case, complex processing is only done for more interesting examples [37].

3. INTRODUCTION TO PROPOSED METHOD

3.1. Primary Components Analysis (PCA)

The main components analysis is defined in mathematics as an orthogonal linear transformation that transmits the data to a new coordinate system which the largest data variance is set on the first coordinate, the second variance on the second coordinate, and so on for others. In this paper, we analyze main components, which is preserving data components that have the most effect on the variance and removing the others, for reducing the dimensions of the data.

Data has two main characteristics in the new coordinates: 1. There is no correlation between different dimensions of the new data, and 2. Dimensions are arranged according to the importance of their information. Following equations map a data matrix with dimensions $m \times n$,Z_t, to a matrix with dimensions $m \times r$, Z_{tr}.

$$G = \frac{1}{m} \times Z_t^T \times Z_t \tag{1}$$

$$\left[U,S,V\right] = svd(G) \tag{2}$$

$$Z_{tr} = Z_t \times V(:,1:r) \tag{3}$$

where $Z_{tr} = \left[z^{(1)}, z^{(2)}, \dots z^{(r)}\right]^T$ is a matrix which contains features of m different image. S is a diagonal matrix with non-negative elements which is arranged in descending order. Matrices U and V are common matrices that apply to $G = USV^T$ and the svd function parses into singular values. If S_{ii} be the singularity value of the i_{th} feature, the bigger the S_{ii}, the more information the i_{th} feature will contain. In many systems, including visual information, many elements have negligible information. For choosing the r properly, the smallest r that applies to the following inequality is chosen:

$$\frac{\sum_{i=1}^{r} S_{ii}}{\sum_{i=1}^{m} S_{ii}} \times 100 \geq \varepsilon \tag{4}$$

Therefore, ε% remains from the variance. After mapping the features into a lower-dimensional space in (3), machine-identifying techniques can use for classifying individuals (identifying individuals from the database).

Hence, it can be seen that the specific components analysis selects the components that maximize the variance. This important feature justifies the good performance of the proposed method, even with less information (lower dimensions). In fact, not only the PCA extracts all information, but maximizes the separation between different classes. For this reason, PCA has been used in this paper.

3.2. Fuzzy-Based Support Vector Machine

The support vector machine generally designs super-plates or collections of super-plates for classification or regression. By assuming the set of labeled training data like $S = (x_l, y_l), l = 1, \dots, L$ and $y_l = \{-1,1\}$, the support vector machine is solved by the following optimization problem:

$$\min_{\omega,\xi,b} \quad \frac{1}{2}\omega^T \omega + C\sum_{i=1}^{L} \xi_i, \tag{5}$$

$$\text{Subject to} \quad y_i(\omega^T \phi(x_l) + b) \geq 1 - \xi_l \tag{6}$$

$$\xi_i \geq 0, l = 1, \dots, L. \tag{7}$$

where $\phi(x_l)$ is a nonlinear transform that mapped x_l to a higher space. The variable ξ_l is nonlinear for separable training sets and C is the positive parameter for adjustable regulation.

A fuzzy-based support vector machine is used for collections that the importance of all data is not the same. In this way, the importance and impact of data with higher membership rates will be greater. On the other hand, due to the use of one against all mode for generalizing the method to multi-class mode, we will face the asymmetric classes problem. Therefore, comparative coefficients are also added to the equations to overcome this problem. Thus, equations (5) - (7) are rewritten as follows:

$$\min_{\omega,\xi^+,\xi^-,b} \quad \frac{1}{2}\omega^T \omega + C^+\sum_{i=1}^{L_p} s_i^+ \xi_i^+ + C^-\sum_{i=1}^{L_n} s_i^- \xi_i^-, \tag{8}$$

$$\text{Subject to} \quad y_i(\omega^T \phi(x_i) + b) \geq 1 - \xi_i^+, \quad \forall x_i \in S_f^+ \tag{9}$$

$$y_i(\omega^T \phi(x_i) + b) \geq 1 - \xi_i^-, \quad \forall x_i \in S_f^- \tag{10}$$

$$\xi_i^+ \geq 0, \ l = 1, \dots, L_p, \ \forall x_i \in S_f^+. \tag{11}$$

$$\xi_i^- \geq 0, \ l = 1, \dots, L_n, \ \forall x_i \in S_f^-. \tag{12}$$

where L_p and L_n are the number of positive and negative class members, C^+ and C^- the regulating parameter for the positive and negative classes, $\xi^\pm$ the forgetting factor for the positive and negative classes, $S_f^\pm$ the set of data belonging to the positive and negative classes and $s_i^\pm$ the membership rate of i_{th} fuzzy element of the positive or negative class. The fuzzy membership rates are calculated based on the distance between the given data and the class average:

$$\text{Subject to} \quad y_i(\omega^T \phi(x_i) + b) \geq 1 - \xi_i^+, \quad \forall x_i \in S_f^+ \tag{13}$$

Adaptive coefficients are also determined according to the size of the classes. Since the one against all approach is used, the coefficients change for designing each super-plate because size of the classes changes. So, for designing the i_{th} super-plate, if the penalty coefficient C^+ be positive, the penalty factor of negative class will be $C^- = C^+(n-i)$, where n is the total number of classes.

3.3. AdaBoost

The basic idea of Adabost is the simple combination of rules for obtaining a collective classification that works better than each member of the general classification or, in other words, improves their performance. If assume that $h_1, h_2, \dots, h_T$ are sets of hypotheses (classifiers), the general assumption is as follows:

$$f(x)=\sum_{t=1}^{T}\alpha_t h_t(x). \qquad (14)$$

where α_t are the coefficients that combine the hypotheses h_t. α_t and classifiers or h_t hypotheses are learned in the AdaBoost process. There are many methods for choosing the coefficients α_t and the basic assumptionsh_t. The basic idea is that samples that are classified incorrectly gain more weight in the next step. For example, samples which are near the boundaries usually categorize hardly and as a result, they have a higher weight after several steps.

The idea of weighting the training samples at each stage is essential for improving the performance of the final classification. In general, in t_{th} level, non-negative weights $d^{(t)}=(d_1^{(t)},d_2^{(t)},\ldots,d_N^{(t)})$ are attributed to the data, and the weak classifier h_t is constructed based on $\boldsymbol{d}^{(t)}$. These weights are updated in each level t based on the weighting error of the previous stage classifier. In each step t, the weak classifier is attempting to minimize the empirical weight error which is defined in (15):

$$\varepsilon_t(h_t,d^t)=\sum_{i=1}^{N}d_i^{(t)}I(y_i\neq h_t(x_n)). \qquad (15)$$

where $\boldsymbol{I}(.)$ is a vector which its argument takes one in case of setting the condition, otherwsie takes zero. After choosing classifierh_t, its weight α_t is calculated to minimize the following cost function:

$$G(\alpha)=\sum_{i=1}^{N}\exp\{-y_i(\alpha\, h_t(x_i)+f_{t-1}(x_i))\}, \qquad (16)$$

where f_{t-1} which is combination of classifiers of preceding levels is as $f_{t-1}(x_i)=\sum_{r=1}^{t-1}\alpha_r h_r(x_i)$. In this way, coefficients of classifier in each step are calculated as following:

$$\alpha_t=\frac{1}{2}\log\frac{1-\varepsilon_t}{\varepsilon_t} \qquad (17)$$

In the following, AdaBoos algorithm is briefly explained:

3.4. AdaBoost Algorithm

1. Inputs:$S=\{(x_1,y_1),\ldots,(x_N,y_N)\}$y_N)}, number of levels T.
2. Initialize: $d_i^{(1)}=1/N\ i=1,\ldots,N$
3. For $t=1,\ldots,T$

a. Training Classifier with weighted samples $\{S,\boldsymbol{d}^{(t)}\}$ and obtaining the hypothesis (weak classifier) $h_t:\boldsymbol{x}\to\{-1,+1\}$

b. Calculating the weight error ε_t from equation (15).

c. Calculating the classifier coefficient from equation (17).

d. Update the weights with the relation $d_i^{(t+1)}=d_i^{(t)}\exp(-\alpha_t y_i h_i(\boldsymbol{x}_i))/Z_t$, where Z_t is normalization constant which makes $\sum_{i=1}^{N}d_n^{(t+1)}=1$.

e. Calculating the classifier coefficient from equation (17).

f. Update the weights with the relation $d_i^{(t+1)}=d_i^{(t)}\exp(-\alpha_t y_i h_i(\boldsymbol{x}_i))/Z_t$, where Z_t is

Table 1. **Results of using comparative penalty coefficients**

		Train time	Test time	CCR	Row Number
$C^+=C^-=1$	SVM	8,871	3,623	0,8895	1
	FSVM	8,628	3,313	0,7382	2
$C^-=1$	SVM	8,008	4,316	0,8895	3
$C^+=C^-\frac{L_n}{L_p}$	FSVM	8,398	3,659	0,9000	4
$C^-=10$	SVM	7,991	3,284	0,8291	5
$C^+=C^-\frac{L_n}{L_p}$	FSVM	8,174	3,329	0,8697	6

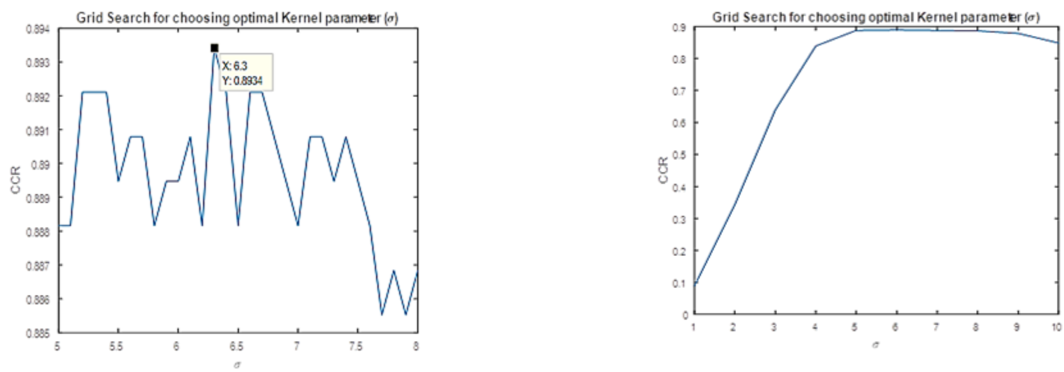

Fig. 1. CCR chart which is affected by change in Kernel Function parameter with low resolution (right) and high resolution (left).

normalization constant which makes $\sum_{i=1}^{N} d_n^{(t+1)} = 1$.

4. Stopping the algorithm if $\varepsilon_t = 0$ or $\varepsilon_t \geq \frac{1}{2}$ and setting $T = t - 1$.
5. Output: $f_T(\boldsymbol{x}) = \sum_{t=1}^{T} \frac{\alpha_t}{\sum_{r=1}^{T} \alpha_r} h_t(\boldsymbol{x})$

4. EXPERIMENTAL RESULTS

In this research, data is images of different individuals with different conditions which are obtained from the Yale University database [38]. But here the fundamental challenge is image preparation and feature extraction. At first, person's image should be separated from the overall picture, which this procedure had been done in the used images.

All the images should be sized (in terms of number of pixels). Then the light intensity of each pixel of the image (which is black and white) is considered as its feature. Thus, for each image, a feature vector which includes the light intensity of all pixels of the image is obtained. This vector will be the basis of classification. The main problem is the high dimensions of the feature vector (dimensions are equal to the number of image pixels), which will be resolved by analyzing main components. In this research, the dimensions are reduced from 32256 to 59 by recovering 95% of the initial data which indicates that much of the early features did not have much effect on data formation. Input data is now ready for classifier.

In this research, at first the classifier of support vector machine is compared with its fuzzy-based version. Correct Classification Rate (CCR) is the scale which is used for evaluating classifier performance. For this purpose, a table is made up of the number of classes which rows represent the actual class of data and columns represent the class that classifier gives to the data. In this way, each data is located in the above table (Confusion Matrix) according to the actual class and its assigned. It is desirable that class which the classifier recognizes be equal to the actual class of data. That is, locating the data on the main diameter of matrix. In this way, CCR can be defined as follows:

$$\text{CCR} = \frac{\text{tr (Confusion Matrix)}}{N}$$

where tr(.) Is the Trace function and N is the total number of data. It is desired that the CCR be one or close to one. Since the one against all approach is adopted, size of the negative class is changed in the design of each classifier. Therefore, the adaptive coefficient of the positive class is determined based on the magnitude of the negative class. For example, the coefficient of the negative class is taken 1 or 10, and the coefficient of the positive class is taken as a coefficient of negative class size and its coefficient. In this way, the problem of asymmetric classes is raised to the optimal level and the generality of the class is improved. Table.1 shows the effect of using adaptive coefficients on the improvement of fuzzy-based support vector machine operation.

It can be seen that without FSVM's adaptive coefficients, the results are not very favorable and even SVM performs better, but as soon as these coefficients are used, not only FSVM performance is improved but also it performs much better than SVM. It can be seen that the implementation time for both training and testing data is the same in both methods.

Hence, there is no difference between them in terms of computational speed. It should be noted that in all of the above problems, the RBF Kernel function with parameter 6.3 has been used. This parameter is obtained using the grid search method which gives us the most optimal possible answer. Fig.1 shows the change in the CCR value by changing the kernel function parameter value. In this case, Fig.2 illustrates the classification of testing data more tangibly. In this figure, the horizontal and vertical axes are data number and the actual class of data, respectively. Each 20 testing data belongs to a class, respectively, and each color indicates the class which is attributed to each data.

It should be noted that because of limitations in colors, the colors are repeated after each seven classes. It can be seen that the classification is largely done correctly, and in each class, a handful of data are mistakenly categorized.

However, we use AdaBoost method to improve classifier performance. In the following, results of using AdaBoos algorithm for improving the performance of the classifier are examined.

In next step, the previously designed FSVM classifier enters the AdaBoost algorithm and repeats 10 levels. In fact, the AdaBoost considers the classifier with different coefficients for different data and finally presents the classification derived from the total class of preceding classes with specific coefficients that significantly reduce the error. Fig.2 depicts a graphical description of results of AdaBoost and FSVM. It is seen that the error value has been greatly reduced in the Adaboost algorithm. This indicates the superiority of the proposed method.

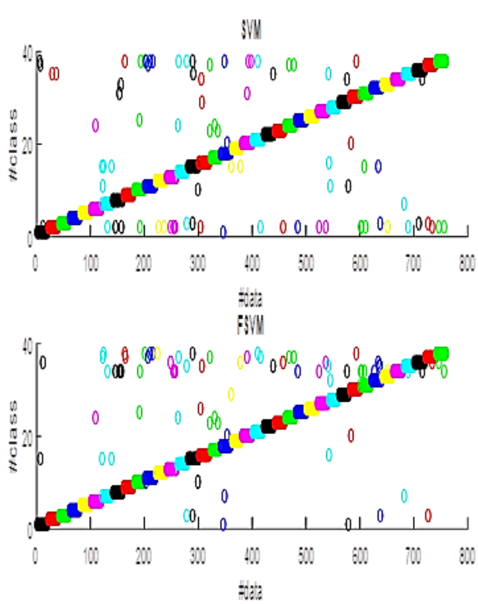

Fig. 2. Comparing testing data classification by the SVM and FSVM (right), FSVM and AdaBoost (left).

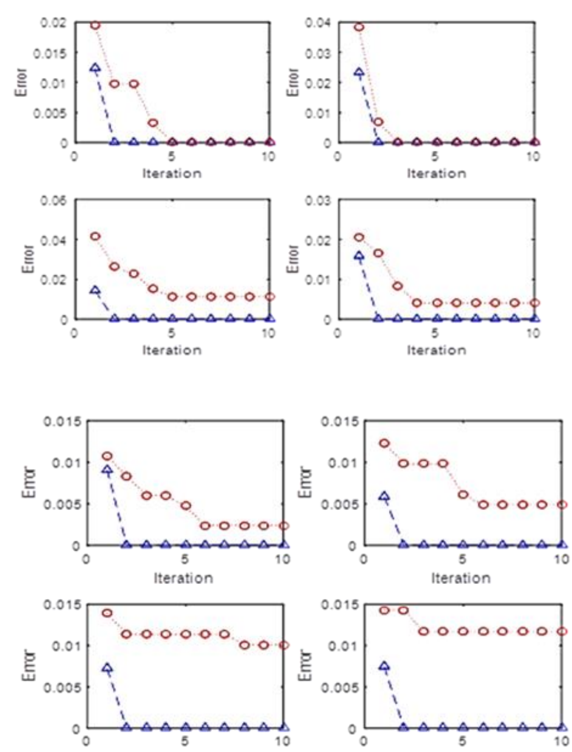

Fig. 3. Reduction in error of training data (blue) and testing data (red) in AdaBoost repetition algorithm for classifiers 1 to 4 (right) and classifiers 25 to 28.

Table 2. **Results of proposed and FSVM**

	Train time	Test time	CCR	Row Number
Proposed Method	12/627	5/705	0/9724	1
FSVM	8/398	3/659	0/9000	2

Actually, AdaBoost attempts to reduce the error rate of testing data at any level. This is shown in Fig.3. These shapes are examples of reduced error rate of a classifier which is designed for each class, but since there are so many classes (38 classes), here are some examples. It is seen that the error of training data decreases during the repetition levels, which decreases further by increasing the levels of the algorithm. Table 2-4 also highlights the superiority of the proposed method than FSVM.

In fact, this advantage is at the expense of increasing training and testing time, but this increase in time for testing data is not impressive. Therefore, although more time is spent for training, this increase in testing time is not discouraging. In addition, a significant improvement in the performance of the proposed method is a good incentive for using this method.

5. CONCLUSION

In this research, it has been shown that the proposed method utilized fuzzy-based support vector machine, adaptive coefficients and Adaboos algorithm to increase the accuracy of facial recognition. In this method, although the precision of training classifier is sacrificed to speed, testing time and measuring precision are good drivers for encouraging individuals to use this method. In addition, this method does not require any quality or color photographs, and can be identified with any full-scale photograph. The proposed method is also resistant to facial changes, because the data is taken from all facial expressions (such as anger, grief, joy).

REFERENCES

[1] E. G. P. V. R. R. S. D. O. G. Miguel De-la-Torre, "Partially-supervised learning from facial trajectories for face recognition in video surveillance," Information Fusion, vol. 24, p. 31–53, 2015.

[2] R. C. S. L. L. Z. Stan Z. Li, "Illumination Invariant Face Recognition Using Near-Infrared Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 627 - 639, 2007.

[3] T. V. V. Blanz, "Face recognition based on fitting a 3D morphable model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1063 - 1074, 2003.

[4] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, John Wiley & Sons, 2012.

[5] A. P. B. Moghaddam, "Probabilistic visual learning for object representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 696 - 710, 2002.

[6] P. N. J Ruiz-del-Solar, "Toward a generalized eigenspace-based face Recognition System," Lecture Note in Computer Science, 2002.

[7] V. V. Corinna Cortes, "Support Vector Networks," Machine Learning, vol. 20, no. 3, pp. 273-297, 1995.

[8] A. S. K.-R. M. Bernhard Schölkopf, "Nonlinear component analysis as a kernel eigenvalue problem," Neural Computation, vol. 10, no. 5, pp. 1299-1319, 1998.

[9] K. J. H. J. K. Kwang In Kim, "Face recognition using kernel principal component analysis," IEEE Signal Processing Letters , vol. 9, no. 2, pp. 40-42, 2002.

[10] H. L. S. M. Qingshan Liu, "Improving kernel fisher discriminant analysis for face Recognition," IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 1, pp. 1051-8215, 2004.

[11] J. Lu, N. Plataniotis and A. N. Venetsanopoulos, "Face recognition using kernel discriminant analysis algorithm," IEEE Transaction on Neural Network, vol. 14, no. 1, pp. 117-126, 2003.

[12] C. Cortes and V. Vapnik, "Support Vector Networks," Machine Learning, vol. 20, no. 3, pp. 273-297, 1995.

[13] Y. Wen, "An improved discriminative common vectors and support vector machine based face recognition approach," Expert Systems with Applications, vol. 39, no. 4, pp. 4628-4632, 2011.

[14] L. H. Thai, T. S. Hai and N. T. Thuy, "Image Classification using Support Vector Machine and Artificial Neural Network," International Journal of Information Technology and Computer Science, vol. 4, no. 5, pp. 32-38, 2012.

[15] I. Guyon, N. Matic and V. Vapnik, "Discovering informative patterns and data cleaning," Advanced in knowledge discovery and data mining, pp. 181-203, 1996.

[16] C. F. Lin and S. D. Wang, "Fuzzy Support Vector Machines," Neural Network, IEEE Transaction on, vol. 13, no. 2, pp. 464-471, 2002.

[17] Y. Zhang, P. Fu, W. Liu and L. Zou, "SVM classification for imbalanced data using conformal kernel transformation," in Neural Networks (IJCNN), 2014 International Joint Conference on, Beijing, 2014.

[18] S. Maldonado, R. Weber and F. Famili, "Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines," Information Sciences, vol. 286, pp. 228-246, 2014.

[19] Y. Wang, S. Wang and K. Lai, "A new fuzzy vector machine to evaluate credit risk," *Fuzzy systems, IEEE Transactions on*, vol. 13, no. 6, pp. 820-831, 2005.

[20] Y. Hao, Z. Chi and D. Yan, "Fuzzy support vector machine based on vague sets for credit assessment," in Proceeding of the Forth International Conference on fuzzy Systems and Knowledge Discovery, Changsha, China, 2007.

[21] S. Abe and T. Inoue, "Fuzzy support vector machines for multiclass problems," in Proceedings of the Tenth European Symposium on Artificial Neural Networks, Burges, belgium, 2002.

[22] T. Wang and H. Chiang, "fuzzy support vector machine for multi-class text categorization," Information Processing & Management, vol. 43, no. 4, pp. 914-929, 2007.

[23] Z. Sun and Y. Sun, "Fuzzy support vector machine for regression estimation," in IEEE International Conference on systems, man and cybernetics, 2003.

[24] X. Jiang, Y. Zhang and J. Lv, "Fuzzy SVM with a new fuzzy membership function," Neural Computation & Application, vol. 15, no. 4, pp. 268-276, 2006.

[25] X. Zhang, X. Xiao and G. Xu, "Fuzzy support vector machine based on affinity among samples," Journal of Software, vol. 17, no. 5, pp. 951-958, 2006.

[26] H. Tang and L.-s. Qu, "Fuzzy support vector machines with a new fuzzy membership function for pattern classification," in Machine Learning and Cybernetics, 2008 International Conference on , Kunming, 2008.

[27] Z. Xie, Q. Hu and D. Yu, "Fuzzy output support vector machine for classification," in Proceedings of the International Conference on Advances in Neural Computation, Changsha, china, 2005.

[28] T. Inoue and S. Abe, "Fuzzy support vector machine for pattern classification,"

in Proceedings of the International Conference on Neural Networks, Washington, DC, 2001.

[29] W. An and M. Liang, "Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noise," Neurocomputing, vol. 110, pp. 101-110, 2013.

[30] Z. Wu, H. Zhang and J. Liu, "A fuzzy support vector machine algorithm for classification based on a novel PIM fuzzy clustering method," Neurocomputing, vol. 125, pp. 119-124, 2014.

[31] A. Chatterjee, D. Mishra and S. S. Gorthi, "Enhancing Face Recognition Under Unconstrained Background Clutter Using Color Based Segmentation," Advances in Intelligent Systems and Computing, vol. 425, pp. 51-62, 2016.

[32] H. Zhang and M. Li, "RWO-Sampling: A random walk over-sampling approach to imbalanced data classification," Information Fusion, vol. 20, p. 99–116, 2014. N. Chawla, K. Bowyer and L. Hall, "SMOTE: synthetic minority over-sampeling technique," Journal of Artificial Inteligence, vol. 16, pp. 321-357, 2002.

[33] R. Akbani, S. Kwek and N. Japkowicz, "Applying support vector machine to imbalanced datasets," in Proceedings of the 15th European Conference on Machine Learning, Pisa, Italy, 2004.

[34] S. Yen and Y. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," Expert Systems with Applications, vol. 36, no. 3, pp. 5718-5727, 2009.