Kinship verification through facial images using multiscale and multilevel handcrafted features

Abdelhakim Chergui, Salim Ouchtati, Sébastien Mavromatis, Salah Eddine Bekhouche, Jean Sequeira, Fadi Dornaika

Journal of Electronic Imaging

JElectronicImaging.org

Kinship verification through facial images using multiscale and multilevel handcrafted features

Abdelhakim Chergui
Salim Ouchtati
Sebastien Mavromatis
Salah Eddine Bekhouche
Jean Sequeira
Fadi Dornaika

SPIE. IS&T imaging.org

Abdelhakim Chergui, Salim Ouchtati, Sebastien Mavromatis, Salah Eddine Bekhouche, Jean Sequeira, Fadi Dornaika, "Kinship verification through facial images using multiscale and multilevel handcrafted features," *J. Electron. Imaging* **29**(2), 023017 (2020), doi: 10.1117/1.JEI.29.2.023017

Kinship verification through facial images using multiscale and multilevel handcrafted features

Abdelhakim Chergui,[a,*] Salim Ouchtati,[a] Sebastien Mavromatis,[b] Salah Eddine Bekhouche,[c,d] Jean Sequeira,[b] and Fadi Dornaika[e,f]

[a]University of August 20th, 1955-Skikda, Electronics Research Laboratory of Skikda, Skikda, Algeria

[b]Aix Marseille University, Université de Toulon, National Committee for Scientific Research, Computer Science and Systems Laboratory, Marseille, France

[c]University Bourgogne Franche-Comté, Université de technologie de Belfort Montbéliard, Laboratoire Connaissance et Intelligence Artificielle Distribuées, Unités Mixtes de Recherche, Belfort, France

[d]University of Djelfa, Department of Electrical Engineering, Djelfa, Algeria

[e]IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

[f]University of the Basque Country Universidad del País Vasco/Euskal Herriko Unibertsitatea, San Sebastian, Spain

Abstract. We address kinship verification, which is a challenging problem in computer vision and pattern discovery. It has several applications, such as organizing photoalbums, recognizing resemblances among humans, and finding missing children. We present a system for facial kinship verification based on several kinds of texture descriptors (local binary patterns, local ternary patterns, local directional patterns, local phase quantization, and binarized statistical image features) with pyramid multilevel (PML) face representation for feature extraction along with our proposed paired feature representation and our proposed robust feature selection to reduce the number of features. The proposed approach consists of the following three main stages: (1) face preprocessing, (2) feature extraction and selection, and (3) kinship verification. Extensive experiments are conducted on five publicly available databases (Cornell, UB KinFace, Family 101, KinFace W-I, and KinFace W-II). Additionally, we provided a wide experiment for each stage to find the best and most suitable settings. We present many comparisons with state-of-the-art methods and through these comparisons, it appears that our experiments show stable and good results. [DOI: 10.1117/1.JEI.29.2.023017]

Keywords: Kinship verification; pyramid multilevel; feature representation; feature selection; support vector machines.

Paper 190828 received Sep. 8, 2019; accepted for publication Mar. 17, 2020; published online Apr. 1, 2020.

1 Introduction

Automatic face analysis based on images has always been an important subject of study in the communities of pattern recognition and artificial intelligence. Facial images contain much information about the person they belong to, such as identity, age, gender, ethnicity, and expression. For this reason, the analysis of facial images has many applications in real-world problems such as face recognition, demographic estimation, facial expression recognition, and kinship verification.

The recognition of kinship among people has mainly concentrated on studying the similarity between human faces. Indeed, there are four main types of kinship relationships: father–daughter (F–D), mother–son (M–S), father–son (F–S), and mother–daughter (M–D) relationships. Recently, other kinship relationship types have emerged, such as the relationship between

*Address all correspondence to Abdelhakim Chergui, E-mail: a.chergui@univ-skikda.dz

1017-9909/2020/$28.00

grandparents and grandchildren. The recognition of kinship is a very challenging topic since the face conveys different facial features.

Currently, the recognition of the relationships between blood relatives has become an active area of research, and it has many applications, such as organizing a photoalbum, annotating images, and identifying lost or wanted people. Furthermore, the determination of kinship is no longer limited exclusively to genetic analysis; it cannot be used in many situations, but it is now extending to the field of biometrics and remote surveillance. The kinship verification problem has many difficulties, such as variations in age, gender, ethnicity, resemblances between persons without a real relationship, and differences in facial attributes between persons from the same family.

In this paper, we propose an automatic facial kinship verification system that is composed of three parts: (1) face preprocessing, (2) feature extraction, and (3) kinship verification. In the face preprocessing stage, we localize faces in images, rectifying the two-dimensional (2-D) pose of each face and crop the region of interest. Then we extract the features by a texture descriptor using a pyramid multilevel (PML) face representation to obtain two feature vectors. These two feature vectors are transformed by the used features representation equation into a single features vector. These are then selected using our proposed feature selection method to obtain the best features and omit the bad features. The best features are fed to the support vector machines (SVM) classifier to determine whether there is a kinship.

The main contributions of this paper are as follows.

- The evaluation of the performance of different texture descriptors: local binary patterns (LBP), local ternary pattern (LTP), local directional pattern (LDP), local phase quantization (LPQ), and binarized statistical image features (BSIF), along with the influences of other steps in the feature extraction stage.
- We investigated the efficiency of three different face image representations when combined with the above texture descriptors. The number of extracted features can be increased by varying the geometric layout of the face image representation in multiblock (MB), multilevel (ML), and PML.
- Our method shows the effectiveness of PML-based descriptors as a face descriptor in the feature extraction stage and the superiority of the PML-based LPQ descriptor.
- We investigated the best pair fusion scheme allowing fusion of the image paired features.
- We introduce an innovative robust feature score for feature ranking. This feature score is based on the use of the difference between two known weights. The new score can further improve the result of the kinship verification system.
- Many experiments are performed on five public databases (Cornell, UB KinFace, Family 101, KinFace W-I, and KinFace W-II) with good and stable results of the proposed approach compared with most of the state-of-the-art approaches.

The remainder of this paper is organized as follows. In Sec. 2, we summarize the existing techniques of facial kinship verification. In Sec. 3, we introduce our approach. In Sec. 4, we present the different databases and their protocols. The experimental results are given in Sec. 5. In Sec. 6, we present the conclusion and some perspectives.

2 Related Work

Many computer vision researchers have investigated the problem of kinship verification. Many approaches have been proposed and classified into two categories: kinship-verification-based and metric-learning-based. In our research, we categorize these approaches into three categories: image-texture-based, manifold-learning-based, and deep-learning-based.

2.1 *Image-Texture-Based*

Early kinship verification approaches usually extract handcrafted features using texture descriptors from facial images and then train these features using a classifier. These approaches have been used in many visual analysis applications. Among their strengths, they are suited for real-time applications, fast and easy implementation, and have low computational cost.

On the other hand, they are vulnerable to profile faces and wild poses and are considered classic approaches.

Fang et al.[1] proposed a system for kinship verification based on a pictorial structure model (PSM). Additionally, they introduced the Cornell database. First, they extracted 22 kinds of facial features composed of colors, facial parts, facial distances, and histograms of gradient features. Then they calculate the differences between feature vectors of the corresponding parents and children and applied *k*-nearest neighbors and SVM classifiers. Zhou et al.[2] introduced a new private database and its images are under uncontrolled conditions. They proposed a spatial pyramid learning-based feature descriptor that utilized both local and global information and they used SVM for the classification phase. The obtained results were promising.

Another interesting work was proposed by Shao et al.,[3] where they used version 2 of the UB KinFace database to verify kinship based on robust local Gabor filters for extracting genetic-invariant features. In other words, a metric and transfer subspace learning were adopted to bridge the discrepancy between children and their old parents. Kohli et al.[4] proposed encoding kinship similarity through a self-similarity descriptor and formalized kinship verification as a two classification problem. They applied their method on the IIITD kinship database, which was annotated with respect to the particular kinship relation, ethnicity, and gender. Chergui et al.[5] proposed an approach based on the ML-LPQ and ML-LDP features and they applied their method on the Cornell and UB KinFac databases. Additionally, they proposed another approach in Ref. 6 based on the LBP and BSIF feature descriptors and the PML feature representation; they applied their method on both the Cornell and UB KinFace databases.

2.2 *Manifold-Learning-Based*

Manifold-learning-based approaches have been motivated by the idea of modeling high-dimensional data using an approximate low-dimensional submanifold of the original space. They have good performance, however, they have high computational costs.

Xia et al.[7] used another database called UB KinFace, which contains images of child, young parent, and old parent faces and used an extended transfer subspace learning method to mitigate the enormous divergence of distributions between children and old parents. An intermediate distribution was used to close the bridge and reduce the divergence between the source distributions.

Lu et al.[8] proposed a neighborhood repulsed metric learning (NRML) method for kinship verification. In addition, they proposed a multiview NRML (MNRML) method to seek a common metric distance to better use the multiple descriptor features and they applied their method on the KinFaceW-I and KinFaceW-II datasets. Hu et al.[9] proposed a large margin multimetric learning method and applied their method on the KinFaceW-I and KinFaceW-II datasets.

Yan et al.[10] proposed a discriminative multimetric learning (DMML) method for kinship verification. First, they extracted multiple features using different face descriptors; then, they jointly learned multiple distance metrics for those descriptor features under which the probability of a pair of face images where the kinship relation had a smaller distance than the pair that had no kinship relation. In their work, they applied their method to two databases: Cornell and UB KinFace databases.

Zhou et al.[11] proposed ensemble similarity learning (ESL). First, they introduced a sparse bilinear similarity function to model the relative of the encoded properties in kin data. The similarity function parameterized by a diagonal matrix enjoys superiority in computational efficiency, making it more practical for real-world high-dimensional kinship verification applications.

Liang et al.[12] developed a framework of weighted graph embedding-based metric learning (WGEML) for facial kinship verification. They extract four types of features: LBP, histogram of oriented gradients, scale-invariant feature transform, and visual geometry group face (VGG-FACE). Then they constructed an intrinsic graph and two penalty graphs to characterize the intraclass compactness and interclass separability for each feature representation. They conducted extensive experiments on the KinFaceW-I, KinFaceW-II, and TSKinFace databases.

Yan et al.[13] proposed a prototype-based discriminative feature learning (PDFL) method for kinship verification. This method aims to learn discriminative mid-level features where they constructed a set of face samples with unlabeled kinship relations from a wild dataset that is considered as the reference set. Then each sample in the training face kinship dataset is

represented as a mid-level feature vector, where each entry corresponds to decision value from one SVM. They applied their method on both the Cornell and UB KinFace databases.

2.3 *Deep-Learning-Based*

Deep learning approaches mainly use convolutional neural networks (CNN), which is a type of feed-forward artificial neural networks in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. Its strengths include achieving good and stable results, suitability for real-time applications, and immunity against facial poses; the downside is the high computational cost.

Zhang et al.[14] proposed extracting high-level features based on deep CNN. These features are produced from the neuron activations of the last hidden layer and then fed into a softmax classifier to verify the kinship of two persons. They applied their method on the KinFaceW-I and KinFaceW-II databases.

Kohli et al.[15] proposed a hierarchical kinship verification via a representation learning framework to learn the representation of different face regions in an unsupervised manner. They proposed an approach for feature representation termed filtered contractive deep belief networks and applied their method on five databases: Cornell, UB KinFace, KinFaceW-I, KinFaceW-II, and WVU Kinship.

Dehghan et al.[16] proposed an algorithm using deep learning that fuses the features and metrics discovered via gated autoencoders with a discriminative neural network layer. They further analyzed the correlation between these automatically detected features and those found in anthropological studies. They applied their method on the KinFaceW-I and KinFaceW-II databases. Wang et al.[17] proposed a deep kinship verification model in which they integrated a deep learning architecture into metric learning to select nonlinear features, which can find the appropriate project space to ensure that the margin of negative pairs is as large as possible and the margin of positive pairs is as small as possible. They applied their method to the KinFaceW-I and KinFaceW-II databases.

Chergui et al.[18] proposed another approach based on the selected deep features of the VGG-FACE descriptor using the Fisher score. They applied their approach on five databases (Cornell, UB KinFace, Family 101, Kinface W-I, and KinFace W-II). Wang et al.[19] proposed a young cross-generation model for kinship verification. They used a deep architecture with a newly designed sparse discriminative metric loss for features extraction and applied their method on the (families in the wild) databases.

3 Kinship Verification Approach

The proposed approach of kinship verification is the operation of using two people's faces to determine whether there is a familial relationship between them. Our proposed method consists of three stages: (1) face preprocessing: for detecting and cropping human faces from an input image; (2) feature extraction: using different descriptors (LBP, LDP, LTP, LPQ, and BSIF), and face representation: with different methods (MB, ML, and PML), (3) classification stage: feature representation and normalization, feature selection (Fisher score, *t*-test, Kullback–Leibler (KL), and the proposed feature selection), and the decision of the kinship verification with an SVM classifier. Figure 1 illustrates the general structure of the proposed framework.

3.1 *Face Preprocessing*

Face preprocessing is a very important stage and it consists of three steps: (I) face detection, (II) eye detection, and (III) face cropping.

(I) We apply the cascade object detector, which uses the Viola–Jones algorithm[20] to detect people's faces. (II) We detect the right and left eye positions using the ensemble of regression trees algorithm.[21] For pose correction, we apply 2-D transformation based on the eye center to correct the pose.[22] (III) To crop the face region of interest, we rescale the face image via normalizing the distance between the new coordinates of the two eyes and then crop the image based on

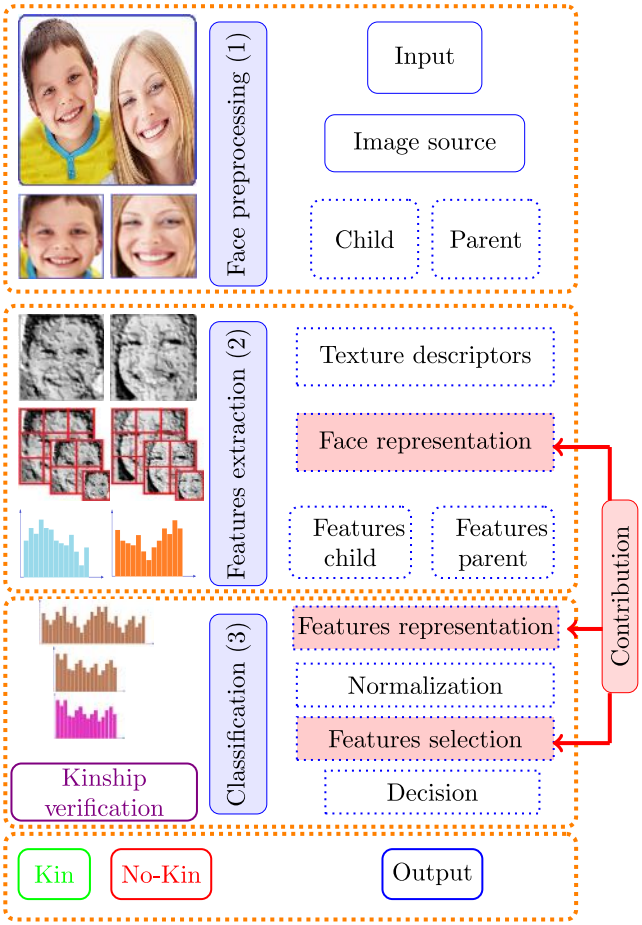


Fig. 1 General structure of the proposed approach.

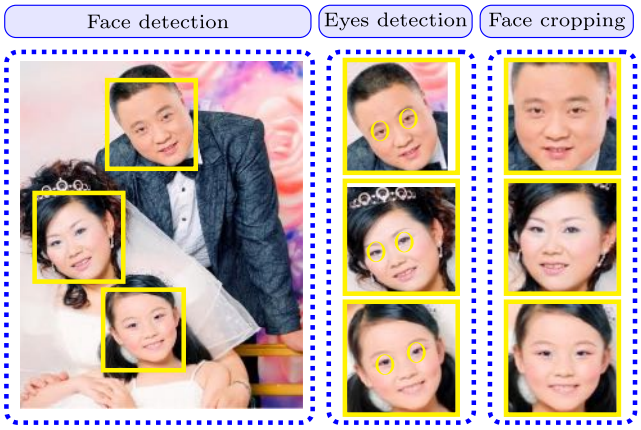


Fig. 2 Face preprocessing steps.

these coordinates.[23] Finally, the cropped facial images are resized to 224×224. Figure 2 presents the steps in this stage.

3.2 *Feature Extraction*

The feature extraction stage has been the most studied topic among the remaining stages due to its effectiveness on facial kinship verification system performance. We divided our feature

extraction stage into two parts. In the first part, we present the different texture descriptors used in our work. In the second part, the different face representations are illustrated.

3.2.1 *Image texture descriptor*

In the image classification field, there are many image texture descriptors that can be used in facial kinship verification or that have already been used in this field. In our work, we use five widely known image texture descriptors: LBP, LTP, LDP, LPQ, and BSIF.

A. *LBPs.* The LBP was proposed by Ojala et al.[24] The LBP is one of the most well-known descriptors, which has led to the development of many descriptors. The LBP is a local descriptor that extracts the local features from image I by thresholding the value of each pixel with its neighborhood pixels using the function $S(z)$, which is defined as

$$S(z) = \begin{cases} 1, & z \geqslant 0 \\ 0, & \text{otherwise} \end{cases}, \tag{1}$$

where z is the the result of subtracting the center pixel value from the intended neighborhood pixel value. The LBP is denoted by $\text{LBP}_{P,R}$, where P is the number of neighborhoods that are equally spaced on a circle of radius R from the center pixel. Each thresholding process between the center pixel value and a neighberhood produces a binary value. The number of bins is 2^P, so the bin value is between 0 and 2^{P-1}. After the thresholding of the pixel and its neighborhoods, a histogram is used to accumulate the occurrence of the various bins over a region by the following equation:

$$\text{LBP}(i_c, j_c) = \sum_{n=0}^{7} S(m_n - m_c)2^n, \tag{2}$$

where i_c, j_c, m_c, and m_n are the coordinates of the intended pixel, the value of the intended pixel, and the value of the neighborhood pixel. Figure 3 illustrates the LBP encoding process for $\text{LBP}_{8,1}$.

$\text{LBP}^{U}_{P,R}$ is the uniform pattern extension of LBP. This extension is used to reduce the length of the feature vector. It was inspired by the fact that some binary patterns occur more commonly in textured images than others. LBP is considered uniform if the binary pattern contains at most two binary transitions (0–1 or 1–0). The number of LBP histogram bins (N_{bins}) can be calculated using the following equation:

$$N_{\text{bins}} = P \cdot (P - 1) + 3. \tag{3}$$

B. *LTP.* The LTP was proposed by Liao[25] The LTP, a descriptor inspired by the LBP descriptor, is a local descriptor that extracts the local features from image I by thresholding the value of each pixel $I(x, y)$ with its neighborhood pixels. For the LBP descriptor, we have two possibilities when comparing pixels with a central pixel (>0 and <0), but in the LTP there are three possibilities in a zone of width t, around m_c pixels are quantified to zero, pixels above this are quantified to $+1$, and pixels below it to -1, i.e., the indicator $S(z)$ is replaced with a three-valued function using the following equation:

$$S'(m_c; m_n; t) = \begin{cases} 1, & m_n > m_c + t \\ 0, & m_n > m_c - t \text{ and } m_n < m_c + t \\ -1, & m_n < m_c - t \end{cases}, \tag{4}$$

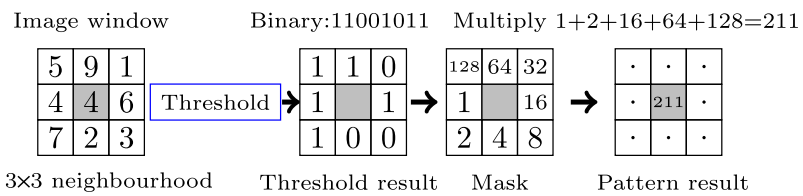


Fig. 3 Illustration of the basic LBP operator.

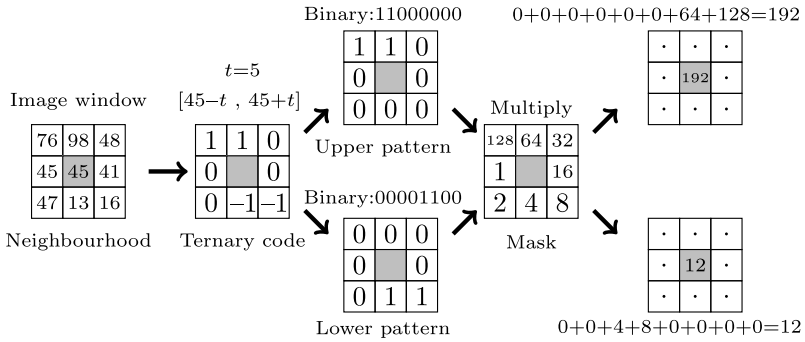


Fig. 4 Illustration of the LTP operator.

and the binary LBP code is replaced by a ternary LTP code. Here, t is a user-specified threshold—so LTP codes are more resistant to noise, but no longer strictly invariant to gray-level transformations. The LTP encoding procedure is illustrated in Fig. 4. Here, the threshold t was set to 5, so the tolerance interval is [40; 50].

C. *LDP.* The LDP was proposed by Jabid et al.[26,27] The LDP is a local texture pattern that computes the edge response values in eight directions at each pixel position and encodes the texture using Kirsch masks in eight different orientations centered on its position $M_0, M_1, M_2, M_3, \ldots, M_7$. Figure 5 shows these masks.

For each pixel of an input image, the obtained result is eight edge response values $m_0, m_1, \ldots, m_7$. Actually, not all the responses are equivalently important, where the k most prominent directions are selected. The top k directional bit responses are set to 1, and the rest of the directional bits become 0. Therefore, LDP code for each pixel is obtained using the following equation:

$$\mathrm{LDP}_k = \sum_{i=0}^{7} S(m_i - m_k) \cdot 2^i, \tag{5}$$

$$S(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{otherwise} \end{cases}. \tag{6}$$

After obtaining the LDP code for all the pixels (i, j), Eq. (7) gives the histogram obtained by LDP:

$$H(\tau) = \sum_{r=1}^{M} \sum_{c=1}^{N} f[\mathrm{LDP}_k(i, j), \tau], \tag{7}$$

$$f(a, \tau) = \begin{cases} 1, & \text{if } z = \tau \\ 0, & \text{otherwise} \end{cases}, \tag{8}$$

M_0 $\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$ M_1 $\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$ M_2 $\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$ M_3 $\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$

M_4 $\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$ M_5 $\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$ M_6 $\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$ M_7 $\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$

Fig. 5 Eight-directions Kirsch edge masks (LDPs).

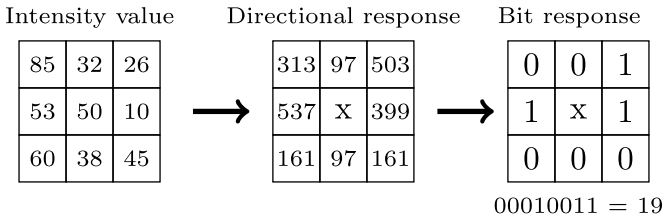


Fig. 6 Illustration of the LDP operator.

where τ is the LDP code value. Equation (9) gives the number of features (N_{bins}) extracted by LDP_k:

$$N_{\text{bins}} = \frac{8!}{k! \cdot (8-k)!}. \tag{9}$$

Figure 6 illustrates the LDP encoding process for example $k = 3$.

D. *LPQ*. LPQ was originally proposed by Ojansivu and Heikkila.[28] LPQ is a texture descriptor based on the application of short-time Fourier transform (STFT). It uses the short-term Fourier transform STFT 2-D computed over a rectangular $M \times M$ neighborhood N_x centered at each pixel position x of the image $f(x)$ defined by

$$F(u,x) = \sum_{y \in N_x} f(x-y) e^{-j2\pi u^{\mathrm{T}} y} = w_u^{\mathrm{T}} f_x, \tag{10}$$

where w_u is the basis vector of the 2-DDFT at frequency u (a 2-D vector) and fx is another vector containing all M^2 image samples from N_x.

In LPQ, only four complex coefficients are considered, corresponding to 2-D frequencies: $u_1 = [a, 0]^{\mathrm{T}}$, $u_2 = [0, a]^{\mathrm{T}}$, $u_3 = [a, a]^{\mathrm{T}}$, and $u_4 = [a, -a]^{\mathrm{T}}$, where a is a sufficiently small scalar. For each pixel, the vector obtained is represented by the following equation:

$$F_x = [F(u_1,x), F(u_2,x), F(u_3,x), F(u_4,x)]. \tag{11}$$

The phase information in the Fourier coefficients is recorded by observing the signs of the real and imaginary parts of each component in $F(x)$. This is done using a scalar quantization defined by

$$q_j = \begin{cases} 1, & \text{if } g_j \geq 0 \\ 0, & \text{otherwise} \end{cases}, \tag{12}$$

where g_j is the j'th component of the vector $G(x) = [\text{Re}\{F(x)\}, \text{Im}\{F(x)\}]$. The resulting eight binary coefficients q_j represent the binary code pattern. This code is converted to decimal numbers between 0 and 255. Froam that, the LPQ histogram has 256 bins. Figure 7 illustrates the LPQ encoding process.

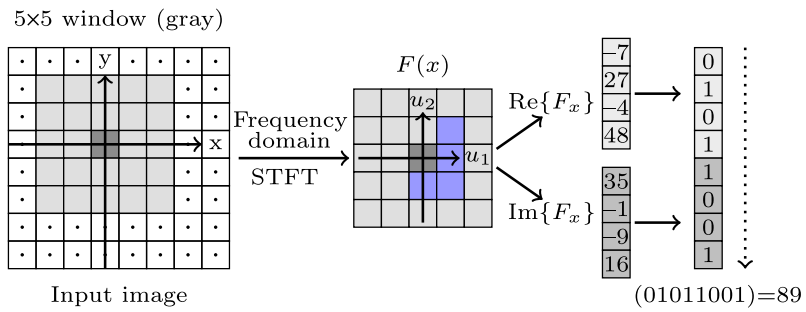


Fig. 7 Illustration of the LPQ operator

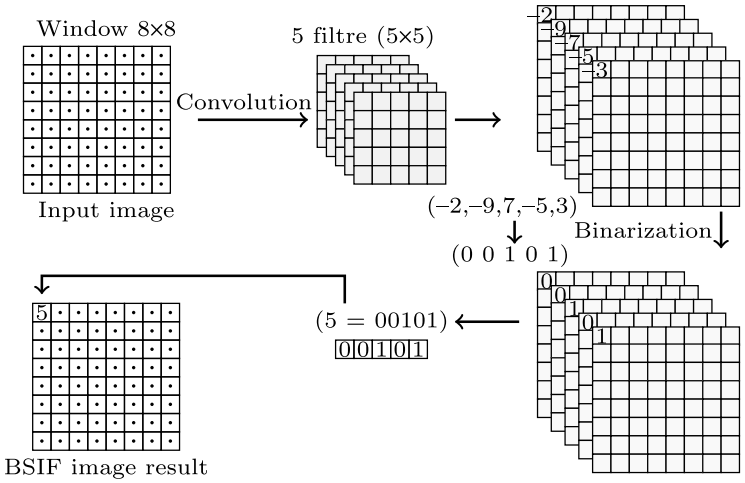


Fig. 8 Illustration of the BSIF operator.

E. *BSIF*. The BSIF is an image texture descriptor proposed by Kannala and Rahtu.[29] The idea behind BSIF is to automatically learn a fixed set of filters from a small set of natural images instead of using hand-crafted filters. The set of filters is learned from a training set of natural image patches by maximizing the statistical independence of the filter responses.

Given an image patch I of size $L \times L$ pixels and a linear filter W_k of the same size, the filter response S_k is obtained by

$$S_k = \sum_{i,j} W_k(i,j) I(i,j) = W_k'^{\mathrm{T}} I', \tag{13}$$

where W_k' and I' are the vectors of size $L \times L$ (vectorized form of the 2-D arrays W_k and I). The binarized feature b_k is obtained by

$$b_k = \begin{cases} 0, & \text{if } S_k \geq 0 \\ 1, & \text{otherwise} \end{cases}. \tag{14}$$

The filters W_k are learned using independent component analysis by maximizing the statistical independence of S_k. The number of histogram bins (N_{bins}) obtained by the BSIF descriptor is calculated using the following equation:

$$N_{\mathrm{bins}} = 2^{N_f}, \tag{15}$$

where N_f is the number of the filters used by BSIF. Figure 8 illustrates the BSIF encoding process.

3.2.2 *Face representation*

A. *MB*. The most common face representation in image processing is a regular grid of fixed size regions that we call MB representation. MB face representation divides the image into n^2 blocks where n is the intended level of MB. Figure 9 illustrates the MB face representations.

B. *ML*. Recently, a similar representation called ML was used in the age estimation and gender classification topics.[30] ML face representation is a spatial pyramid representation constructed by a sorted series of MB representations. The ML face representation level n is constructed from level $1, \ldots, n$ MB representations. Figure 10 illustrates the ML face representations.

C. *PML*. PML face representation is an approach that extracts the local texture features using different descriptors and adopts an explicit pyramid representation of the original image. This pyramid represents the image at different scales. For each such level or scale, a corresponding MB representation is used. PML sub-blocks have the same size, which is determined by the image size and the chosen level. The main idea of the PML is to extract features

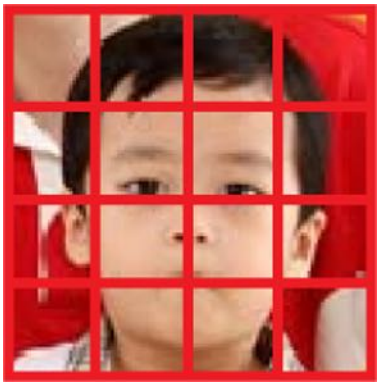


Fig. 9 Multiblock example of four levels.

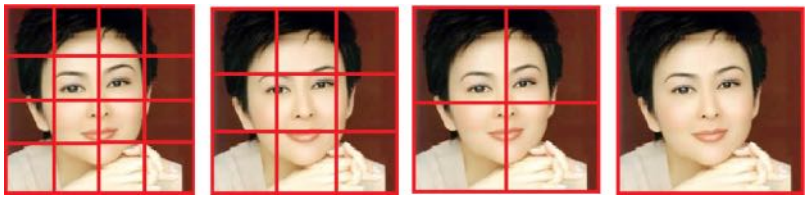


Fig. 10 A multilevel representation adopting four levels.

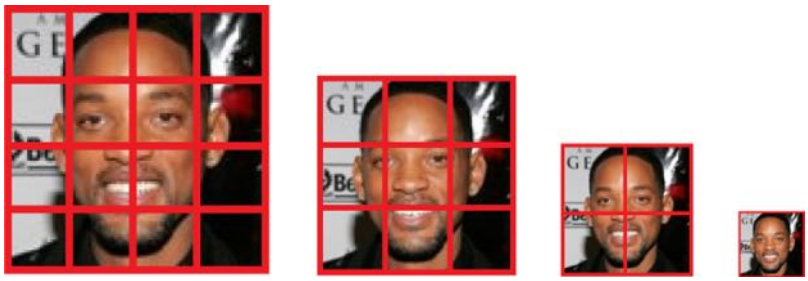


Fig. 11 PML example of four levels.

from different divisions of each pyramid level, assuming that the original image corresponds to level n, and the pyramid levels $(n, n-1, n-2, \ldots, 1)$ are built as follows:

$$\left[w' = w.\frac{(n-1)}{n}, h' = h.\frac{(n-1)}{n}\right]. \tag{16}$$

At each level l, the obtained image is divided into l^2 blocks. Figure 11 illustrates the PML principle.

3.3 *Classification*

The last stage in our proposed approach is the classification stage where the decision for kinship verification is made. It is composed of four steps:

3.3.1 *Paired feature representation*

In this step, the two feature vectors (child/parent) are fused into one feature vector using the following equation:

$$F = \frac{F_{\text{child}} + F_{\text{parent}}}{|F_{\text{child}} - F_{\text{parent}}|}, \tag{17}$$

where F_{parent} and F_{child} are the feature vectors of the parent and the child, respectively, and F is the final feature vector. In Eq. (17), the operations are carried out element by element.

3.3.2 *Normalization*

After having the paired features representation vector, we normalized the final feature vector based on the following equation:

$$F_{\text{norm}}(i) = \frac{F(i)}{\sqrt{\sum_{j=1}^{N} F(j)^2}}. \tag{18}$$

3.3.3 *Feature weighting/selection*

Due to the large number of features obtained from the different face representations, we propose to reduce the number of features using different feature selections along with our proposed approach.

The first feature selection is the Fisher score.[31] It is a linear discriminant approach that quantifies the discriminative power of features. This score is given by

$$W_{\text{Fisher}} = \frac{N_k(m_k - \overline{m})^2 + N_n(m_n - \overline{m})^2}{N_k.\sigma_k^2 + N_n.\sigma_n^2}, \tag{19}$$

where W_{Fisher} is the weight of feature i, $\overline{m}$ is the feature mean, N_X is the number of samples in the kinship class ($k \rightarrow \text{kin}/n \rightarrow \text{nonkin}$), and m_X and σ_X^2 are the mean and the variance of the kinship class in the intended feature. The features are sorted according to their weights.

Another feature selection scheme is given by the t-test. Its weights[32] are based on the absolute value two-sample t-test with a pooled variance estimate. The weight is given by

$$W_{t\text{-test}} = \left| \frac{(m_k - m_n)}{\sqrt{\frac{\sigma_k^2}{N_k} + \frac{\sigma_n^2}{N_n}}} \right|. \tag{20}$$

The KL weight[32] is given by

$$W_{\text{KL}} = \frac{\frac{\sigma_k^2}{\sigma_n^2} + \frac{\sigma_n^2}{\sigma_k^2} - 2}{2} + (m_k - m_n)^2 \frac{\frac{1}{\sigma_k^2} + \frac{1}{\sigma_n^2}}{2}. \tag{21}$$

Our proposed feature weighting approach uses the difference between the t-test and KL weights. This is given by

$$W(i) = W_{t\text{-test}}(i) - W_{\text{KL}}(i), \tag{22}$$

where $W(i)$ is the weight of feature i.

This choice was based on a set of experiments with different mathematical relationships. We found that the best feature weights were obtained when the t-test weight was very high and the KL weight was very low. The experimental evaluation of the proposed weight [Eq. (22)] is presented in Sec. 5.

3.3.4 *Decision*

The last step in the classification uses the SVMs to determine whether there is kinship or not. SVM constructs a hyperplane or set of hyperplanes in a high-dimensional space, which can be used for classification or regression tasks. Intuitively, good separation is achieved by the

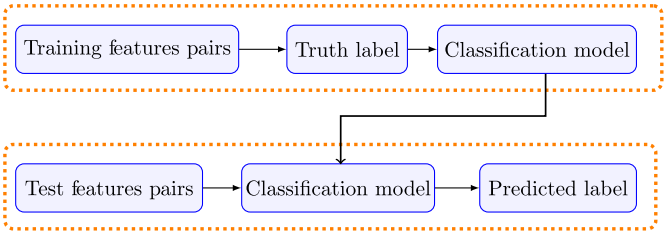


Fig. 12 Evaluation process.

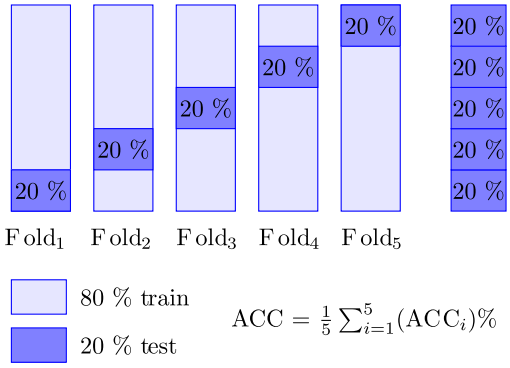


Fig. 13 Fivefold cross validation.

hyperplane that has the largest distance to the nearest training data point of any class (the so-called functional margin), since, in general, the larger the margin is, the lower the generalization error of the classifier.[33]

We used the binary SVM classifier to train and test our proposed approach. The two binary classes are either a kinship relationship or not, which are represented by 1 and 0, respectively. The evaluation process has two steps. The first step consists of training the SVM models using the training pairs. The second step predicts the class of all pairs in the test set.

We adopt a fivefold cross-validation scheme for evaluation. In this scheme, 80% of the data are used for training and the remaining data are used for testing. Figure 12 illustrates the two steps of the evaluation process.

K-fold cross validation is a standard protocol that allows the evaluation of a given algorithm or a classifier.

It proceeds as follows. First, the whole dataset is split into k subsets. One subset is chosen as a test set and the remaining $k-1$ subsets are used for training. This process is repeated k times to ensure that all data are used in the test. The final performance is set to the average performance over the k folds.

Figure 13 illustrates the principle of k-fold cross validation when k is equal to five. The classification algorithm is trained five times on as many different data segments and five independent tests are conducted thereafter. The number of subsets can be modified to meet the particular needs of an experiment.

4 Databases and Protocols

In this section, we discuss the different publicly available databases and their related details. Figure 14 shows some images with examples of positive and negative pairs from five databases (Cornell KinFace, UB KinFace, KinFaceW-I, KinFaceW-II, and Family 101).

4.1 *Cornell KinFace (2010)*

The Cornell KinFace database is the first database that was used for kinship verification. It was created by Fang et al.[1] from the University of Cornell. They collected 150 pairs (300 images)

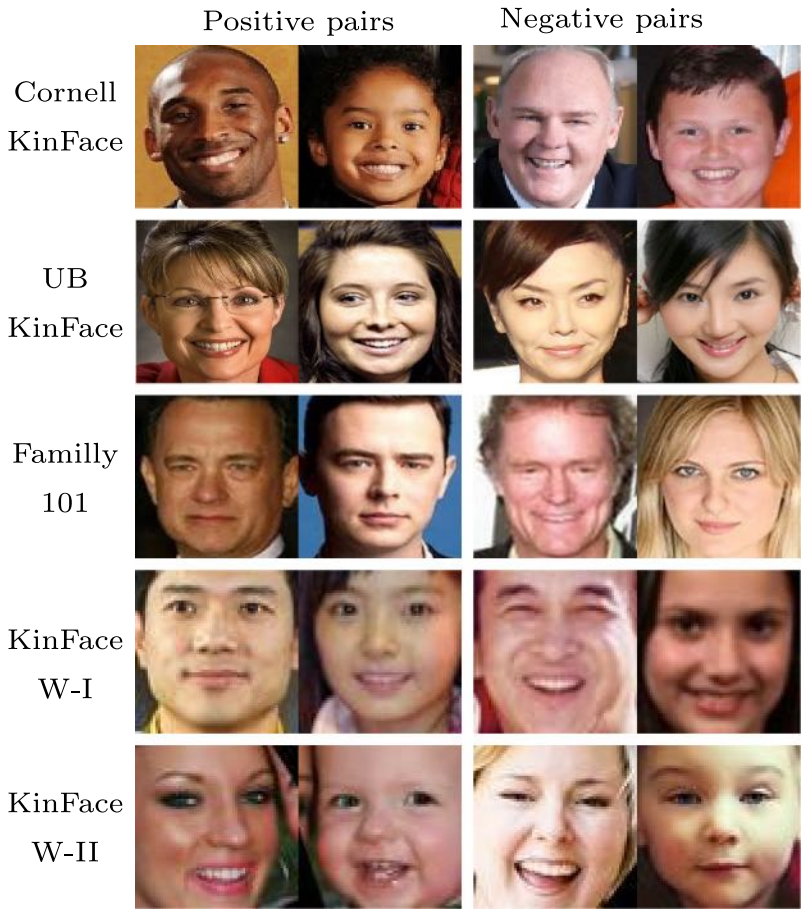


Fig. 14 Examples of positive and negative pairs from used databases.

with variations in demographic attributes. The distribution of the kinship pairs is as follows: 40% (F–S), 22% (F–D), 13% (M–S), and 26% (M–D).

4.2 *UB KinFace (2011)*

The UB KinFace database was created by Shao et al.[3] and was collected from 400 persons and produced 600 static images that were divided into 200 groups where each group was composed of child, young parent, and old parent images. The UB KinFace database was the first database designed for kinship verification with children of different ages and their parents. This database is composed of the following kinship relations: (F–S), (F–D), (M–S), and (M–D).

4.3 *Family 101 (2013)*

The family 101 database was created by Fang et al.;[34] this database contains 101 families, including 206 nuclear families and 607 persons, and contains a total of 14,816 images that were collected from public families but with changes in real family names. It has four kinship relations as follows: 213 (F–S), 147 (F–D), 184 (M–S), and 148 (M–D).

4.4 *KinFaceW-I (2014)*

KinFaceW-I was created by Lu et al.[8] The images in this database were collected from the web and captured under uncontrolled environments in terms of gestures, demographic attributes, lighting, backgrounds, expressions, and partial occlusions. This database contains four kin relations, 156 (F–S), 134 (F–D), 116 (M–S), and 127 (M-D) kinship pairs. The images in this database were aligned and cropped manually.

Table 1 Kinship verification databases used and their related details.

Databases	Images	Pairs	Controlled condition	Year
Cornell KinFace	286	286	No	**2010**
UB KinFace	600	800	No	**2011**
Family 101	14,816	2000	No	**2013**
KinFace W-I	1066	2000	No	**2014**
KinFace W-II	2000	2000	No	**2014**

4.5 *KinFaceW-II (2014)*

KinFaceW-II was created by Lu et al.[8] The images in this database include some celebrity face images as well as their children or parents. These images were collected from the Internet. The KinFaceW-II dataset contains four kin relations (F–S), (F–D), (M–S), and (M–D), 250 pairs of kinship images for each kin relation, and the images in this database were aligned and cropped manually (Table 1).

5 Experimental Results and Discussion

We conducted many experiments using a set of texture descriptors with different face representations (MB, ML, and PML). Furthermore, we quantified the performance using different pair fusion schemes as well as different feature selection schemes including our proposed selection scheme. The experiments were conducted on the kin datasets that are publicly available (Cornell, UB KinFace, Family 101, KinFaceW-I, and KinFace-II) to verify the effectiveness of our proposed kinship verification approach. We present the experimental results in the following sections.

5.1 *Effect of Descriptor Type*

With regard to the feature extraction step, we investigated the effectiveness of the proposed method with different feature descriptors using the PML face representation adopting seven levels. This face representation achieved good results in the facial age estimation problem.[35]

Table 2 shows the results obtained for each descriptor on the used databases. From these obtained results, we can observe that the LPQ descriptor achieved the highest accuracy. It was followed by the BSIF descriptor. The LTP and LDP descriptors achieved low accuracy compared to the first descriptors (LPQ and BSIF). The LBP descriptor achieved good accuracy, yet its accuracy was still lower than that of the LPQ and BSIF descriptors and much better than that of the LTP and LDP descriptors.

LPQ was originally linked to the spectral analysis of the image content. The use of the PML face representation on the spectral information coded in the LPQ descriptor seemed to be more

Table 2 The accuracy (%) on the different descriptors for the used databases.

	LDP	LTP	LBP	BSIF	LPQ
Cornell KinFace	56.10	64.34	88.46	94.41	**95.21**
UB KinFace	57.88	73.38	90.50	93.13	**94.46**
KinFace W-I	60.63	66.94	90.15	92.21	**93.20**
KinFace W-II	59.16	64.45	87.30	89.15	**91.40**
Family 101	57.47	61.41	76.24	84.67	**90.38**

useful than other descriptors such as LBP, LDP, and BSIF that directly encode local texture patterns. Based on these results, we conclude that the LPQ descriptor is the best for the feature extraction stage. Thus we use it in the following experiments.

5.2 *Effect of Face Representation*

To study the effect of face representation, we compared the performance obtained by three face representations: MB, ML, and PML. Each was tested with a level number that varied from 1 to 10. The descriptor used was the LPQ descriptor. Figure 15 summarizes the performance associated with different face representations. It also presents the number of features as a function of the number of levels of the face representation. These results were obtained from the KinFace W-II database. The curves associated with the other databases have a similar shape.

According to our experiments, we observe that by increasing the number of levels of the face representation, the number of features of the final descriptors increases. In return, the performance improves and the feature extraction stage becomes time consuming. For each level, the number of features for the PML and ML representations are identical and larger than that associated with the MB representation, which achieved a lower performance compared to that of ML and PML. Despite the fact that the ML and PML face representations have the same number of features for a given number of levels, the PML representation achieved better results than the ML representation due to the advantage of PML in terms of feature weight balance. We also observe that after level 7, the results improved slightly despite the increase in the number of features. We conclude that the use of PML with seven levels achieves good results with a reasonable number of features and computational cost.

5.3 *Effect of the Image Paired Feature Representation*

Since the input to the system is a pair of images, it is worth studying different schemes to fuse the features of the two images into a single feature descriptor. We conducted several experiments for the paired feature representation step using different combinations such as absolute difference [Eq. (23)], sum [Eq. (24)], division [Eq. (25)], and multiplication [Eq. (26)]. We tested the following pair fusion schemes:

$$F_1 = |F_{\text{child}} - F_{\text{parent}}|, \tag{23}$$

$$F_2 = F_{\text{parent}} + F_{\text{child}}, \tag{24}$$

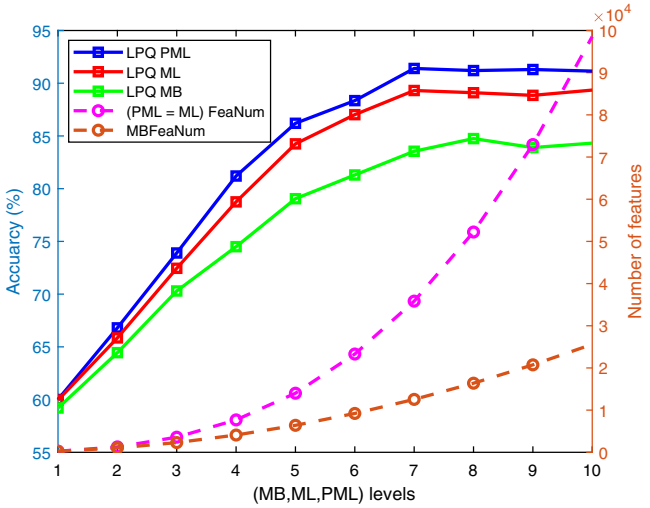


Fig. 15 The accuracy (%) and number of features on the different levels of the face representation.

Table 3 Accuracy (%) obtained with different pair fusion schemes for the KinFace-WII database.

	F_1	F_2	F_3	F_4	F_5	F_6
LDP	66.10	51.86	64.05	59.10	59.60	56.16
LTP	72.35	52.64	71.25	70.15	71.35	62.45
LBP	80.40	55.15	76.65	69.90	71.55	87.30
BSIF	78.30	57.98	73.30	73.30	65.40	89.15
LPQ	74.30	58.69	77.50	65.85	68.70	**91.40**

$$F_3 = \frac{F_{\text{parent}}}{F_{\text{child}}}, \tag{25}$$

$$F_4 = F_{\text{parent}} \times F_{\text{child}}, \tag{26}$$

$$F_5 = \frac{F_{\text{parent}} \times F_{\text{child}}}{|F_{\text{child}} - F_{\text{parent}}|}, \tag{27}$$

$$F_6 = \frac{F_{\text{parent}} + F_{\text{child}}}{|F_{\text{child}} - F_{\text{parent}}|}. \tag{28}$$

In the above equations, the fusion operates in an element-wise fashion.

Table 3 summarizes the performance of the different pair fusion schemes obtained with the five descriptors: LDP, LTP, LBP, BSIF, and LPQ. The results were obtained on the KinFace-WII database. From this table, we can see that the fusion scheme that used the element-wise absolute difference achieved the best results compared with the fusion schemes: sum, division, and multiplication. For that reason, we included this distance in the two other fusion schemes F_5 and F_6. The performance is significantly increased when using the F_6 scheme with the LBP, BSIF, and LPQ descriptors. It was the best fusion scheme for the LPQ descriptor.

5.4 *Effect of Feature Weighting/Selection*

In this group of experiments, we used and compared different functions (Fisher score, KL score, *t*-test score, and the proposed score) for the feature weighting step. After feature weighting, the fused features were ranked in descending order. Table 4 summarizes the obtained performances using feature weighting and selection for the KinFace W-II database. The original features correspond to LPQ-PML adopting seven levels.

The proposed feature weighting outperformed Fisher weighting, KL weighting, and *t*-test weighting. To have a good idea about the choice of the proposed feature weighting [Eq. (22)], we conducted the following experiment. First, we ranked the KinFace W-II features according to the Fisher weight. Second, we retained the first 1000 ranked features. At this stage, we had 1000 relevant features. Third, for each such feature, we computed its weight according to four different functions: *t*-test, KL, *t*-test + KL, and our proposed weight *t*-test − KL. Figure 16 illustrates the four weights for the retained 1000 features. We can observe that the weight *t*-test − KL [i.e., Eq. (22)] had the largest spread, which indicates that the proposed weight is very sensitive to the feature relevance. However, the spreads associated with either the *t*-test weight alone or the KL weight alone were small explaining their lower performances in detecting the most relevant features.

The Fisher score and *t-test* achieved good performances despite the slight superiority of the Fisher score, which is a well-known feature weighting technique. KL achieved a poor performance compared to that of the other feature weighting schemes. Our idea was to generate a more sensitive feature weight function that was derived from the *t*-test weight and the KL weight, as shown in Eq. (28). The performance is significantly improved when using our proposed feature

Table 4 The accuracy (%) using different feature weighting schemes.

	KL	*t*-test	Fisher score	Proposed
Cornell KinFace	62.01	92.36	94.19	**95.21**
UB KinFace	60.41	88.47	93.46	**94.46**
KinFace W-I	59.78	87.71	92.13	**93.20**
KinFace W-II	58.92	82.61	89.70	**91.40**
Family 101	56.35	76.24	88.97	**90.38**

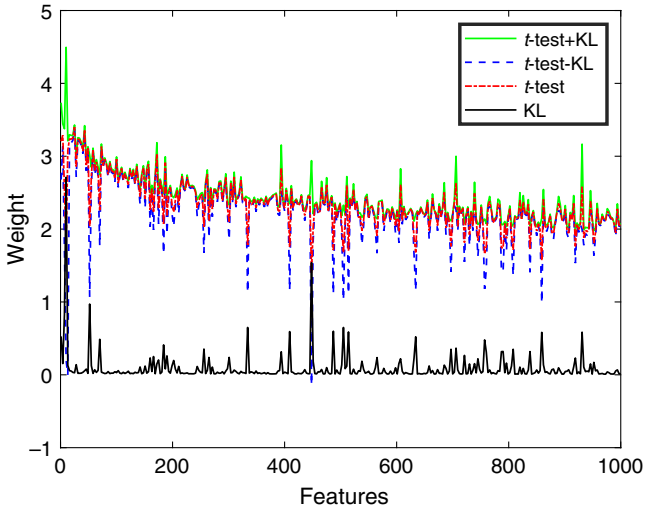


Fig. 16 The estimated weights for 1000 retained featured functions are: *t*-test, KL, *t*-test + KL, and our proposed weighting function *t*-test − KL.

weighting function. Figure 17 shows the superiority of our proposed feature selection in terms of accuracy compared with the other feature weighting schemes.

5.5 *Number of Features Used and CPU Time*

As a first experiment, we used the LPQ descriptor with PML representation adopting seven levels. The corresponding fused descriptor had 38,500 features. We used the total data of 2000 × 38,500 for the KinFaceW-II database without using a feature selection function. We obtained an accuracy of 90.12% in 320.6213 s (CPU time) for the training phase and 11.2164 s (CPU time) for the testing phase. Then we used our proposed feature weighting function to sort the features according to their weight. Finally, we tested several feature dimensions to select the best number of features.

Figure 18 shows both the accuracy of the kinship verification and the CPU time (in seconds), which was needed for training the model and testing as a function of the number of retained features that were obtained after using our proposed feature weighting scheme. The CPU time (training/testing) of the proposed approach increases linearly with the increase in the number of features. Additionally, the accuracy increased exponentially up to 1000 features and then varied as a logarithmic function. Based on that, the choice of 1000 features can be considered a good trade-off between accuracy and computational complexity. These results were obtained with the KinFace W-II database. The curves associated with the other databases have a similar shape. The experiments were carried out on a laptop DELL 7510 Precision (Xeon Processor E3-1535M v5, 8M Cache, 2.90 GHz, 64 GB RAM, GPU NVIDIA Quadro M2000M, Windows 10 using MATLAB R2018b.)

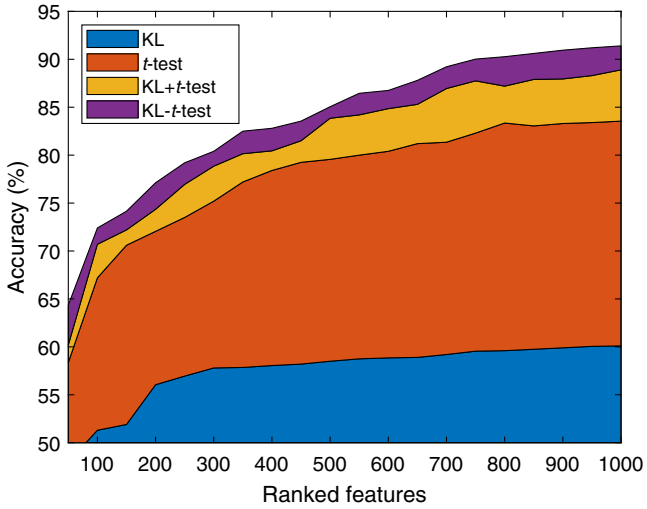


Fig. 17 Obtained accuracy (%) using different weighting schemes: *t*-test, KL, *t*-test + KL, and our proposed feature selection *t*-test − KL.

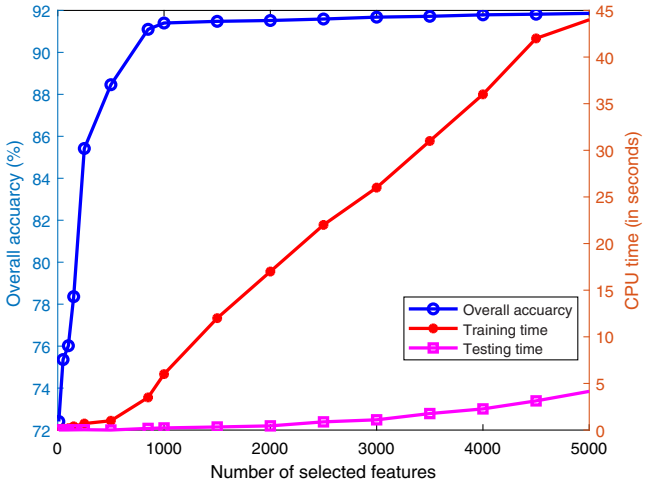


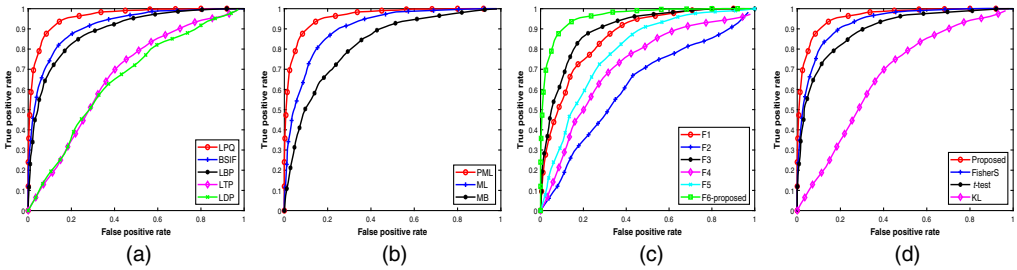
Fig. 18 The accuracy (%) and CPU time (training/test).

5.6 *Best Results and Settings*

Finally, we obtained the best results using PML-LPQ level 7 with the proposed paired feature representation and the proposed feature selection approach. Table 5 shows the performance of the proposed approach with different relationships (F–S), (F–D), (M–S), and (M–D). We

Table 5 The accuracy (%) on the four relationships of the used database

	F_S	F_D	M_S	M_D	Mean
Cornell KinFace	97.56	96.97	96.97	89.36	**95.21**
UB KinFace	95.84	95.58	93.84	92.59	**94.46**
KinFace W-I	93.91	93.30	92.91	92.67	**93.20**
KinFace W-II	92.00	92.00	91.60	90.00	**91.40**
Family 101	91.81	90.29	89.85	89.85	**90.38**



(a) (b) (c) (d)

Fig. 19 ROC curves of different effects on the KinFace W-II database. (a) Texture descriptors. (b) Face representation approaches. (c) Features representation approaches. (d) Feature selection approaches.

Table 6 A comparison of the proposed approach with other approaches

Approaches	Cornell	UB KinFace	KinFace W-I	KinFace W-II	Family 101
PSM 2010[1]	70.67	—	—	—	—
TSL 2011[3]	—	69.67	—	—	—
SSRWF 2012[4]	—	69.67	—	—	—
MNRML 2014[8]	—	—	69.90	76.50	—
DMML 2014[10]	73.50	74.50	72.00	78.00	—
RSBM 2015[36]	—	—	—	—	69.6
ESL 2015[11]	—	—	78.60	75.70	—
PDFL 2015[13]	71.90	67.30	70.10	77.00	—
CNN-pnt 2015[14]	—	—	77.50	88.40	—
SSML 2016[37]	—	—	79.55	80.15	—
S.scoring 2016[38]	81.40	52.20	71.40	80.10	—
EHRMFS 2017[39]	—	—	80.20	80.16	—
NRML 2017[40]	—	—	66.30	78.70	—
DDMML 2017[41]	—	—	83.50	84.30	—
DIEDA 2018[42]	—	—	80.60	88.60	—
DCTNet 2018[43]	—	—	84.75	89.25	—
PML-COV 2018[44]	—	84.5	88.20	88.20	—
LTP*ML 2018[45]	90.02	87.16	82.84	79.06	77.32
WLD*RGB 2018[46]	92.06	87.97	—	—	83.20
VGG-FACE 2019[18]	92.74	90.39	86.49	79.37	84.67
MSIDA 2019[47]	86.87	83.34	—	—	—
WGEML2019[12]	—	—	81.90	82.80	—
(Mixed-D)*MB 2019[48]	84.74	82.89	81.69	80.12	78.16
KML 2019[49]	81.40	75.50	82.80	85.70	—
Proposed	**95.21**	**94.46**	**93.20**	**91.40**	**90.38**

observe that: (i) the easiest relationship to classify is the relationship in which there is a male, whether a child or father. Thus the relationship (F–S) is easier than (F–D), and (M–S) is easier than (M–D). (ii) The hardest relationship to classify is the relationship in which there is a female, whether daughter or mother. Thus (M–S) is harder than (F–S) and (M–D) is harder than (M–S). Thus the easiest relationship to classify is (F–S) and the hardest one to classify is (M–D).

To better visualize the best results and the performance of our approach, the receiver operating characteristic (ROC) curves of different descriptors are shown in Fig. 19. This figure summarizes the ROCs obtained by (a) different descriptors, (b) different face representations, (c) different feature fusion schemes, and (d) different feature weighting schemes. This figure illustrates the ROC curves of the classification results on the KinFace W-II database.

We observe that the use of the LPQ descriptor with PML-7 face representation and the proposed pair fusion scheme that adopted the proposed feature weighting scheme yielded the best performance in terms of the ROC curve. The curves associated with the other databases had similar shapes.

5.7 *Comparison with State-of-the-Art Approaches*

The performance evaluation on several publicly available benchmark databases validates that our approach outperforms existing state-of-the-art approaches. Table 6 illustrates the performance of our proposed kinship verification approach as well as that of some competing approaches. As can be seen in this table, we obtain 95.21%, 94.46%, 93.20%, 91.40%, and 90.38% kinship verification accuracy on Cornell, UB KinFace, KinFace-I, KinFace-II, and Family 101, respectively. This confirms the robustness of our approach on all the databases that were used.

6 Conclusion

In this paper, we described a approach for facial kinship verification based on different descriptors with PML face representation. We also introduced a new paired feature representation and a new feature weighting. We investigated different kinds of effects, including image texture descriptors and face and feature representations. The experimental results showed that our approach provides good performance and stable results on five publicly available databases compared to the previous approaches. In future work, we propose using deep features provided by models such as VGGNet, AlexNet, ResNet, and ImageNet. Moreover, we envision developing CNN architectures for multitask estimation. Additionally, we envision the use of other soft biometric traits, such as age and gender, to improve the kinship verification accuracy.

References

1. R. Fang et al., "Towards computational models of kinship verification," in *17th IEEE Int. Conf. Image Process.*, IEEE, pp. 1577–1580 (2010).
2. X. Zhou et al., "Kinship verification from facial images under uncontrolled conditions," in *Proc. 19th ACM Int. Conf. Multimedia*, ACM, pp. 953–956 (2011).
3. M. Shao, S. Xia, and Y. Fu, "Genealogical face recognition based on UB KinFace database," in *IEEE Conf. Comput. Vision and Pattern Recognit. Workshops*, IEEE, pp. 60–65 (2011).
4. N. Kohli, R. Singh, and M. Vatsa, "Self-similarity representation of Weber faces for kinship classification," in *IEEE Fifth Int. Conf. Biometrics: Theory, Appl. and Syst.*, IEEE, pp. 245–250 (2012).
5. A. Chergui et al., "LPQ and LDP descriptors with ml representation for kinship verification," in *Second Edition Int. Workshop Signal Process. Appl. Rotating Mach. Diagn.*, pp. 1–10 (2018).
6. A. Chergui et al., "Kinship verification using BSIF and LBP," in *Fourth Edition Int. Conf. Signal, Image, Vision and their Appl.* (2018).

7. S. Xia, M. Shao, and Y. Fu, "Kinship verification through transfer learning," in *Proc. Twenty-Second Int. Joint Conf. Artif. Intell.*, pp. 2539–2544 (2011).
8. J. Lu et al., "Neighborhood repulsed metric learning for kinship verification," *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(2), 331–345 (2014).
9. J. Hu et al., "Large margin multi-metric learning for face and kinship verification in the wild," *Lect. Notes Comput. Sci.* **9005**, 252–267 (2014).
10. H. Yan et al., "Discriminative multimetric learning for kinship verification," *IEEE Trans. Inf. Forensics Secur.* **9**(7), 1169–1178 (2014).
11. X. Zhou et al., "Ensemble similarity learning for kinship verification from facial images in the wild," *Inf. Fusion* **32**, 40–48 (2016).
12. J. Liang et al., "Weighted graph embedding-based metric learning for kinship verification," *IEEE Trans. Image Process.* **28**(3), 1149–1162 (2019).
13. H. Yan, J. Lu, and X. Zhou, "Prototype-based discriminative feature learning for kinship verification," *IEEE Trans. Cybern.* **45**(11), 2535–2545 (2015).
14. K. Zhang et al., "Kinship verification with deep convolutional neural networks," in *Proc. Br. Mach. Vision Conf.*, X. Xie, M. W. Jones, and G. K. L. Tam, Eds., BMVA Press, pp. 148.1–148.12 (2015).
15. N. Kohli et al., "Hierarchical representation learning for kinship verification," *IEEE Trans. Image Process.* **26**(1), 289–302 (2017).
16. A. Dehghan et al., "Who do I look like? Determining parent-offspring resemblance via gated autoencoders," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1757–1764 (2014).
17. M. Wang et al., "Deep kinship verification," in *IEEE 17th Int. Workshop Multimedia Signal Process.*, pp. 1–6, IEEE (2015).
18. A. Chergui et al., "Deep features for kinship verification from facial images," in *Proc. Third Int. Conf. Adv. Syst. and Emergent Technol.* (2019).
19. S. Wang, Z. Ding, and Y. Fu, "Cross-generation kinship verification with sparse discriminative metric," *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(11), 2783–2790 (2019).
20. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, IEEE, Vol. 1 (2001).
21. V. Kazemi and S. Josephine, "One millisecond face alignment with an ensemble of regression trees," in *27th IEEE Conf. Comput. Vision and Pattern Recognit.*, IEEE Computer Society, Columbus, pp. 1867–1874 (2014).
22. S. E. Bekhouche, "Facial soft biometrics: extracting demographic traits," PhD Thesis, Faculté des Sciences et Technologies (2017).
23. S. E. Bekhouche et al., "Pyramid multi-level features for facial demographic estimation," *Expert Syst. Appl.* **80**, 297–310 (2017).
24. T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," in *Proc. 12th Int. Conf. Pattern Recognit.*, IEEE, Vol. 1, pp. 582–585 (1994).
25. W.-H. Liao, "Region description using extended local ternary patterns," in *20th Int. Conf. Pattern Recognit.*, IEEE, pp. 1003–1006 (2010).
26. T. Jabid, M. H. Kabir, and O. Chae, "Local directional pattern (LDP)–a robust image descriptor for object recognition," in *7th IEEE Int. Conf. Adv. Video and Signal Based Surveillance*, IEEE, pp. 482–487 (2010).
27. T. Jabid, M. H. Kabir, and O. Chae, "Local directional pattern (LDP) for face recognition," in *Digest Tech. Papers Int. Conf. Consumer Electron.*, IEEE, pp. 329–330 (2010).
28. V. Ojansivu and J. Heikkilä, "Blur insensitive texture classification using local phase quantization," *Lect. Notes Comput. Sci.* **5099**, 236–243 (2008).
29. J. Kannala and E. Rahtu, "BSIF: binarized statistical image features," in *21st Int. Conf. Pattern Recognit.*, pp. 1363–1366 (2012).
30. S. E. Bekhouche et al., "Facial age estimation and gender classification using multi level local phase quantization," in *3rd Int. Conf. Control, Eng. Inf. Technol.*, pp. 1–4 (2015).
31. H. Lohninger, *Teach/Me: Data Analysis*, Springer, Berlin (1999).

32. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed., 58:16, John Wiley & Sons, New York (2001).
33. M. Arumugam et al., "Implementation of two class classifiers for hybrid intrusion detection," in *Int. Conf. Commun. and Comput. Intell.*, pp. 486–490 (2010).
34. R. Fang et al., "Kinship classification by modeling facial feature heredity," in *20th IEEE Int. Conf. Image Process.*, IEEE, pp. 2983–2987 (2013).
35. S. E. Bekhouche et al., "Personality traits and job candidate screening via analyzing facial videos," in *IEEE Conf. Comput. Vision and Pattern Recognit. Workshops*, IEEE, pp. 1660–1663 (2017).
36. X. Qin, X. Tan, and S. Chen, "Tri-subject kinship verification: understanding the core of a family," *IEEE Trans. Multimedia* **17**(10), 1855–1867 (2015).
37. Y. Fang et al., "Sparse similarity metric learning for kinship verification," in *Visual Commun. and Image Process.*, IEEE, pp. 1–4 (2016).
38. M. B. López, E. Boutellaa, and A. Hadid, "Comments on the 'kinship face in the wild' data sets," *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(11), 2342–2344 (2016).
39. C. N. R. Kumar, "Harmonic rule for measuring the facial similarities among relatives," *Trans. Mach. Learn. Artif. Intell.* **4**(6), 29 (2017).
40. H. Yan, "Kinship verification using neighborhood repulsed correlation metric learning," *Image Vision Comput.* **60**, 91–97 (2017).
41. J. Lu, J. Hu, and Y.-P. Tan, "Discriminative deep metric learning for face and kinship verification," *IEEE Trans. Image Process.* **26**(9), 4269–4282 (2017).
42. R. Aliradi et al., "DIEDA: discriminative information based on exponential discriminant analysis combined with local features representation for face and kinship verification," *Multimedia Tools Appl.* 1–18 (2018).
43. A. Tidjani et al., "Deep learning features for robust facial kinship verification," *IET Image Process.* **12**(12), 2336–2345 (2018).
44. A. Moujahid and F. Dornaika, "A pyramid multi-level face descriptor: application to kinship verification," *Multimedia Tools Appl.* **78**, 9335–9354 (2019).
45. A. Chergui et al., "Robust kinship verification using local descriptors," in *Proc. Third Int. Conf. Adv. Technol. and Electr. Eng.* (2018).
46. A. Chergui et al., "Discriminant analysis for facial verification using color images," in *Proc. First Int. Conf. Electr. Eng.* (2018).
47. M. Bessaoudi et al., "Multilinear side-information based discriminant analysis for face and kinship verification in the wild," *Neurocomputing* **329**, 267–278 (2019).
48. A. Chergui et al., "Kinship verification using mixed descriptors and multi block face representation," in *Int. Conf. Networking and Adv. Syst.*, IEEE, pp. 1–6 (2019).
49. X. Zhou et al., "Learning deep compact similarity metric for kinship verification from face images," *Inf. Fusion* **48**, 84–94 (2019).

Abdelhakim Chergui received his master's degree in signals and communications from Mohammed Khider University, Biskra, Algeria, in 2015. Currently, he is a PhD student at the Université of 20 aout 1955 of Skikda. His research interests include pattern recognition, image processing, machine learning, deep learning, biometric techniques, and computer vision.

Salim Ouchtati received his BEng and MSc degrees in electronics from Annaba University, Algeria, in 1994 and 1999, respectively. He obtained his doctorate degree in 2007 in automation and his HDR degree (habilitation to direct the research) in electronics in 2010 from Annaba University. He is currently an associate professor at Skikda University. His research areas are focused mainly on handwritten recognition, artificial intelligence, kinship verification, deep learning, and medical images processing.

Sebastien Mavromatis received his PhD in computer science from the University of Marseille, France, in 2001. He is currently an associate professor at Aix-Marseille University. He is also a member of the Laboratory of Computer Science and Systems (LIS-CNRS UMR 7020). He is regularly involved in many international cooperation projects in China, Peru, and Algeria. His research interests include image analysis, three-dimensional scene reconstruction, and virtual and augmented reality.

Salah Eddine Bekhouche received his PhD in electronics from the University of Biskra, Algeria, in 2017. He is currently a postdoctoral researcher at EPAN Research Group, University of Technology of Belfort-Montbéliard, France. His research interests include computer vision, pattern recognition, and image processing.

Jean Sequeira graduated from the Ecole Polytechnique, Paris, in 1977 and TelecomNational Engineering School, Paris, in 1979. He received his PhD in computer science in 1982 and his accreditation to supervise research in 1987. From 1981 to 1991, he was a researcher at the IBM company. He has been a full professor at Aix-Marseille University since 1991 (exceptional class full professor since 2010). He is an IEEE senior member.

Fadi Dornaika received his PhD in computer science from the Grenoble Institute of Technology, France, in 1995. He is currently a research professor at IKERBASQUE (Basque Foundation for Science) and the University of the Basque Country. Prior to joining IKERBASQUE, he held numerous research positions in Europe, China, and Canada. His research interests include pattern recognition and computer vision.