الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
المدرسة الوطنية العليا للتكنولوجيات المتقدمة
National Higher School of Advanced Technologies

Génie Électrique et Infomatique Industrielle

Department of Electrical Engineering and Industrial Computing

Final Year Project to Obtain the Diploma of

Master

-Field -

Electronics

-Specialty-

Embedded systems

- Subject -

State of the art on facial recognition methods for determination of kinship

Realized by

AMEUR Lina

Members of The Jury:

BOUTARFA Souhila	MCB ENSTA	President
CHERABIT Noureddine	MAA ENSTA	Examiner
MOKDAD Fatiha	MRB CDTA	Supervisor
CHOUAF Seloua	MAA ENSTA	Co-Supervisor

Algiers, the 25/06/2024

Academic year 2023–2024

State of the art on facial recognition methods for determination of kinship

1st AMEUR Lina
Department of Electrical Engineering And Industrial Computing
National Higher School of Advanced Technologies (ENSTA)
Algeiers, Algeria
l_ameur@ensta.edu.dz

2nd MOKDAD Fatiha
Center for Development of Advanced Technologies (CDTA)
Algiers, Algeria
fmokdad@cdta.dz

3rd CHOUAF Seloua
Department of Electrical Engineering And Industrial Computing
National Higher School of Advanced Technologies (ENSTA)
Algiers, Algeria
seloua.chouaf@ensta.edu.dz

***Abstract*—Facial recognition technology has significantly evolved over the past decade, finding applications in security, social media, and various other fields. One of the emerging and intriguing applications of this technology is kinship determination, which involves determining familial relationships based on facial features. In this paper, we established the state-of-the-art understanding of these methods is essential for advancing this field and leveraging its potential for practical applications.**

***Index Terms*—LBP, feature selection, kinship verification, face analysis, feature extraction, deep learning, metric learning**

I. Introduction

Facial kinship determination models aim to determine whether two individuals belong to the same family based on faces patterns. The automatic kinship systems take two facial images as input and decide if there is a familial relationship, such as between a parent and child or siblings. While DNA tests are the most accurate for verifying kinship, they are impractical for many scenarios like video surveillance.

Early kinship determination methods used features like Local Binary Patterns (LBP) [2], Local Phase Quantization (LPQ) [1], and Histograms of Gradients (HOG) [5] with Support Vector Machines (SVMs) for determination [8]. These methods, however, struggled with image variations and generalizing to unseen data. Recent advancements in machine learning, particularly deep learning, have shown superior performance with learned features. Researchers have explored various multi-view and multi-feature approaches for better accuracy. The development of these sophisticated techniques underscores the potential of deep learning in enhancing kinship determination accuracy across diverse applications such as forensics, locating missing children, and image annotation.

Additionally, two kinship determination competitions were held in 2014 and 2015. The key goal of these competitions are to compare the performance of different methods on a new-collected dataset with the same evaluation protocol and develop the first standardized benchmark for kinship verification in the wild.

II. Main challenges in facial kinship verification

Face Kinship verification (FKV) is a binary classification problem complicated by the fact that kinship pairs do not share the same facial identity, showing only subtle genetic similarities. Psychological research [6] indicates that while facial similarity and kinship judgments are correlated, they are not identical, making FKV even more challenging. Key challenges include large interclass variations, such as changes in age, pose, expression, and imaging conditions, which make it difficult to extract discriminative features. Additionally, significant interpersonal variations, like age gaps and gender differences, further complicate FKV. Small interclass variations also pose a problem, as positive kinship pairs may show minimal similarities while negative pairs might appear quite similar, making it hard to define clear decision boundaries. Moreover, the lack of large-scale kinship datasets hampers the develop-

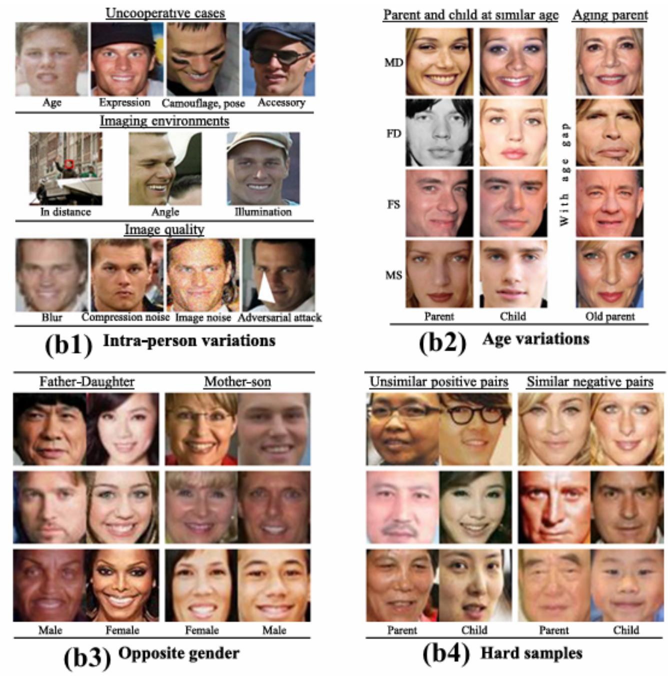

Fig. 1. Summarizing the main challenges [22]

ment of effective FKV algorithms, particularly deep learning-based methods, due to privacy and security concerns. Therefore, it is essential to create extensive kinship datasets that accurately represent family data worldwide to address these challenges.[22]
Figure 1 above summarizes the main challenges of kinship determination, as well Figure 1 (b1, b2 and b3) show interclass variations, in which Figure 1 (b1) contains the pos sible variations within one subject, with each image line demonstrating influences from different factors. Then, Figure 1 (b2 and b3) illustrate the facial similarity gap between kinship caused by age and gender differences, as well as variations among kin pairs and families. Figure 1 (b4) demonstrates less discrimination of FKV that hard kin and non-kin samples exist when kin pairs have less similarity on appearance, while non-kin pairs inversely show significant similarities

III. Facial kinship verification approaches

acial kinship verification approaches can be grouped into three main types based on their key contributions: feature extraction approaches, metric learning approaches and deep learning techniques.

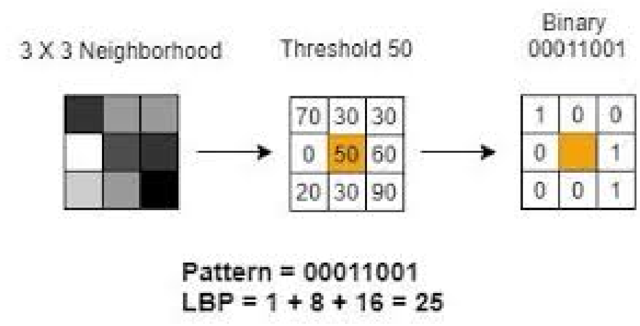

Fig. 2. Local binary pattern method

A. Feature extraction methods

Facial description is a crucial step in any facial analysis system. In this step, images get analyzed using a set of signal processing techniques and geometrical methods to look for properties useful for representing and summarizing them into a one-dimensional feature-vector.
In this section, we will discuss several image descriptors commonly employed in research. Specifically, we'll focus on descriptors used to represent facial images in current approaches to kinship verification. Many of these descriptors were initially developed for texture classification and have proven effective in accurately representing facial images too.

*1) **Local Binary Patterns**:* LBP is a texture descriptor introduced in 1994 in [17]. LBP has proven its high performance on several datasets, especially when combined with the HOG descriptor.
The $LBP_{PR}(x_c, y_c)$ code of each pixel (x_c, y_c) is computed by comparing the gray value g_c of the central pixel with the gray values g_i for $i = 0$ of its P neighbors, as follows:

$$LBP_{PR} = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p \quad (1)$$

where g_c is the gray value of the central pixel, g_p is the gray value of the p-th neighbor, and $s(g_p - g_c)$ is defined as:

$$s(g_p - g_c) = \begin{cases} 1 & \text{if } (g_p - g_c) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Figure 2 explains a simple case of LBP implementation method.

Many variations of the classic Local Binary Patterns (LBP) method exist, and these will be introduced in the following sections.

- **Uniform Local Binary Patterns**

The idea behind uniform LBP is that the generated LBP codes can be split into two groups: uniform and non-uniform codes.

An LBP code is considered uniform if it has at most two transitions between 0 and 1 in its binary form. For example, the LBP codes $(0)_{10} = (00000000)_2$ and $(255)_{10} = (11111111)_2$ are uniform because they have no transitions between 0 and 1.
Codes like 224_{10}=$(11100000)_2$ and $(199)_{10} = (11000111)_2$ are also uniform because they have one and two transitions, respectively. In contrast, codes like $(213)_{10} = (11010101)_2$ and $(117)_{10} = (01110101)_2$ are non-uniform because they contain more than two transitions in their binary sequence. [1].

To extract the uniform LBP feature vector, we can either build a histogram using only the uniform LBP codes or group all non-uniform codes into a single histogram bin. There are three main reasons for using uniform LBP:

Uniform codes represent more **a)** natural and smooth features, while several transitions indicate noise or random texture. Uniform LBP reduces the size of the feature vector, **b)** simplifying the data. Research by Ojala et al. [16] showed that uniform patterns occur **c)** more frequently in textures than non-uniform patterns.

- **Color Local binary pattern (CLBP)**

It is an extension of LBP texture descriptor, which incorporates color information in addition to the grayscale information used by the original LBP.
The basic idea behind CLBP is to encode the local color and texture information in an image by comparing the color values of a central pixel with its neighboring pixels. This is done for each color channel (e.g., red, green, and blue) separately, resulting in three different LBP codes that are then combined to form the final CLBP code.

*2) **Local phase quantization:*** LPQ is a method for constructing local image descriptors that are robust to blur and other types of degradations. It is based on quantizing the Fourier transform phase in local neighborhoods, which is a blur-invariant property under certain conditions. Following this equation :

$$F(u,x) = \sum_{x \in N_x} f(x-y)e^{-i2\pi u^\top y} = w_u^\top f_x \quad (3)$$

First, the method extracts small patches from the original image. These patches are then transformed into their frequency domain using Fourier transforms as presented in equation 3, which helps capture the variation in pixel values across different frequencies. The next step involves simplifying the phase information of these frequency components by assigning them a specific number of bits. This simplified phase data is then used to create a unique description for each pixel within the patch. Lastly, all these individual pixel descriptions are combined to create a single overall description that characterizes the entire image.

*3) **Scale Invariant Feature Transform (SIFT):*** The Scale Invariant Feature Transform (SIFT) descriptor, introduced by David G. Lowe in 2004 [14], is a powerful tool designed to extract features from images that are robust against various transformations. As the name suggests, SIFT features remain consistent even when the image is scaled up or down. Additionally, they are resilient to changes in translation, rotation, and partially invariant to illumination variations and affine or 3D projections. The process of computing SIFT descriptors begins with identifying key points in the image that are stable and robust against geometric transformations and small signal distortions. This is achieved by constructing a scale-space pyramid, where the image is repeatedly convolved with a Gaussian kernel and sampled at different scales. The key points are then identified as the local maxima or minima of this scale-space representation. Once these stable key points are localized, the next step is to represent the local image regions around each key point in a way that is invariant to location, scale, and rotation. To accomplish this, Lowe proposed creating multiple orientation planes for each key point, where each plane contains gradients corresponding to a specific orientation. These orientation planes capture the local image characteristics in a way that is robust to the aforementioned transformations, making SIFT descriptors highly effective for various computer

vision tasks.
(See figure3)

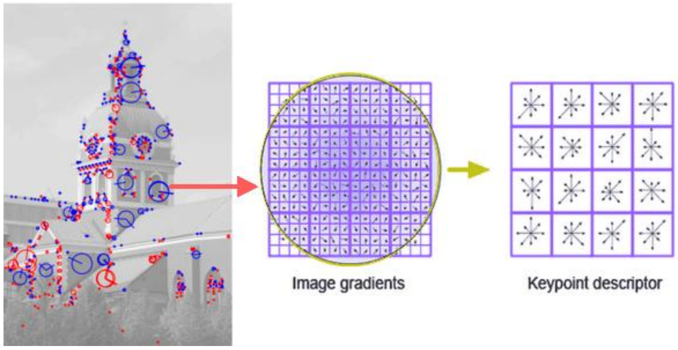

Fig. 3. Example of SIFT descriptor

*4) **Learning-based descriptor (LE):*** The learning-based descriptor (LE), introduced by Cao et al. [4], is a feature extraction technique that involves sampling neighboring pixels around each pixel in an image. The sampling process selects a subset of pixels that belong to different concentric rings, or circles centered on the current pixel. The authors recommended sampling 8R pixels from each ring with radius R, for instance, 16 pixels would be sampled from a ring with a radius of 2. Several types of sampling patterns were experimented with, as illustrated in their paper [4]. After sampling, the low-level feature vector obtained from the neighboring pixels is normalized to a unit-length vector, enhancing its robustness to illumination changes. The next step involves clustering the pixels into M groups using K-Means clustering on the feature vectors extracted during the sampling step. Finally, the image is represented by a 1xM histogram, where each bin corresponds to one of the M clusters obtained from the clustering process. This learning-based descriptor aims to capture local image characteristics in a compact and robust manner, making it suitable for various computer vision tasks,

*5) **Binarized Statistical Image Features (BSIF):*** It is a method for constructing local image descriptors that efficiently encode texture information and are suitable for histogram-based applications. The approach is based on statistics of natural images, which improves its modeling capacity compared to other methods like Local Binary Patterns (LBP) and Local Phase Quantization (LPQ).
The process of BSIF involves several key steps. Initially, local image patches are extracted from the input image, allowing a focused analysis of specific areas within the image. Subsequently, the responses of linear filters are computed for each of these patches, capturing important information about the image's texture. These filter responses are then binarized by applying a threshold at zero, converting the responses into binary form. The resulting binarized responses are combined to generate a binary code string for each pixel. Finally, these binary code strings are aggregated to form a histogram, which encapsulates the texture properties within a subregion of the image, providing a comprehensive representation of the image's texture characteristics. [9]

*6) **Histogram of oriented gradiant:*** The Histogram of Oriented Gradients (HOG) is a sophisticated method used in computer vision to detect objects in images. It works by counting how often gradients (changes in pixel intensity) point in specific directions within different parts of an image. This approach creates a detailed and reliable representation of both the appearance and shape of objects in the image. [3]

Here's how HOG works: first, it calculates gradients across the entire image using methods like the Sobel operator or Canny edge detector. Then, it divides the image into smaller sections called cells and computes a histogram for each cell, showing the distribution of gradient directions. These histograms capture the intensity changes or edges within each part of the image.

To make the descriptors more accurate, HOG normalizes these local histograms across larger areas called blocks. This normalization helps HOG ignore differences in brightness and shadows, making it robust even when lighting conditions change.

Overall, HOG is valued for its ability to handle variations in lighting and its resistance to common geometric transformations like rotation and scaling (though it's sensitive to changes in object orientation). It's also efficient to compute, using a grid of evenly spaced cells with overlapping normalization.

HOG finds applications in a wide range of tasks such as detecting objects in images and videos, classifying images based on their content, and retrieving images that are similar in appearance. Its versatility and reliability make it a cornerstone in the field of computer vision.

B. Metric learning methods

Since facial similarity between parents and children is generally higher than that between unrelated individuals, it is crucial to develop effective approaches for verifying parenting from facial images. Metric learning is a method that aims to learn an automatic similarity measure from data rather than using pre-defined distances. In the context of facial kinship verification, the goal is to learn a metric where the distance between a pair of faces with a parent-child relationship is smaller than the distance between a pair without a relationship.

The distance between any parent-child facial image pair [21]$(\mathbf{x}_i, \mathbf{y}_j)$ can be defined as:

$$d_M(\mathbf{x}_i, \mathbf{y}_j) = \sqrt{(\mathbf{x}_i - \mathbf{y}_j)^T M (\mathbf{x}_i - \mathbf{y}_j)} \quad (1)$$

where M is symmetric and positive semi-definite, and it can be decomposed into $M = W^T W$. Therefore, the distance metric can be reformulated as:

$$d_M(\mathbf{x}_i, \mathbf{y}_j) = \sqrt{(\mathbf{x}_i - \mathbf{y}_j)^T W^T W (\mathbf{x}_i - \mathbf{y}_j)} \quad (2)$$

$$= \|W\mathbf{x}_i - W\mathbf{y}_j\| \quad (3)$$

Among the early approaches to tackle kinship verification, Somanath and Kambhamettu [18] applied ensemble learning techniques. Later, Zhou et al. [28] learned a set of weak bilinear similarity functions from kinship databases. They did this by minimizing the kinship constraints between pairs of images (forcing images of related people to be similar) and maximizing the diversity of the learned similarities.

Zhao et al. [26] proposed a novel metric learning method called the Multiple Kernel Similarity Metric (MKSM). In this method, instead of using a single Mahalanobis distance metric, the similarity computation is essentially based on an implicit non-linear feature mapping. The overall MKSM is a weighted combination of base similarities and thus has the ability to fuse features.

C. Deep learning methods

Traditional hand-crafted feature extraction methods are limited in their ability to describe features effectively. In contrast, CNN-based deep learning methods excel at capturing complex, non-linear patterns. These methods can learn efficient feature embeddings from raw data by applying task-specific constraints, bypassing the need for traditional hand-crafted feature extraction rules [22]. With the rapid advancement of deep learning in computer vision and the availability of large-scale kinship datasets, researchers began exploring deep learning techniques for kinship analysis around 2016. Various innovative deep learning architectures have been employed, including basic neural networks [20], deep metric learning , auto-encoder-based architectures [10], and attention networks [24].

The first method, proposed by Wang et al. [20], consists of two stages: feature extraction and deep metric learning. Traditional methods are used to extract facial features, which are then processed through non-linear auto-encoders and a Mahalanobis distance metric to project the features into a non-linear space. However, this approach has a drawback: it uses LBP features, which lack the detailed information of the original image. The first end-to-end deep learning method for kinship verification was introduced by Zhang et al. [25]. Their network takes two stacked facial images as input and outputs the verification result.

*1) **Deep Metric Learning Methods**:* To optimize the distance between two input facial images, Zhou and al [27] incorporated a distance metric into network training, leading to the development of Deep Metric Learning methods. A typical network architecture used is the Siamese Network. Unlike single-stream networks, Siamese networks have two parallel streams with shared weights and use the distance metric as a loss function. This setup helps the network learn an optimal feature space where positive pairs (related individuals) have small distances and negative pairs (unrelated individuals) have large distances.

Li and al [11]. proposed the Similarity Metric-based Convolutional Neural Networks (SMCNN) method. This network takes two facial images, X and Y, as inputs, with $G(\cdot)$ representing the output of the fully connected (FC) layer. The l_1-norm is used to compute the distance between the two output embeddings, as shown in Equation 4:

$$D(X, Y) = \|G(X) - G(Y)\|_1 \quad (4)$$

During training, Li et al. introduced a threshold (τ) to further distinguish positive and negative samples, with labels $y = 1$ for positive samples and $y = -1$ for negative samples. The cost function of the network is defined as:

$$L_{SMCNN} = f(1 - y(\tau - D(X, Y))) \quad (5)$$

where $f(\cdot)$ is the generalized logistic loss function. Gradient descent is used to optimize the convolutional neural networks.

Commonly used metric-based loss functions include Contrastive Loss and Triplet Loss. These functions are based on distance measurements like Euclidean distance. Contrastive loss takes positive and negative pairs as inputs, while triplet loss uses three inputs: the anchor (a), the positive (p), and the negative (n). This setup clusters positive sample pairs and separates positive and negative samples.

The efficiency and performance of deep metric learning techniques depend significantly on the selection of sample pairs/tuples. Hard Sample Mining methods have been proposed to address this, focusing on finding positive sample pairs with large distances and negative sample pairs with small distances within training batches. This approach generates large backward losses and effectively trains the network [12, 24].

*2) **Architectures Based on Auto-Encoders:*** Another approach to deep kinship verification involves auto-encoders (AE) [10]. The first application of auto-encoders in kinship verification aimed to train a model for facial feature extraction [20], producing a reduced feature representation of the input.

Auto-encoder methods are motivated by the correlation between inputs and outputs and can be categorized into traditional auto-encoders [20] and neural network (NN)-based auto-encoders. Traditional auto-encoders learn the relation mapping representation by minimizing a loss function designed to fit two input images, while NN-based auto-encoders use multiple layers of projection and optimize the network through back-propagation.

Liang et al. [13] utilized intermediate layers to describe the relationship between input features and their encodings.

IV. Related works

The first study on kinship verification was reported in 2010 by Fang et al.[8]. The authors compared the performance of an automatic kinship verification method against human performance. However, they used the Cornell KinFace database which has limited data (only 150 parent/child pairs). Their initial attempt was to verify the resemblance between parent-child pairs. They based this on extracting 22 facial feature descriptors for classification such as skin color, facial structure identification of eyes, mouth, distance-based features, as well as statistical features like histogram of oriented gradients (HOG). Then, to classify the face pairs as kin or non-kin, either the k-nearest neighbor (KNN) classifier with Euclidean metric or a support vector machine (SVM) with a radial basis function kernel was used. Although this technique gave good results, it did not extend well to general kinship verification due to physical and genetic variations such as age difference between father and son or gender (brother/sister) for example. Genetic statistics-based studies critically observed that parents younger facial images resemble their children more than images captured when they are older. This led to the creation of the UB KinFace database, comprising face images of children, young parents, and old parents. Using this database, Xia et al.[23] proposed the transfer learning (TSL) method in hopes of reducing the large distribution divergence between children and old parents, by leveraging an intermediate distribution close to both distributions as well as using Gabor wavelets for feature extraction. This approach improved the overall kinship verification accuracy and made the task more discriminative. After that, two databases KinFaceW-I and KinFaceW-II were assembled by Lu et al. [15] to guide further research. The availability of these databases motivated more researchers to contribute to this topic. They also proposed the Neighborhood Repulsed Metric Learning (NRML) method, where metric learning aims to learn a good distance metric to minimize distances between positive kin image pairs while pushing non-kin image pairs farther apart. This method was tested using different local feature descriptors like Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and Scale-Invariant Feature Transform (SIFT). Fang et al. proposed a new approach for kinship verification on their Family101 dataset [8], by modeling the problem as reconstructing a face from mixed parts across a set of families, inspired by the biological

inheritance process. Their approach segments the face into parts (eyes, nose, mouth..) instead of taking the whole face, and reconstructs each part as a linear combination of parts from the dataset. To evaluate this approach, they used a dense SIFT descriptor on resized 61x49 pixel facial images [7].
The table I above shows facial verification approaches with different methods.

TABLE I
REVIEW OF FACIAL KINSHIP VERIFICATION APPROACHES [19]

Catégories	Method	year	descriptors	Classificator	Dataset	Accuracy %
Features	Computational model [17]	2010	Face appearance and geometry	KNN	Cornell KinFace	70,7
	Apparence [15]	2014	Face appearance and geometry	SVM	Family101	92
	Multi- Features[19]	2015	TPLBP+LPQ+WLD	SVM	KinFace W-I	86,3
					KinFaceW-II	83,1
Metric learning	(metric learning)	2012	SIFT+ PHOG	Kernelsim	VADANA	80,2
	DMML [20]	2014	LBP+SPLE+ SIFT	SVM	KinFaceW-I	72.25
					KinFaceW-II	78.25
					Cornell	73.75
	MKSM [21]	2018	LPQ	Mahalanobis distance	KinFaceW-I	81.46
					KinFaceW-II	82.45
					TSKinFace	81.89
Deep Learning	CNN [22]	2015	CNN	CNN	KinFaceW-I	77,5
					KinFaceW-II	88,4
	Deep+Shallow [23]	2016	Deep+Spatio- temporal	SVM	UvA- NEMO	91

V. Conclusion

The field of kinship verification is rapidly evolving, with researchers continually pushing the boundaries of accuracy and exploring innovative applications. As these challenges are met, kinship verification holds the potential to become an even more valuable tool across various sectors.

However, as the technology advances, ensuring the security and privacy of the data involved becomes increasingly critical. Robust security measures must be implemented to protect sensitive information from unauthorized access and misuse.

Looking to the future, kinship verification systems could see broader utilization in several areas. In law enforcement, they could enhance the identification and reunification of missing persons and trafficked individuals. In genealogy, these systems could provide more accurate family tree constructions and help individuals connect with their biological relatives. Social services could use kinship verification to streamline processes in adoption and foster care, ensuring that children are placed with their biological relatives when appropriate.

Further advancements in machine learning and biometric technologies will likely enhance the reliability and applicability of kinship verification systems. Continued research and development, coupled with a focus on ethical and secure practices, will be essential to fully realizing the benefits of kinship verification while safeguarding individual privacy rights.

References

[1] S. Abdellah. *Image-based Facial Kinship Verification*. PhD thesis, University of Science and Technology HOUARI BOUMEDIENE, 2022.

[2] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I 8*, pages 469–481. Springer, 2004.

[3] Built In. Histogram of oriented gradients. https://builtin.com/articles/histogram-of-oriented-gradients. Accessed: 2024-06-19.

[4] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2707–2714. IEEE, 2010.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[6] L. M. DeBruine, F. G. Smith, B. C. Jones, S. C. Roberts, M. Petrie, and T. D. Spector. Kin recognition signals in adult faces. *Vision Research*, 49(1):38–43, 2009.

[7] R. Fang, A. Gallagher, T. Chen, and A. Loui. Kinship classification by modeling facial feature heredity. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 2983–2987, 2013.

[8] R. Fang, K. D. Tang, N. Snavely, and T. Chen. Towards computational models of kinship verification. In *2010 IEEE International Conference on Image Processing*, pages 1577–1580, 2010.

[9] J. Kannala and E. Rahtu. Bsif: Binarized statistical image features. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, pages 1363–1366. IEEE, 2012.

[10] N. Kohli, D. Yadav, M. Vatsa, R. Singh, and A. Noore. Supervised mixed norm autoencoder for kinship verification in unconstrained videos. *IEEE Transactions on Image Processing*, 28(3):1329–1341, 2019.

[11] L. Li, X. Feng, X. Wu, Z. Xia, and A. Hadid. Kinship verification from faces via similarity metric based convolutional neural network. In *International Conference on Image Analysis and Recognition*, pages 539–548. Springer, 2016.

[12] W. Li, S. Wang, J. Lu, J. Feng, and J. Zhou. Meta-mining discriminative samples for kinship verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021.

[13] J. Liang, J. Guo, S. Lao, and J. Li. Using deep relational features to verify kinship. In *CCF Chinese Conference on Computer Vision*, pages 563–573. Springer, 2017.

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Jour-*

nal of Computer Vision, 60(2):91–110, 2004.

[15] J. Lu, J. Hu, X. Zhou, J. Zhou, M. Castrillón-Santana, J. Lorenzo-Navarro, T. F. Vieira, et al. Kinship verification in the wild: The first kinship verification competition. In *2014 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–6. IEEE, September 2014.

[16] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[17] T. Ojala, M. Pietikäinen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *12th IAPR International Conference on Pattern Recognition, Conference A: Computer Vision & Image Processing, ICPR 1994*, volume 1, pages 582–585, Jerusalem, October 1994. IEEE.

[18] A. W. Senior and R. M. Bolle. Face recognition and its application. In *Biometric Solutions*, pages 83–97. Springer, Boston, MA, 2002.

[19] University of Biskra. Verification faciale de la parenté par un réseau de neurone. Retrieved from http://archives.univ-biskra.dz/bitstream/123456789/11109/1.

[20] M. Wang, Z. Li, X. Shu, and J. Tang. Deep kinship verification. In *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6, 2015.

[21] Z. Wei, M. Xu, L. Geng, H. Liu, and H. Yin. Adversarial similarity metric learning for kinship verification. *IEEE Access*, pages 100029–100035, 2019.

[22] X. Wu, X. Feng, X. Cao, et al. Facial kinship verification: A comprehensive review and outlook. *International Journal of Computer Vision*, pages 1494–1525, 2022.

[23] S. Xia, M. Shao, J. Luo, and Y. Fu. Understanding kin relationships in a photo. *IEEE Transactions on Multimedia*, 14:1046–1056, 2012.

[24] H. Yan and S. Wang. Learning part-aware attention networks for kinship verification. *Pattern Recognition Letters*, pages 169–175, 2019.

[25] K. Zhang, Y. Huang, C. Song, H. Wu, and L. Wang. Kinship verification with deep convolutional neural networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[26] Y. G. Zhao, Z. Song, F. Zheng, and L. Shao. Learning a multiple kernel similarity metric for kinship verification. *Information Sciences*, 430:247–260, 2018.

[27] X. Zhou, K. Jin, M. Xu, and G. Guo. Learning deep compact similarity metric for kinship verification from face images. *Information Fusion*, 48:84–94, 2019.

[28] X. Zhou, Y. Shang, H. Yan, and G. Guo. Ensemble similarity learning for kinship verification from facial images in the wild. *Information Fusion*, 32:40–48, 2016.