Non-Linear Fusion of Local Matching Scores for Face Verification

Ziheng Zhou Samuel Chindaro Farzin Deravi
Department of Electronics, University of Kent
Canterbury, UK CT2 7NT
{Z.Zhou,S.Chindaro,F.Deravi}@kent.ac.uk

Abstract

This paper presents a face verification framework for fusing matching scores that measure similarities of local facial features. The framework is aimed to handle an open-set verification scenario when users who try to enroll can be unknown to the system at the training phase. The kernel discriminant analysis is adopted within the framework to explore the discriminatory information of local matching scores in a high-dimensional non-linear space. A large sample size problem is raised for system training and an effective strategy is provided for tackling this problem. We demonstrate the framework by fusing the scores calculated using local binary pattern features. The experimental results show that our method improves the verification performance significantly when compared to a number of competitive techniques.

1. Introduction

The human face has long been studied as an important source of biometric information to facilitate the recognition of individuals [14]. Recently, researchers have attempted to extract local features from images for face recognition [6, 13, 1, 7, 15]. In these methods, features are computed directly from images, without a training process for feature extraction. To compute a similarity score for two face images, local facial features are extracted and used to calculate some local matching scores. The overall similarity score is often computed as a weighted sum of the local matching scores and the weights are picked empirically. Kyperountas *et al*. [7] introduced the weighted piecewise linear discriminant analysis to learn weights in a systematic way. The weights found were bound to subjects, that is, different subjects had different weights. They showed that the system outperformed the original local matching method. However, their approach can only be applied to the close-set situation when all users are already known to the system at the training phase.

In this paper, a framework for fusing local matching scores is proposed to improve the verification performance. It is aimed to deal with the open-set situation, meaning that the verification system will be expecting some new users whom it does not know at the training phase. It is a typical situation that has to be handled by a large-scale face verification system. In the framework, the local matching scores gained from a comparison of two face images are formed into a vector, and the vector is labeled as a *genuine* sample if the two images are from the same subject or an *imposter* sample if the images are from different subjects. The kernel discriminant analysis (KDA) [8] is then performed on the two classes of score vectors to explore their discriminatory potential in a high-dimensional non-linear space. Instead of having a small sample size problem [5] when training some face-recognition systems [4, 12, 3], we consider a large sample size problem. It is noticed that the number of score vectors that can be calculated from N face images is $\frac{1}{2}(N^2 + N)$. For instance, if 500 face images are used for training, a training set that contains more than 10^5 score vectors can be calculated. Very often, the number of training images is larger than 500. Such large amount of training data could make it computationally difficult for kernel-based methods. In this paper, an effective strategy is provided to tackle the large sample size problem.

The framework is demonstrated by fusing the local matching scores calculated in the same way as Ahonen *et al*. [1] using local binary pattern (LBP) features. (Note that the framework can also be applied to the scores computed from other local features.) The system is tested on a subset of the face recognition grand challenge (FRGC) database [9]. In our experiments, images are divided in such a way that subjects included in the test data are not also included in the training data. Therefore, we can test our system's capability of handling the open-set situation. Experimental results indicate that the proposed framework significantly improves the verification performance.

The rest of paper is organized as follows: Section 2 describes how local matching scores are fused using KDA and how the large sample size problem is dealt with. The experiments and results are presented in Section 3.

978-1-4244-2154-1/08/$25.00 2008 IEEE

Section 4 concludes our work.

2. Proposed Framework

2.1. Fusing Scores Using KDA

The purpose of the fusion of local matching scores is to combine all the available scores into one score whose genuine and imposter distributions are more separated than those of the single scores to reach a better verification performance. The linear discriminant analysis (LDA) is suitable for this task. The basic idea of LDA is to search in the feature space for a direction along which the mapped data points belonging to different classes are maximally separated. If the features are set to be the local matching scores, the fusion can be done simply by finding in the feature space the direction of maximal separation and mapping scores onto it. In this section, $\boldsymbol{x}$ is used to denote the score vector constructed from all the local matching scores computed from a comparison of two face images.

Given a training set comprised of genuine and imposter score vectors, the extent of separation for a particular direction is determined by the ratio of the inter-class variations of the mapped scores over the intra-class variations. The optimal direction $\boldsymbol{w}^*$ maximizes the ratio. In the case of verification, the LDA method can be expressed as [8]:

$$\begin{aligned} \boldsymbol{w}^* &= \arg\max_{\boldsymbol{w}} \frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_b\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_w\boldsymbol{w}} && (1)\\ \boldsymbol{S}_b &= (\boldsymbol{\mu}_{\mathrm{g}}-\boldsymbol{\mu}_{\mathrm{i}})(\boldsymbol{\mu}_{\mathrm{g}}-\boldsymbol{\mu}_{\mathrm{i}})^{\mathrm{T}} && (2)\\ \boldsymbol{S}_w &= \boldsymbol{C}_{\mathrm{g}}+\boldsymbol{C}_{\mathrm{i}} && (3) \end{aligned}$$

where $\boldsymbol{S}_b$ and $\boldsymbol{S}_w$ are the between-class and within-class scatter matrices, $\boldsymbol{\mu}_{\mathrm{g}}$ and $\boldsymbol{\mu}_{\mathrm{i}}$ are the means of the genuine and imposter score vectors and $\boldsymbol{C}_{\mathrm{g}}$ and $\boldsymbol{C}_{\mathrm{i}}$ are their covariance matrices. The fused scores x^* is the dot product of $\boldsymbol{w}^*$ and the vector $\boldsymbol{x}$ containing scores to be fused.

When the genuine and imposter scores cannot be separated linearly to reach the accuracy requirement, we have to increase the complexity of the feature space. The kernel discriminant analysis is used to tackle this problem. The basic idea of KDA is to map the data points to a new space with a higher dimension and search a direction that better separates them using LDA in the new feature space. Let Φ be the mapping to the new space $\mathcal{F}$ and $\Phi(\boldsymbol{x})$ be the features mapped from $\boldsymbol{x}$. The optimal direction $\boldsymbol{w}^*$ in $\mathcal{F}$ maximizes the ratio $J(\boldsymbol{w})$:

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_b^{\Phi}\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_w^{\Phi}\boldsymbol{w}} \qquad (4)$$

where $\boldsymbol{S}_b^{\Phi}$ and $\boldsymbol{S}_w^{\Phi}$ are the corresponding between- and within-class scatter matrices measured in $\mathcal{F}$. Following the definitions of the between- and within-class scatter matrices, $\boldsymbol{S}_b^{\Phi}$ and $\boldsymbol{S}_w^{\Phi}$ can be written as:

$$\begin{aligned} \boldsymbol{S}_b^{\Phi} &= \left(\boldsymbol{\mu}_{\mathrm{g}}^{\Phi}-\boldsymbol{\mu}_{\mathrm{i}}^{\Phi}\right)\left(\boldsymbol{\mu}_{\mathrm{g}}^{\Phi}-\boldsymbol{\mu}_{\mathrm{i}}^{\Phi}\right)^{\mathrm{T}} && (5)\\ \boldsymbol{S}_w^{\Phi} &= \boldsymbol{C}_{\mathrm{g}}^{\Phi}+\boldsymbol{C}_{\mathrm{i}}^{\Phi} && (6) \end{aligned}$$

where $\boldsymbol{\mu}_{\mathrm{g}}^{\Phi}$ and $\boldsymbol{\mu}_{\mathrm{i}}^{\Phi}$ are the means of the mapped genuine and imposter scores and $\boldsymbol{C}_{\mathrm{g}}^{\Phi}$ and $\boldsymbol{C}_{\mathrm{i}}^{\Phi}$ are the corresponding covariance matrices.

In general, it is difficult to solve (4) directly since $\mathcal{F}$ could be very high-dimensional. However, this problem can be tackled using the 'kernel trick' [10] that allows us to evaluate the dot product between two mapped feature points $\Phi(\boldsymbol{x})$ and $\Phi(\boldsymbol{x}')$ by a kernel function $k(\boldsymbol{x},\boldsymbol{x}')$ without having to explicitly compute $\Phi(\boldsymbol{x})$ and $\Phi(\boldsymbol{x}')$. To express the problem in terms of dot products, it is assumed that $\boldsymbol{w}$ is a linear combination of the mapping of a set of score vectors $\mathcal{X} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^M\}$:

$$\boldsymbol{w} = \sum_{j=1}^{M} \alpha_j \Phi(\boldsymbol{x}^j). \qquad (7)$$

Using (7) and the kernel trick, we have:

$$\begin{aligned} \boldsymbol{w}^{\mathrm{T}}\Phi(\boldsymbol{x}) &= \sum_{j=1}^{M}\alpha_j k(\boldsymbol{x}^j,\boldsymbol{x})\\ &= \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{k}_{\mathcal{X}}(\boldsymbol{x}) && (8) \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{\alpha} &= [\alpha_1, \alpha_2, \ldots, \alpha_M]^{\mathrm{T}}\\ \boldsymbol{k}_{\mathcal{X}}(\boldsymbol{x}) &= \left[k(\boldsymbol{x}^1,\boldsymbol{x}), k(\boldsymbol{x}^2,\boldsymbol{x}), \ldots, k(\boldsymbol{x}^M,\boldsymbol{x})\right]^{\mathrm{T}} \end{aligned}$$

After having (8) and the definitions of $\boldsymbol{S}_b^{\Phi}$ and $\boldsymbol{S}_w^{\Phi}$, we can rewrite the numerator and denominator in (4) as:

$$\begin{aligned} \boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_b^{\Phi}\boldsymbol{w} &= \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{S}_b^{\mathcal{X}}\boldsymbol{\alpha} && (9)\\ \boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_w^{\Phi}\boldsymbol{w} &= \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{S}_w^{\mathcal{X}}\boldsymbol{\alpha} && (10) \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{S}_b^{\mathcal{X}} &= \left(\boldsymbol{\mu}_{\mathrm{g}}^{\mathcal{X}}-\boldsymbol{\mu}_{\mathrm{i}}^{\mathcal{X}}\right)\left(\boldsymbol{\mu}_{\mathrm{g}}^{\mathcal{X}}-\boldsymbol{\mu}_{\mathrm{i}}^{\mathcal{X}}\right)^{\mathrm{T}} && (11)\\ \boldsymbol{S}_w^{\mathcal{X}} &= \sum_{s=1}^{N_{\mathrm{gen}}}\left(\boldsymbol{k}_{\mathcal{X}}(\boldsymbol{x}_{\mathrm{g}}^s)-\boldsymbol{\mu}_{\mathrm{g}}^{\mathcal{X}}\right)\left(\boldsymbol{k}_{\mathcal{X}}(\boldsymbol{x}_{\mathrm{g}}^s)-\boldsymbol{\mu}_{\mathrm{g}}^{\mathcal{X}}\right)^{\mathrm{T}}\\ &+ \sum_{t=1}^{N_{\mathrm{imp}}}\left(\boldsymbol{k}_{\mathcal{X}}(\boldsymbol{x}_{\mathrm{i}}^t)-\boldsymbol{\mu}_{\mathrm{i}}^{\mathcal{X}}\right)\left(\boldsymbol{k}_{\mathcal{X}}(\boldsymbol{x}_{\mathrm{i}}^t)-\boldsymbol{\mu}_{\mathrm{i}}^{\mathcal{X}}\right)^{\mathrm{T}} && (12)\\ \boldsymbol{\mu}_{\mathrm{g}}^{\mathcal{X}} &= \frac{1}{N_{\mathrm{gen}}}\sum_{s=1}^{N_{\mathrm{gen}}}\boldsymbol{k}_{\mathcal{X}}(\boldsymbol{x}_{\mathrm{g}}^s) && (13)\\ \boldsymbol{\mu}_{\mathrm{i}}^{\mathcal{X}} &= \frac{1}{N_{\mathrm{imp}}}\sum_{t=1}^{N_{\mathrm{imp}}}\boldsymbol{k}_{\mathcal{X}}(\boldsymbol{x}_{\mathrm{i}}^t). && (14) \end{aligned}$$

Here $\{\boldsymbol{x}_{\mathrm{g}}^s\}_{s=1}^{N_{\mathrm{gen}}}$ and $\{\boldsymbol{x}_{\mathrm{i}}^t\}_{t=1}^{N_{\mathrm{imp}}}$ are the genuine and imposter score vectors in the training set, and N_{gen} and N_{imp} are the numbers of the genuine and imposter score vectors. The problem then turns out to find $\boldsymbol{\alpha}^*$ that maximizes

$$J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{S}_b^{\mathcal{X}} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{S}_w^{\mathcal{X}} \boldsymbol{\alpha}}. \tag{15}$$

It can be considered equal to finding an LDA solution in the M-dimensional space defined by the mapping $\boldsymbol{k}_{\mathcal{X}}(\boldsymbol{x})$. When the within-class matrix $\boldsymbol{S}_w^{\mathcal{X}}$ is non-singular, the solution is given by:

$$\boldsymbol{\alpha}^* = \left(\boldsymbol{S}_w^{\mathcal{X}}\right)^{-1} \left(\boldsymbol{\mu}_{\mathrm{g}}^{\mathcal{X}} - \boldsymbol{\mu}_{\mathrm{i}}^{\mathcal{X}}\right) \tag{16}$$

and the fused score x^* can be computed as a dot product of $\boldsymbol{\alpha}^*$ and $\boldsymbol{k}_{\mathcal{X}}(\boldsymbol{x})$.

2.2. Large Sample Size Problem

The dataset $\mathcal{X}$ is usually set to be the whole training set to exploit all the information within the training data. However, when the training set is very large, this makes it almost computationally impossible since the number of score vectors in the training set could easily become too large to evaluate $\boldsymbol{\alpha}^*$ that maximizes (15). To solve this problem, we choose $\mathcal{X}$ as a subset of the training data. By limiting the size of $\mathcal{X}$, we can control the dimension of the space defined by $\boldsymbol{k}_{\mathcal{X}}(\boldsymbol{x})$ and therefore make it computationally affordable to perform LDA in this space. It is noticed that there would some loss of information since we cannot use all the training data to form the bases of the space where the solution of KDA, $\boldsymbol{\alpha}^*$ is evaluated. However, in spite of the information loss, some significant improvement of the verification performance can still be achieved by performing KDA on some carefully prepared training dataset.

There are two major drawbacks of using all the score vectors that can be calculated from the images to train the system. Firstly, it makes the computation of KDA still expensive (e.g., when evaluating $\boldsymbol{S}_w^{\mathcal{X}}$). Secondly, the system training can be biased by some subjects who have a large number of images used to calculate the score vectors for training. Normally, we want a training set to have the following two characteristics:

1. The training set should represent the distribution of the score vectors as much as possible. Therefore, the way we separate the genuine and imposter data within the training set can be applied to other test data and result in reasonable verification accuracy.

2. The training set should have a reasonable size for computational purposes.

One usual way to prepare a training set is to choose an equal number of images for each subject and calculate scores using the selected images. To control the size of the training set, we have to limit the number of images from each subject. For example, if we have 300 subjects and select four images for each of them, we will have 1200 images. From these images, we can calculate only 1800 genuine score vectors (there are $\frac{1}{2} \times 4 \times (4-1) = 6$ genuine score vectors computed from four images for each subject) which might not be enough to represent the distribution of genuine score vectors. On the contrary, we can calculate more than 7×10^5 imposter score vectors which might be too many for training.

It is proposed not to select images from every subject for computing the score vectors for training. Here the subjects involved in training are denoted as $\{s_1, s_2, \ldots, s_n\}$ and the images belonging to s_i as $\{I_1^i, I_2^i, \ldots, I_{n_i}^i\}$ where n_i is the number of images from s_i. Between subjects s_i and s_j $(i \leq j)$, we want to calculate n_{gen} genuine score vectors when $i = j$ or n_{imp} imposter score vectors when $i < j$. We do not allow $i > j$ to avoid calculating score vectors for two subjects twice. Since score vectors can be uniquely labeled by the image pairs from which they are computed, the genuine score vectors for s_i (if $i = j$) can be expressed as

$$\mathcal{P}_{\mathrm{gen}}^i = \{(I_u^i, I_v^i) | u < v, 1 \leq u \leq (n_i - 1), 2 \leq v \leq n_i\}. \tag{17}$$

and the imposter score vectors (if $i < j$) can be written as:

$$\mathcal{P}_{\mathrm{imp}}^{i,j} = \{(I_u^i, I_v^j) | 1 \leq u \leq n_i, 1 \leq v \leq n_j\}. \tag{18}$$

We then randomly choose n_{gen} (n_{imp}) image pairs from $\mathcal{P}_{\mathrm{gen}}^i$ ($\mathcal{P}_{\mathrm{imp}}^{i,j}$) and use the corresponding images to calculate the genuine (imposter) score vectors. The process of selecting image pairs and calculating score vectors is carried out for every subject pair, resulting in, in total, $N_{\mathrm{gen}} = n \times n_{\mathrm{gen}}$ genuine score vectors and $N_{\mathrm{imp}} = \frac{n(n-1)}{2} \times n_{\mathrm{imp}}$ imposter score vectors in the training set. We can choose a relatively large n_{gen} to obtain a genuine dataset with a proper size and a suitable n_{i} to avoid too many imposter score vectors.

After the training dataset is ready, $\mathcal{X}$ can then be chosen from the set. Experiments have been performed to find out the proportions of the genuine and imposter score vectors in $\mathcal{X}$ which can give reasonable verification performance. These suggest that once the size of training set is sufficiently large, the proportions had little influence on the performance. In the following experiments, $\mathcal{X}$ was formed using half genuine and half imposter score vectors. The genuine (imposter) score vectors are randomly chosen out of the N_{gen} (N_{imp}) genuine (imposter) score vectors in the training set.

Partition	Mean	Std
1	36.64	24.03
2	41.52	24.31
3	45.76	26.01
4	39.76	27.58
5	32.72	23.82
6	33.12	21.65
7	30.32	24.47

Table 1. Means and standard deviations of the number of images per subject in different image partitions.

3. Experiments and Results

Since the framework has to be tested in an open-set scenario, we have designed some experiments where test images are from the subjects who are unknown to the system at the training phase. Experiments were performed on the images from the face recognition grand challenge database. All the images used were taken under a controlled environment (with a static clean background and controlled lighting). Since results could be biased depending on the choice of particular training and test data, the framework was trained and tested on various datasets to evaluate the verification performance. To do so, we prepared seven image partitions each of which contained all the images from 50 subjects. The subjects in each partition were unique to the other partitions. Table 1 shows the mean and standard deviation of the number of images per subject in each of the partitions. Cross validation was adopted to test the system, using one image partition for validating and the rest of the images for training. The validating set was changed from one partition to another until all the partitions had been used. In this way, we were able to test the system's capability of handling the users whose images were not used in training.

For each image partition used for validation, a training set and a test set of score vectors have to be calculated to train and test the framework. As described in Section 2.2, two parameters n_{gen} and n_{imp} have to be fixed to calculate a training set. Their values were fixed based on some experiments carried out using training sets with different sizes. It was found out that a very large training set (for both genuine and imposter data) did not result in any significant performance improvement after its size reached a certain level (e.g., 4×10^4 genuine and 5×10^5 imposter score vectors). Below that level, enlarging the training set did improve the performance. Here we set $n_{\text{gen}} = 100$ and $n_{\text{imp}} = 10$. To build the test set, all the images in the selected partition were used to calculate the score vectors. Note that the local matching scores were computed strictly following the way described in [1] in order to show how much the verification performance could be improved by fusing the local matching scores within the framework. Hence, images were normalized following Beveridge *et al.* [2], defined sub-regions by some 7×7 windows, used the extended LBP operator, $\text{LBP}_{8,2}^{\text{u2}}$ to compute the LBP features and calculated scores using the Chi square statistic.

To employ KDA, a kernel function was chosen for the algorithm. Two most commonly used kernels were tested: the Gaussian RBF kernel, $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-||\boldsymbol{x} - \boldsymbol{y}||^2/p)$ and the polynomial kernel, $k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} \cdot \boldsymbol{y})^d$ where p and d are positive constants. We tried different values of p for the RBF kernel and d for the polynomial kernel to train and test the framework using the same training and test data. The RBF kernel outperformed the polynomial kernel significantly. The RBF kernel with $p = 50$ was used in our experiments. All the local matching scores were zero-score normalized [11]. Finally, the size of dataset $\mathcal{X}$ was chosen based on some experiments performed using $\mathcal{X}$ with different sizes. The results showed that a larger $\mathcal{X}$ can result in a better separation of the genuine and imposter data in the training set, however, when its size became too large (more than 5000), the verification performance remained more or less the same on the test set. We decided to build $\mathcal{X}$ with 2500 genuine and 2500 imposter score vectors. They were all randomly selected from the training set.

Here we summarize the key steps in the process of score fusion:

1. Choose an image partition for validating and the rest for training.
2. Calculate a training set and a test set of score vectors from images.
3. Select $\mathcal{X}$ randomly from the training set.
4. Train the system using KDA.
5. Fuse scores.

To test the robustness of the proposed framework, we repeated the procedure from Step 2 to 5 five times for each image partition. For comparison, we implemented Ahonen *et al.*'s face recognition system and tested it for face verification. In addition, we compared the fusion framework with two other fusion methods: the simple sum rule and the LDA-based fusion. The latter was performed on the same training sets used to train our system. For reasons of simplicity, we label our fusion framework by KDA, the LDA-based fusion by LDA, Ahonen *et al.*'s system by LBP and the simple sum rule by SUM in the following tables and figures.

Figure 1 shows the average verification rates at the false acceptance rates (FARs) of 0.1% and 1%. At the FAR of 0.1%, the fusion framework achieved 87% verification rate on the test data, which was 10.5% higher than the LBP system and 6% higher than the LDA-based fusion.

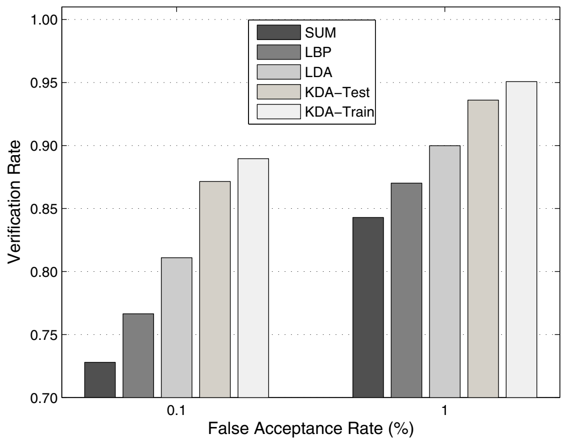


Figure 1. Average verification rates reported for false acceptance rates of 0.1% and 1%. The KDA-test bars show our system's performance on the test data. The performance on the training data was illustrated by the KDA-train bars.

We also show in the figure the verification rates on the training data for our system. They are higher than the rates on the test data by 1.8% when $\mathrm{FAR} = 0.1\%$ and by 1.4% when $\mathrm{FAR} = 1\%$ respectively. Figure 2 shows the receiver operating characteristic (ROC) curves of the experiments carried out on different image partitions. In each figure, there are five KDA and LDA curves since five different sets of training data were calculated after one partition was chosen for validating. The verification performance is varied from one image partition to another. For all the test partitions, the fusion framework significant outperforms the original LBP system. It also outperforms the LDA-based fusion, indicating that the scores computed from the LBP features could be better separated non-linearly in a higher-dimensional feature space. Table 2 lists the equal error rates (EERs) of the fused scores. We show the ranges of the EERs of the scores fused by the fusion framework and compare them with those obtained by other methods. As expected, the maximal EER of KDA is still smaller than other EERs for all the test partitions.

4. Conclusion

This paper has described a framework for fusing local matching scores that can be calculated from different local facial features. The framework is designed to handle not only the users whose images are used to train the system but those who are unknown to the system at the training phase. The kernel discriminant analysis is used to exploit the discriminatory capability of the local matching scores non-linearly in a high-dimensional feature space. The training data size issue has been raised and handled in this paper. The results show that the fusion framework improves the verification performance significantly.

Acknowledgement

The work presented in this paper was supported in part by the European Commission under Contract 3D FACE, a European Integrated Project funded under the European Commission IST FP6 program.

References

[1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

[2] J. R. Beveridge, D. S. Bolme, B. A. Draper, and M. Teixeira. The CSU face identification evaluation system: Its purpose, features, and structure. *Machine Vision and Applications*, 16(2):128–138, 2005.

[3] H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana. Discriminant common vectors for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):4–13, 2005.

[4] L. F. Chen, H. Y. M. Liao, M. T. Ko, J. C. Lin, and G. J. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33:1713–1726, 2000.

[5] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 2nd edition, 1990.

[6] C. Kotropoulos, A. Tefas, and I. Pitas. Frontal face authentication using morphological elastic graphmatching. *IEEE Transactions on Image Processing*, 9(4):555–560, 2000.

[7] M. Kyperountas, A. Tefas, and I. Pitas. Weighted piecewise LDA for solving the small sample size problem in face verification. *IEEE Transactions on Neural Networks*, 18(2):506–519, 2007.

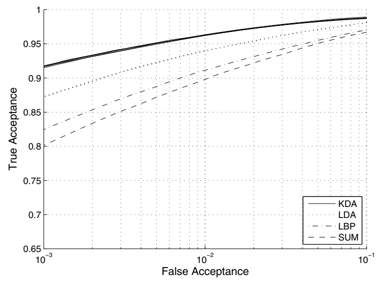
[8] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller. Fisher discriminant analysis with kernels. In Y. H. Hen, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop*, pages 41–48. IEEE Press, Piscataway, NJ, 1999.

[9] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 947–954, San Diego, CA, 2005.

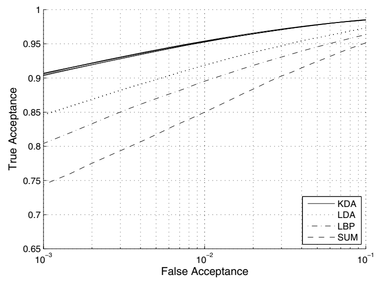
[10] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, 2001.

[11] R. Snelick, U. Uludag, A. Mink, M. Indovina, and A. Jain. Large-scale evaluation of multimodal biometric authentication using state-of-the-art systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3), 20005.

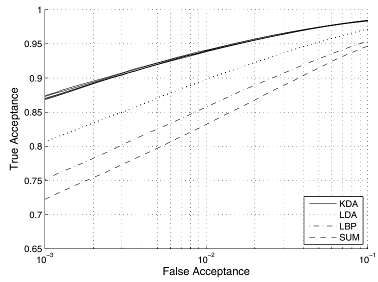
[12] J. Yang, A. F. Frangi, Z. Jin, and J. Y. Yang. Essence of kernel Fisher discriminant: KPCA plus LDA. *Pattern Recognition*, 37:2097–2100, 2004.



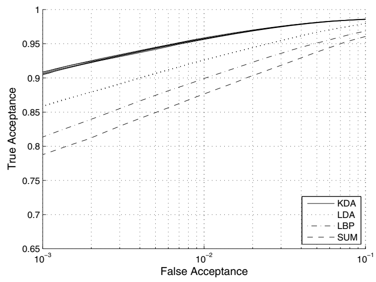
(a) Paritition 1



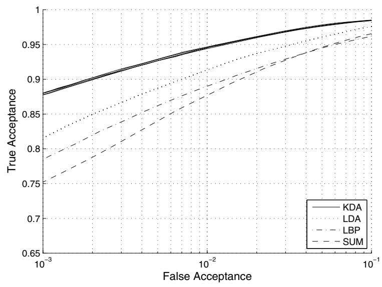
(b) Paritition 2



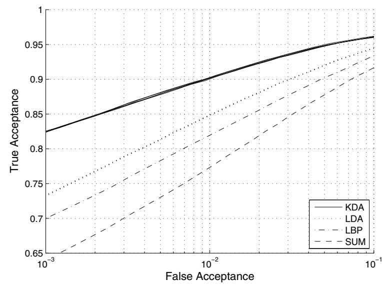
(c) Paritition 3



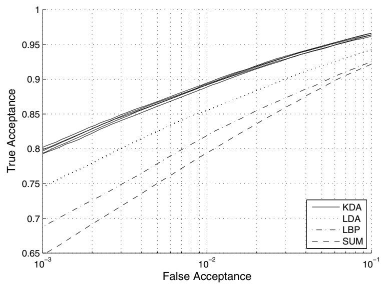
(d) Paritition 4



(e) Paritition 5



(f) Paritition 6



(g) Paritition 7

Figure 2. ROC curves from the experiments carried out on different image partitions.

Partition	K.Max (%)	K.Min (%)	L.Min (%)	LBP (%)	SUM (%)
1	**2.51**	**2.38**	3.49	4.60	4.94
2	**2.96**	**2.91**	4.34	5.29	6.53
3	**3.39**	**3.28**	4.86	6.30	6.89
4	**2.68**	**2.60**	3.89	5.00	5.80
5	**3.15**	**3.04**	4.32	5.41	5.32
6	**5.19**	**5.07**	6.84	7.78	8.89
7	**5.39**	**5.19**	7.06	8.30	8.50

Table 2. EERs calculated from the tests on different image partitions. In the table, K = KDA, L = LDA, Max = maximal rate and Min = minimal rate.

[13] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang. Local Gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition. In *Proceedings of IEEE International Conference on Computer Vision*, pages 786 –791, Beijin, China, 2005.

[14] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys (CSUR)*, 35(4):399–458, 2003.

[15] J. Zou, Q. Ji, and G. Nagy. A comparative study of local matching approach for face recognition. *IEEE Transactions on Image Processing*, 16(10):2617–2628, 2007.