LPQ and LDP Descriptors with ML Representation For Kinship verification

Abdelhakim Chergui[1], Salim Ouchtati[1], Hichem Telli[2], Fares Bougourzi[3], and Salah Eddine Bekhouche[4]

[1] Laboratory of LRES, University of Skikda, Algeria.
[2] Laboratory of LESIA, University of Biskra, Algeria.
[3] Laboratory of LTII, University of Bejaia, Algeria.
[4] Department of Electrical Engineering, University of Djelfa, Algeria.

Abstract. The automatic verification of kinship is a challenging problem that recently attracted much interest in computer vision, the kinship verification has become an active research field due to its potential applications such as organizing photo albums and images annotation, recognizing resemblances among humans and finding of missing children. In this paper, we propose an approach which takes two images as an input then give kinship result (kinship / non-kinship) as an output.This approach based on the Local Phase Quantization (LPQ) and Local directional pattern (LDP) features descriptors and the ML (Multi-Level) representation for the kinship verification from facial images, this work consists six stages which are : (i) face preprocessing, (ii) features extraction, (iii) face representation (iv) pair features representation and normalization, (v) features selection and (vi) kinship verification. Experiments are conducted on four public databases (Cornell KinFace, UB Kin database, KinFace-I, and KinFace-II). The obtained results are good compared with state-of-the-art approaches.

Keywords: Kinship verification, LPQ, LDP, ML.

1 Introduction:

Over the past two decades, a large number of face analysis problems have been investigated in the computer vision and pattern recognition community. Facial images convey many important human characteristics, such as identity, gender, expression, age, ethnicity and so on. Kinship verification from facial images is an interesting and challenging problem, Indeed, there are several types of kinship relationships: father-daughter relationship (F-D), mother-son (M-S), father-son (F-S) and mother-daughter (M-D). Nowadays, the recognition of these familial relationships has become an active area of research and it has much application such as organizing photo albums and images annotation, recognizing resemblances among humans and finding of missing.

There are many studies have been conducted on kinship verification from facial images which can be categorized based on the type of feature extraction and the similarity algorithms. Fang *et al.* [5] proposed a system for kinship verification based on PSM (Pictorial structure model) feature extraction and selection methods and they used KNN for the classification phase, they obtained a promising result on the Cornell KinFace database. Xia *et al.* [13] used another database named UB KinFace which contains the images of the child, young parent and old parent faces, using an extended transfer subspace learning method to mitigate the enormous divergence of distributions between children and old parents, and an intermediate distribution was used to close to bridge and reduce the divergence between the sources distributions.

Another interesting work was proposed by Shao *et al.* [10] where they used the version 2 of UB KinFace database to verify the kinship based on robust local Gabor filters to extract genetic-invariant features. In other words, a metric and transfer subspace learning were adopted to abridge the discrepancy between children and their old parents. Lu *et al.* [8] proposed a neighborhood repulsed metric learning (NRML) method for kinship verification. In addition, they proposed a multi view NRML (MNRML) method to seek a common metric distance in order to better use of the multiple descriptor features, they applied their method on The KinFaceW-I and KinFaceW-II datasets.

Yan *et al.* [15] proposed a discriminative multi metric learning method for kinship verification. First, they extracted multiple features using different face descriptors, then, they jointly learned multiple distance metrics with these multiple extracted features under which the probability of a pair of face images where the kinship relation having a smaller distance than the pair that has not a kinship relation. In this work, they applied their method on two databases: Cornell KinFace and UB Kin database. Yan *et al.* [16] proposed a new 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 relation from a wild dataset which 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 is the corresponding decision value from one SVM, they applied their method on both Cornell KinFace and UB Kin databases.

Wang *et al.* [12] proposed a deep kinship verification (DKV) model by integrating excellent deep learning architecture into metric learning. They employed a deep learning model which was followed by a metric learning formulation to select nonlinear features, which can find the appropriate project space to ensure that the margin between the negative sample pairs (i.e. parent and child without kinship relation) and the positive sample pairs is larger as possible, they applied their method on The KinFaceW-I and KinFaceW-II datasets. Zhou *et al.* [17] proposed an of ensemble similarity learning (ESL), first they introduced sparse bilinear similarity function to model the relative of the encoded properties in kin data. The similarity function parameterized by a diagonal matrix enjoys the superiority in computational efficiency, making it more practical for real-world high-dimensional kinship verification applications. Yan [14] proposed a neighborhood repulsed correlation metric learning (NRCML) method by using the correlation similarity measure where the kin relation of facial images can be better highlighted.

The rest of the paper is organized as follows: Our method is introduced in section 2. Then, the experimental results are presented to demonstrate the efficacy of our proposed methods in section 3. Finally, we conclude our work in section 4.

2 Proposed method

The kinship verification is the operation of using two persons faces to find if there is a familial relationship between them. Our proposed method consists of six stages which are : (i) face preprocessing, (ii) features extraction, (iii) face representation, (iv) pair features representation and normalization, (v) features selection and (vi) kinship verification. Fig. 1 illustrates the general structure of the proposed framework.

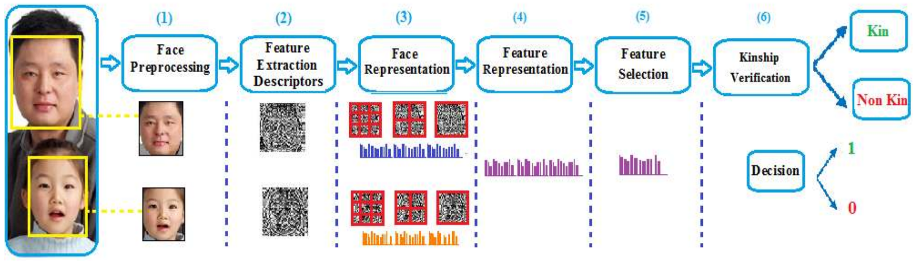


Fig. 1: General structure of the proposed Method.

2.1 Face preprocessing

In the face preprocessing, we applied the Haar cascade object detector that uses the Viola-Jones algorithm [11] in order to detect the face region, then we detected the face landmarks using Ensemble of Regression Trees (ERT) algorithm [7]. The locations of the two eyes are used to rectify the face 2D pose by applying a 2D similarity transform on the original face image [2]. Like in [3], we set the parameters $k_side = 0 : 5$, $k_top = 1$ and $k_bottom = 1 : 75$ to crop the face region of interest (ROI).

2.2 Features extraction

In this stage , we extracted the features by using two different texture descriptors (LDP and LPQ) , and for the face representation we used the ML for increased number of features

Local Directional Pattern (LDP) : is an eight-bit binary code assigned to each pixel of an input gray scale image. The pattern is calculated by comparing the relative edge response value of a pixel in different directions. The eight directional edge response values of a particular pixel are calculated using Kirsch masks in eight different orientations $(M_0 - M_7)$ centered on its own position [6]. These masks are shown in Fig.2.

$$\begin{bmatrix}-3 & -3 & 5\\-3 & 0 & 5\\-3 & -3 & 5\end{bmatrix} M_0(\uparrow) \quad \begin{bmatrix}-3 & 5 & 5\\-3 & 0 & 5\\-3 & -3 & -3\end{bmatrix} M_1(\nwarrow) \quad \begin{bmatrix}5 & 5 & 5\\-3 & 0 & -3\\-3 & -3 & -3\end{bmatrix} M_2(\leftarrow) \quad \begin{bmatrix}5 & 5 & -3\\5 & 0 & -3\\-3 & -3 & -3\end{bmatrix} M_3(\swarrow)$$

$$\begin{bmatrix}5 & -3 & -3\\5 & 0 & -3\\5 & -3 & -3\end{bmatrix} M_4(\downarrow) \quad \begin{bmatrix}-3 & -3 & -3\\5 & 0 & -3\\5 & 5 & -3\end{bmatrix} M_5(\searrow) \quad \begin{bmatrix}-3 & -3 & -3\\-3 & 0 & -3\\5 & 5 & 5\end{bmatrix} M_6(\rightarrow) \quad \begin{bmatrix}-3 & -3 & -3\\-3 & 0 & 5\\-3 & 5 & 5\end{bmatrix} M_7(\nearrow)$$

Fig. 2: Eight directions Kirsch edge masks

By applying eight masks, eight edge response values will be obtained m_0, m_1, ..., m_7, each one represents the edge significance in its respective direction. The response values are not equally important in all directions. In order to generate the LDP codewords, a k value must be given. Then, the top k values of $|m_j|$ are set to 1, and the rest $8-k$ values of $|m_j|$ are set to 0. LDP code for each pixel is calculated using the formulas below:

$$LDP_k = \sum_{i=0}^{7} b_i(m_i - m_k) \cdot 2^i \tag{1}$$

$$b_i(a) = \begin{cases} 1 & if\, a \geq 0 \\ 0 & otherwise. \end{cases} \tag{2}$$

where m_k is the $k-th$ most significant directional response. After computing the LDP code for each pixel (r, c), the histogram H of the image I is represented using this equation:

$$H(\tau) = \sum_{r=1}^{M} \sum_{c=1}^{N} f(LDP_k(r, c), \tau) \tag{3}$$

where τ is the ldp code value. The number of ldp histogram bins is calculated as follow:

$$N_{bins} = \frac{8!}{k! \cdot (8-k)!} \tag{4}$$

Fig. 3: Image conversion to LDP and LPQ

Local Phase Quantization (LPQ) : A texture descriptor called LPQ was proposed in [9]. It is based on the application of STFT. The advantage in STFT is that the phase of the low frequency coefficients is insensitive to centrally symmetric blur. The spatial blurring is represented by a convolution between the image intensity and a PSF. The LPQ descriptor uses the local phase information extracted by the 2-D DFT or, more precisely, a STFT computed over a rectangular $M-by-M$ neighborhood N_x at each pixel position x of the image $f(x)$ defined by this formula:

$$F(u,x) = \sum_{y \in N_x} f(x-y) e^{-j2\pi u^T y} = w_u^T f_x \tag{5}$$

where w_u is the basis vector of the 2-D DFT at frequency u, and f_x is another vector containing all M^2 image samples from N_x.

The local Fourier coefficients are computed at four frequency points $u_1 = [a,0]^T$, $u_2 = [0,a]^T$, $u_3 = [a,a]^T$, and $u_4 = [a,-a]^T$, where a is a scalar frequency below the first zero crossing of $H(u)$ that satisfies the condition $H(u_i) > 0$. So a vector obtained for each pixel, will be built like in this formula:

$$F_x = [F(u_1,x), F(u_2,x), F(u_3,x), F(u_4,x)] \tag{6}$$

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 by using a simple scalar quantization which presented in this formula:

$$q_j = \begin{cases} 1 & if \quad g_j \geq 0 \\ 0 & otherwise. \end{cases} \tag{7}$$

where g_j is the j_th component of the vector $G(x) = [Re\{F(x)\}, Im\{F(x)\}]$. The resulting eight binary coefficients q_j represent the binary code pattern. This code will be converted to decimal number between 0-255. From that, the LPQ histogram will have 256 bins [4].

2.3 Face representations ML (Multi Level) :

The most common face representation in computer vision is a regular grid of fixed size regions which called MB representation. MB face representation divides the image into

n^2 blocks where n is the intended level of MB. Fig. 4 shows the feature extraction procedure using LPQ descriptor with ML representation, level 4 [1].

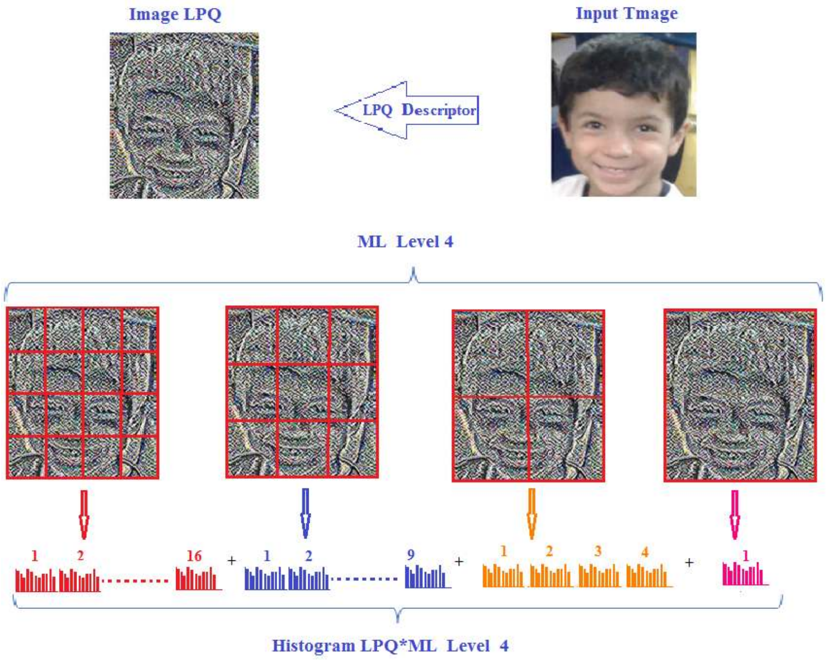


Fig. 4: Exemple: Multi-Level Local Phase Quantization level 4

Recently, a similar representation called ML representation used in the age estimation and gender classification topics. ML face representation is a spatial pyramid representation which constructed by sorted series of ML representations. The ML face representation level n is constructed from level 1, 2 , ... n ML face representations. Fig. 4 illustrates the ML face representations.

2.4 Pair features representation and normalization:

After extracting the features, we normalized the features of each pair (child / parent) using the formula given below:

$$F_{norm} = \frac{F}{\sqrt{\sum_{j=1}^{N} F(j)}} \tag{8}$$

Then this two feature vectors (child / parent) are presented as one feature vector using this formula:

$$F = |F_{child} - F_{parent}| \tag{9}$$

where F, F_{child}, F_{parent} are the new feature vector, the feature vector of the child and the feature vector of the parent respectively.

2.5 Features selection:

For the feature selection, we used a linear discriminant approach based on Fisher's score, which quantifies the discriminating power of features. This score is given by:

$$W_i = \frac{N_k(m_k - \bar{m})^2 + N_n(m_n - \bar{m})^2}{N_k.\sigma_k^2 + N_n.\sigma_n^2} \tag{10}$$

where W_i is the weight of feature i, $\bar{m}$ is the feature mean, N_X is the number of samples in the kinship class (k $\rightarrow$ kin / n $\rightarrow$ non-kin), 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 weight.

2.6 Kinship verification:

Support vector machines (SVM) constructs a hyperplane or set of hyperplanes in a high- dimensional space, which can be used for classification, regression. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. We used the binary SVM to train and test our proposed approach; the two binary classes are either there is a kinship relationship or not which are represented by 1 and 0 respectively.

3 Experiments

To evaluate the performance of the proposed method, we used four publicly available databases (Cornell KinFace, UB Kin database, KinFace-I, and KinFace-II).

3.1 Experimental Settings:

The Cornell KinFace database was created by Fang *et al.* [5]. It consists of 286 images and 143 positive pairs. The pairs are distributed as follows: 67 F-S (Father-Son), 32 F-D (Father-Daughter), 18 M-S (Mother-Son), and 26 M-D (Mother-Daughter).

The UB KinFace database was created by Shao *et al.* [10], and it has two versions (Ver1.0 and Ver2.0). The Ver2.0 contains 600 images and 400 positive pairs. Those pairs are a composition of 180 F-S, 159 F-D, 22 M-S, and 39 M-D.

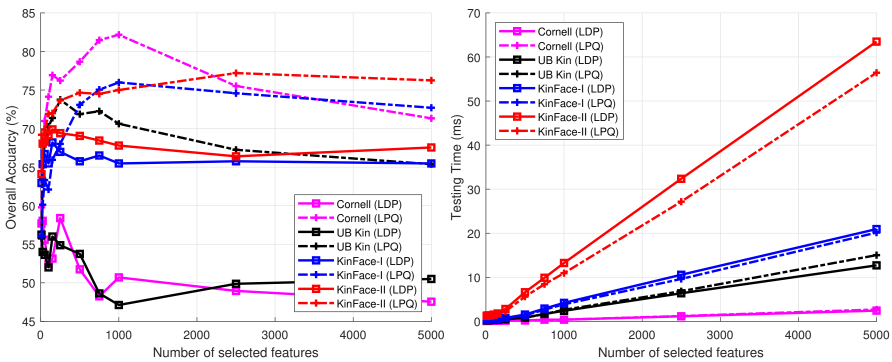
Lu *et al.* [8] provided the researchers with two databases called: KinFaceW-I and KinFaceW-II. The KinFaceW-I contains 1066 images and 533 positive pairs with the following distribution : 156 F-S, 134 F-D, 116 M-S, and 127 M-D. On the other hand, KinFaceW-I has 2000 images and 1000 positive pairs. It has a balanced pairs distribution with 250 pairs for each kin relationship.

The negative pairs which used in these experiments are selected randomly taking into consideration the distribution of the relationships. Also, the 5 folds are selected randomly take into account the distribution of the relationships.

3.2 Experimental Results:

The experimental results on the used databases are summarized on Fig. 5a, which shows that the accuracy proportionally related with the increase of the number of the selected until an optimal value, then the accuracy decreases and finally stabilizes. The explanation of these transitions is: in the first hand the selected features are very few and each time we add more selected features the accuracy becomes better until the optimal features number. The phenomena of the decreasing accuracy after the optimal features number is because of the adding less relevant features decreases the accuracy. From other hand, the use of ML-LPQ outperforms the use of ML-LDP with the varying of the number of selected features for all of the used databases. The difference between the two descriptors is very huge for both UB and Cornell databases, which is about 20 % and 33 % for UB, and Cornell databases ,respectively.

The Figure (5b) shows the different training speed for each database using the two feature extraction methods (ML-LPQ and ML-LDP). We can notice two notes from the results; first, the training time increases with the number of the selected features. Secondly, the databases that contains more samples takes more time in the training phase.



(a) Accuracy as a function of different selected features number.

(b) CPU time (in seconds) of the training phase as a function of different features ratios.

Fig. 5: Accuracy and CPU time results

Table 1: A comparison of the proposed approach with other kinship verification approaches

Year	Approach	Databases			
		Cornell KinFace	**UB KinFace**	**KinFace W-I**	**KinFace W-II**
2010	**PSM**[5]	70.67 %		-	-
2011	**TL**[13]	-	60.00 %	-	-
	TSL[10]	-	69.67 %	-	-
2014	**PDFL** [16]	71.90 %	67.30 %	-	-
	DML [15]	73.50 %	**74.50 %**	-	-
	MNRML [8]	-	-	69.90 %	76.5 %
2015	**DKV** [12]	-	-	66.90 %	69.50 %
2016	**ESL** [17]	-	-	74.10 %	74.30 %
2017	**NRCML** [14]	-	-	65.80 %	65.80 %
2018	**Proposed LPQ_ML**	**82.86 %**	73.25 %	**75.98 %**	**77.20 %**

The comparisons of our proposed approach with the state of art methods are summarized on Table (1). From that Table we observe that our approach (ML-LPQ) performs better than the state of the art methods for the Cornell, Kinface W-I, and Kinface W-II databases. However, for the UB database, our proposed approach has the second best accuracy, and the difference with the best method is very small.

The Fig. (6) is an example of testing our method to verify the kinship between the persons on the picture.

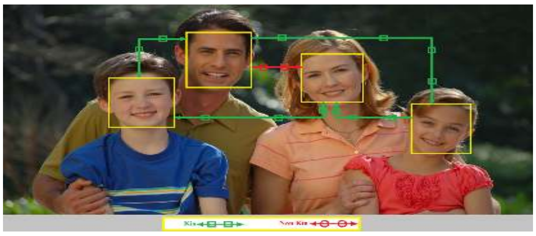


Fig. 6: Example of kinship verification application

4 Conclusion:

In this paper, we described a novel approach for kinship verification based on the descriptors LDP and LPQ with ML representation. The experimental results showed that our approach provides a better performance than previous approaches. As a future work, we propose to use of other descriptors with PML representation. Also, we envision the use of other pair feature representations as well performing different scenarios of experiments such as cross-database experiments.

References

1. S. E. Bekhouche, A. Ouafi, A. Benlamoudi, A. Taleb-Ahmed, and A. Hadid. Facial age estimation and gender classification using multi level local phase quantization. In *2015 3rd International Conference on Control, Engineering Information Technology (CEIT)*, pages 1–4, May 2015.
2. SE. Bekhouche. *Facial Soft Biometrics: Extracting demographic traits*. PhD thesis, Faculté des sciences et technologies, 2017.
3. SE. Bekhouche, A. Ouafi, F. Dornaika, A. Taleb-Ahmed, and A. Hadid. Pyramid multi-level features for facial demographic estimation. *Expert Systems with Applications*, 80:297–310, 2017.
4. F. Bougourzi, SE. Bekhouche, ME Zighem, A. Benlamoudi, A. Ouafi, and Taleb-Ahmed A. A comparative study on textures descriptors in facial gender classification. In *10 me Confrence sur le Gnie Electrique*, Apr 2017.
5. Ruogu Fang, Kevin D Tang, Noah Snavely, and Tsuhan Chen. Towards computational models of kinship verification. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 1577–1580. IEEE, 2010.
6. Taskeed Jabid, Md Hasanul Kabir, and Oksam Chae. Local directional pattern (ldp) for face recognition. In *Consumer Electronics (ICCE), 2010 Digest of Technical Papers International Conference on*, pages 329–330. IEEE, 2010.
7. Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.
8. Jiwen Lu, Xiuzhuang Zhou, Yap-Pen Tan, Yuanyuan Shang, and Jie Zhou. Neighborhood repulsed metric learning for kinship verification. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):331–345, 2014.
9. Ville Ojansivu and Janne Heikkilä. Blur insensitive texture classification using local phase quantization. In *International conference on image and signal processing*, pages 236–243. Springer, 2008.
10. Ming Shao, Siyu Xia, and Yun Fu. Genealogical face recognition based on ub kinface database. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 60–65. IEEE, 2011.
11. Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.
12. Mengyin Wang, Zechao Li, Xiangbo Shu, Jinhui Tang, et al. Deep kinship verification. In *Multimedia Signal Processing (MMSP), 2015 IEEE 17th International Workshop on*, pages 1–6. IEEE, 2015.
13. Siyu Xia, Ming Shao, and Yun Fu. Kinship verification through transfer learning. In *IJCAI*, pages 2539–2544, 2011.
14. Haibin Yan. Kinship verification using neighborhood repulsed correlation metric learning. *Image and Vision Computing*, 60:91–97, 2017.
15. Haibin Yan, Jiwen Lu, Weihong Deng, and Xiuzhuang Zhou. Discriminative multimetric learning for kinship verification. *IEEE Transactions on Information forensics and security*, 9(7):1169–1178, 2014.
16. Haibin Yan, Jiwen Lu, and Xiuzhuang Zhou. Prototype-based discriminative feature learning for kinship verification. *IEEE Transactions on cybernetics*, 45(11):2535–2545, 2015.
17. Xiuzhuang Zhou, Yuanyuan Shang, Haibin Yan, and Guodong Guo. Ensemble similarity learning for kinship verification from facial images in the wild. *Information Fusion*, 32:40–48, 2016.