



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



Department of Electrical Engineering and Industrial Computing

Final Year Project to Obtain the Diploma of

Engineering

-Field -

Electronics

-Specialty-

Embedded systems

- Subject -

Automatic verification of kinship based on facial images

Realized by

AMEUR Lina

Members of The Jury:

BOUTARFA Souhila
CHERABIT Nouredine
MOKDAD Fatiha
CHOUAF Seloua

MCB ENSTA
MAA ENSTA
MRB CDTA
MAA ENSTA

President
Examiner
Supervisor
Co-Supervisor

Algiers, the 25/06/2024

Academic year 2023–2024

Dedication

الحمد لله حياً و شكراً و امتناناً، الذي بفضلہ أعيش اليوم واقعاً أفخر
به.

إلى من زرع فيّ الطموح الذي يدفعني دائماً للأمام، إلى من دعمني بلا
حدود وأعطاني دون مقابل، "أبي".

إلى قوتني ومصدر قوتي، إلى المرأة التي برويتها أنسى كل همومي،
"أمي".

إلى من قيل فيهم "سنشد عضدك بأخيك"،
إلى من مدّ يده دون كلل أو ملل وقت ضعفي، "أيمن"، أدامك الله سنداً
لي.

إلى من كانت مثلاً يُحتذى وفخراً أعتز به، "إيمان".

إلى من أسلم لهم مشعل النجاح، "هيبه" و "بشرى"، أدامكن الله
وحفظكن من كل سوء.

إلى الأصدقاء الذين هم كالإخوة، من آمنوا بي وساندوني في وقت
الشدة، "أميرة" و "منال".

Acknowledgment

I would like to extend my deepest gratitude to the esteemed members of the jury who graciously accepted to be part of this endeavor. Thank you to Mrs. Boutarfa, Mr. Cherabit for their precious contribution.

In particular, I would like to express my gratitude to my supervisor, **Mme. MOKDAD Fatiha**, whose expertise and precious advice were invaluable. Her confidence in my abilities encouraged us to push my limits and achieve remarkable results.

I would like also to extend my warmest thanks to my Co-supervisor, **Mme. CHOUAF Seloua**, for her invaluable advice and constant support. Her presence by my side was a source of inspiration and motivation throughout the process.

Another thank the entire CDTA team, who have to the success of this project. Their collaboration, support and enthusiasm greatly facilitated my work and strengthened my resolve. Last but not least, thank all the teachers of the Department of Electrical Engineering And Industrial Computing for the knowledge imparted during my studies.

Finally, we thank all those who have helped us, directly or indirectly, in the completion of this work.

AMEUR lina

Contents

List of acronyms	iv
List of Figures	v
List of symbols	vii
List of Tables	viii
General Introduction	1
1 Generality about Kinship Verification	3
1.1 Introduction	3
1.2 Biometrics	3
1.3 Facial Recognition	5
1.4 Kinship Verification	6
1.4.1 Applications of Kinship Verification	6
1.4.2 Automatic Kinship Verification Challenges	7
1.5 Kinship Verification Datasets	7
1.5.1 KinFaceW Dataset	7
1.5.2 UB KinFace Dataset	8
1.5.3 Cornell KinFace Dataset	8
1.5.4 TSkinFace Dataset	8
1.6 Kinship Verification System	9
1.6.1 Preprocessing	9
1.6.2 Feature Extraction	9
1.6.3 Feature Normalization	10

1.6.4	Classification	10
1.7	Conclusion	11
2	Techniques and Innovations in Kinship Verification Systems	12
2.1	Introduction	12
2.2	Facial Analysis	12
2.3	Preprocessing	13
2.4	Feature Extraction	15
2.4.1	Local Phase Quantization LPQ	16
2.4.2	Histogram of Oriented Gradients HOG	17
2.4.3	Local Binary Patterns LBP	18
2.4.4	Histogram of Local Binary Patterns Hist LBP	19
2.4.5	Color Local Binary Patterns CLBP	20
2.4.6	Binarized Statistical Image Features BSIF	20
2.5	Feature Normalization	21
2.6	Classification	25
2.7	Conclusion	28
3	Implementation and Results	29
3.1	Introduction	29
3.2	KinFaceW-II Dataset	29
3.3	Main process of Kinship Verification System	30
3.3.1	Data Preparation	31
3.3.1.1	Data Preprocessing	33
3.3.1.2	Feature Extraction	33
3.3.1.3	Feature Normalization	36
3.3.2	Feature Fusion	37
3.3.3	Classification	37
3.4	Evaluation Metrics	39
3.4.1	Accuracy	39
3.4.2	Receiver Operating Characteristic	40
3.4.3	Area Under the Curve	41

3.5	Experimental Protocol for Classification	41
3.5.1	Evaluation of Different Descriptors	41
3.5.2	Impact of Different Parameters on the Hist LBP Descriptor	42
3.5.3	Impact of Normalization on the Hist LBP Descriptor	44
3.5.4	Impact of Power Normalization on the Hist LBP Descriptor	45
3.5.5	Impact of Power Normalization on Best Hist LBP Configuration	47
3.6	Comparison with State-of-the-Art Techniques	48
3.7	Application: Graphical user interface	50
3.8	Conclusion	54
	General Conclusion	56
	Bibliography	57

List of Acronyms

- **AUC:** Area Under the Curve
- **BSIF:** Binarized Statistical Image Feature
- **CIE: International Commission on Illumination**
- **CLBP:** Color Local binary patterns
- **DFT:** Discrete Fourier transform
- **FPR:** False Positive rates
- **HOG:** Histogram of Oriented Gradients
- **Hist LBP:** Histogram Local binary patterns
- **LAB:** L: lightness, a: the red/green coordinate, b: the yellow/blue coordinate.
- **LBP:** Local binary patterns
- **LPQ:** Local Phase Quantization
- **MLP:** Multi Layer Perceptrons
- **ROC:** Receiver Operating Characteristic
- **SVM:** Support Vector Machines
- **TPR:** True positive rates

List of Figures

1.1	Biometrics traits [39]	4
1.2	A generic biometric system [13]	5
1.3	Comparison of various biometric traits [40]	6
1.4	A global system structure for kinship verification	9
1.5	Normalization example	10
2.1	A typical facial analysis system[5]	13
2.2	The three components of color: Hue, Value, and Saturation	14
2.3	An example of textures with increased degrees of blurring from left to right[2]	16
2.4	The necessary steps to build the LPQ descriptor	17
2.5	The main steps to calculate the HOG descriptor	17
2.6	Local binary patterns method with gray image	19
2.7	BSIF code person's image under different condition, lightness, blur, gesture	21
2.8	Z-score normalization	23
2.9	Comparison of data before and after being square root transformed [41]	24
2.10	Processing steps for SVM classification	27
3.1	Example of images from the KinFaceW-II dataset as kin relationship	30
3.2	Example of images from the KinFaceW-II dataset as Non-kin relationship	30
3.3	Kinship Verification Process Schema	31
3.4	Pipeline for feature extraction and data preparation from input images	32
3.5	RGB image conversion to different color spaces	33
3.6	Local binary patterns features	34
3.7	Color local binary patterns steps	35
3.8	Difference between patches and step size	36

3.9	The training process for the system to get four models, model-FD, model-FS, model-MD, model-MS	38
3.10	5-fold cross validation explained [42]	39
3.11	Confusion matrix for machine learning	40
3.12	Accuracy estimation using SVM with different feature extractors	42
3.13	ROC curves with Hist LBP for patches=32, bins=16, step size=2	44
3.14	ROC curves for patches=32, bins=16, step size=2 , with power norm normalization	47
3.15	Kinship Verification Framework	50
3.16	The basic user interface	51
3.17	Selecting Parent Image	51
3.18	Selecting Child Image	52
3.19	User Selection of Relationship Type for Verification	52
3.20	The final result (images are likely kin)	53
3.21	The final result (images are likely not kin)	53

List of symbols

- F : Fourier transform.
- G : Gradient function.
- N : Neighborhood.
- σ : Logistic sigmoid.
- Θ : Orientations.
- T : Transpose of matrix.

List of Tables

1.1	Statistic of the KinFaceW database [44].	8
2.1	Difference between LBP and Hist LBP	19
3.1	Contributions of different Kinface-II source	29
3.2	The statistic of the KinFaceW-II dataset [44]	30
3.3	Evaluation of different feature descriptors based on accuracy	41
3.4	Accuracy results using Hist LBP descriptor with change of patches and bins	42
3.5	Accuracy results with step size variation	43
3.6	Mean accuracy after different normalization techniques, step size=8	45
3.7	Mean accuracy results from power normalization by adjusting step size parameter	46
3.8	Comparison of Hist LBP Accuracy: Power Normalization vs. No Normalization, patches=32, bins=16, step size=2	47
3.9	Comparison of Hist LBP AUC: Power Normalization vs. No Normalization, patches=32, bins=16, step size=2	48
3.10	Comparison of Hist LBP descriptor with State-of-the-Art Techniques	49

General Introduction

Recognizing familial relationships from facial images is a growing field of computer vision with immense potential. Kinship verification technology analyzes facial features to determine if two individuals are related. This technology holds promise for various applications, such as reuniting missing children with their families, enhancing security measures, and aiding genealogical research.

This work explores an interesting approach to kinship verification using a feature extraction technique called Histogram of Local Binary Patterns (Hist LBP), which is initially applied in the field of person re-identification. Traditional methods often rely on extracting specific features like shapes, colors, or textures individually. The Hist LBP extractor goes beyond this limitation by capturing more nuanced and informative characteristics from facial images. The key advantages of Hist LBP are its effectiveness in merging texture and color information into a single descriptor, potentially leading to a richer feature representation. Furthermore, by incorporating the Hist LBP descriptor, we believe the Hist descriptor approach can improve the accuracy and robustness of kinship verification systems compared to traditional methods. In addition, we explore the impact of different normalization techniques applied to the Hist LBP descriptor. By investigating the effectiveness of these techniques, we aim to further enhance the performance of the kinship verification system.

This thesis explores the details of the Hist LBP technique, including its implementation and theoretical underpinnings. We evaluate its performance on the KinFaceW-II kinship verification benchmark and discuss its potential impact on the field. Additionally, for comparison purposes, we evaluate the performance of the Local Phase Quantization (LPQ) and traditional Local Binary Patterns (LBP) descriptor in different color spaces (Grayscale, RGB and HSV).

The present document is organized into three main chapters to provide a comprehensive understanding of our approach:

Chapter 1: This chapter lays the groundwork by introducing the concepts of biometrics, facial recognition, and facial kinship verification. We then proceed to examine the challenges associated with this task and explore potential applications in various domains. In addition, we summarize existing kinship verification datasets.

Chapter 2: This chapter focuses on feature extraction techniques, a crucial step in kinship verification. We explore various techniques employed in the field, including color space conversion, normalization methods, and classification approaches like support vector machines.

Chapter 3: The final chapter focuses on the practical implementation of our kinship verification system. It details the system architecture, highlighting the key components and their functionalities. In particular, we examine the utilization of Histogram of Local Binary Patterns (Hist LBP) and other techniques for feature extraction and present our evaluation using the KinFaceW-II kinship database. Furthermore, it discusses the evaluation metrics used to assess

the performance of kinship verification systems. This chapter is concluded with a presentation of comprehensive analysis of our system's performance. This analysis includes comparisons with existing state-of-the-art approaches to highlight the effectiveness of the method studied. Additionally, the chapter showcases the graphical user interface (GUI) developed for this system, enhancing user interaction and accessibility.

Chapter 1

Generality about Kinship Verification

1.1 Introduction

In this chapter, we provide a general overview of kinship verification, a process that determines whether two individuals are related to each other or not. Kinship verification is a crucial application in various domains, including biometrics, computer vision, and artificial intelligence. The chapter begins by discussing the concept of biometrics, which is a method of identifying individuals based on their unique physical or behavioral traits. It highlights the advantages of biometrics over traditional identification methods, such as passwords or ID cards, due to their unique nature and difficulty in imitation.

We then delve into facial recognition, which is a significant aspect of kinship verification. Facial recognition has numerous benefits over other biometric modalities, including its ability to be discreetly and remotely recorded, and its natural and unobtrusive nature. We also discussed the applications of kinship verification, including its potential use in finding kin relationships from big data resources and in authenticating guardians with children for security purposes.

1.2 Biometrics

A biometric is a method of identifying individuals based on their unique physical or behavioral traits. These traits include fingerprints, facial features, iris, retina, voice, gait, and even the way we type! (Figure 1.1 illustrates this concept) [13]. It is deemed more reliable and secure due to the unique nature of physiological or behavioral traits, making it difficult for someone to fake or steal these characteristics compared to passwords or ID cards. Consequently, biometrics is a more effective method for distinguishing between authorized users and fraudulent impostors.

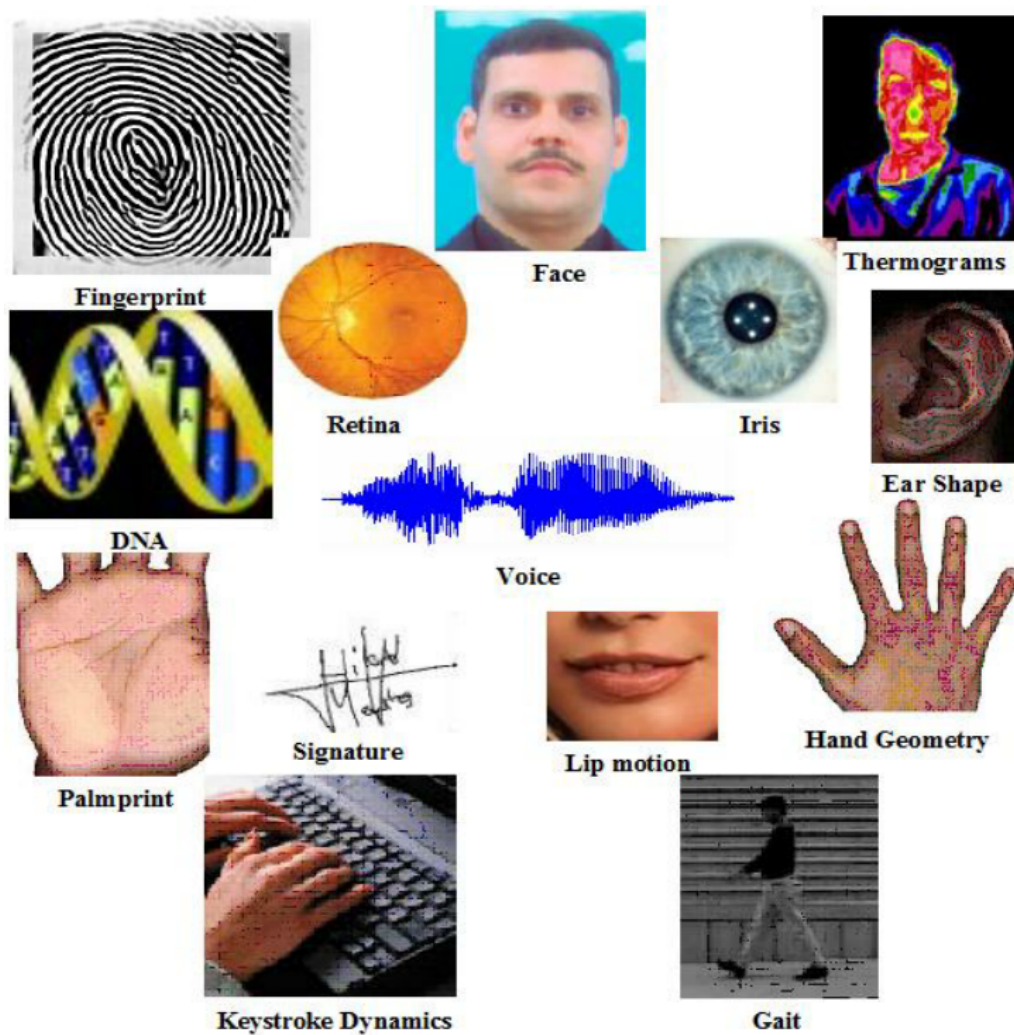


Figure 1.1: Biometrics traits [39]

- **Biometric System**

Biometric system is a pattern recognition tool, identifying by verifying our physiological or behavioral traits. Figure 1.2 presents a block diagram of a generic biometric system. The enrollment module registers individuals into the system, scanning their biometric traits with a reader to create a raw digital representation. This representation is then processed by a feature extractor (which will be discussed in detail in chapter 2) to produce a concise yet informative representation for matching purposes.

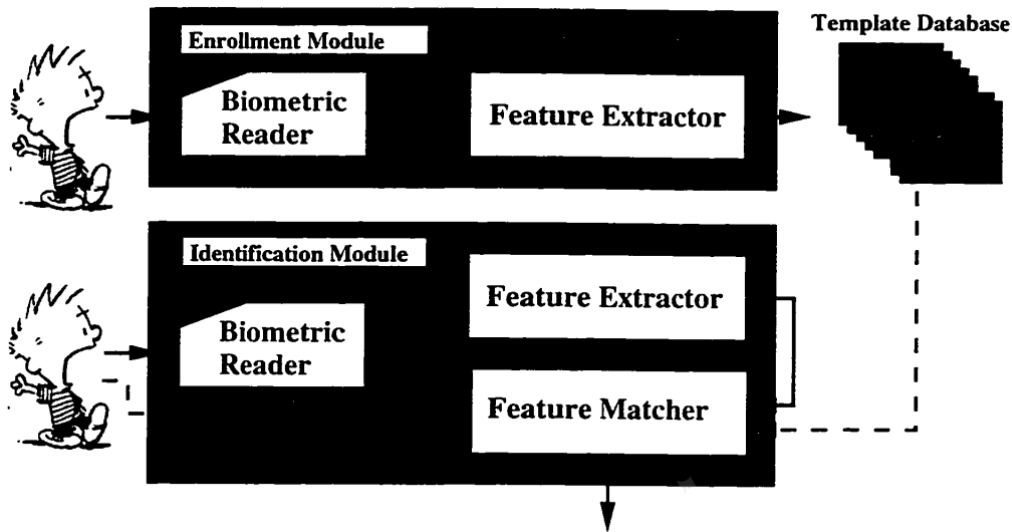


Figure 1.2: A generic biometric system [13]

1.3 Facial Recognition

People perform face recognition every day with ease. Since the widespread availability of powerful, low-cost computing systems, there has been a great deal of interest in the automatic processing of digital photographs. This interest drives research and development in automatic face recognition, spanning a variety of applications such as biometric authentication, surveillance, human-computer interaction, and multimedia management, driving research and development in automatic face recognition [40].

Face recognition has a number of benefits over other biometric modalities like fingerprint and iris recognition. It can be discreetly and remotely recorded, and it is natural and unobtrusive. As shown in figure 1.3, face recognition, has gained increasing importance due to rapid advancements in image capture devices (such as surveillance cameras and cameras in mobile phones). Automatic kinship verification is a typical pattern recognition task that relies on facial recognition technology.

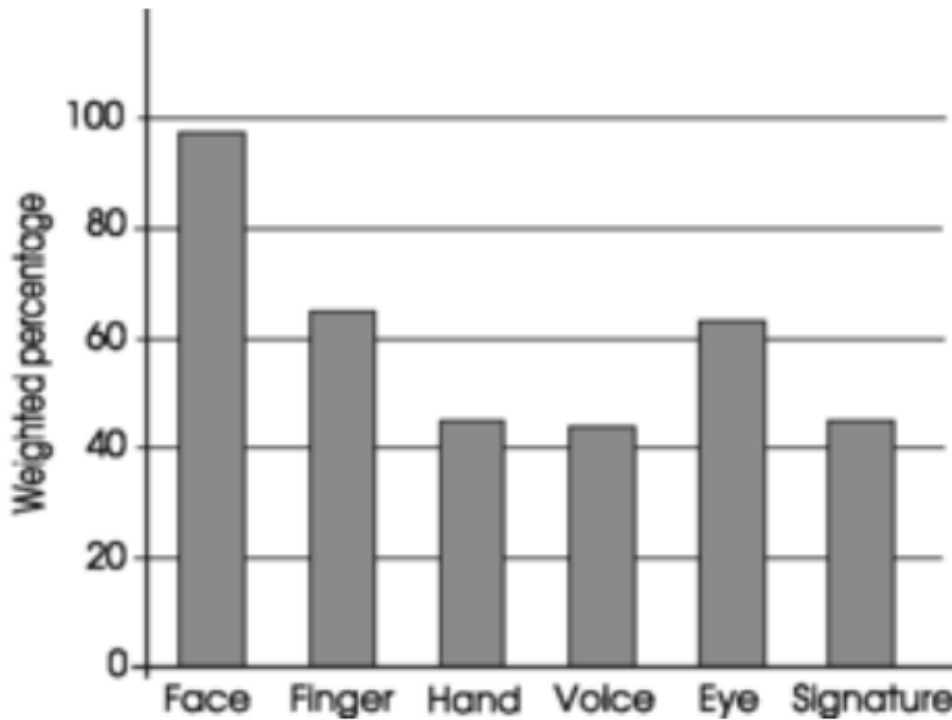


Figure 1.3: Comparison of various biometric traits [40]

1.4 Kinship Verification

Kinship verification (or family verification) is the process of determining whether two individuals are related or not. Automatic kinship verification systems typically take two faces (Face A and Face B) as input and aim to predict whether Person A is related to Person B as a family member (father, mother, brother, sister, etc.) [18, 2]. This technology has applications in various areas, which we will discuss in the next section.

Facial analysis is one of the most promising approaches for automatic kinship verification. This is due to the fact that human faces offer several advantages: a distinct and hierarchical component structure, easy distinction from the background, and the ability to convey a wealth of information about individuals, such as gender, age, and emotional state. As a result, facial image analysis has received significant attention in computer vision research over the past fifteen years.

1.4.1 Applications of Kinship Verification

Kinship verification methods hold vast potential across various domains. While the technology is still under development, advancements are paving the way for exciting new applications.

One promising area lies in leveraging kinship verification for large-scale data analysis. For example, utilizing these methods to identify familial connections within massive genealogy databases, potentially aiding humanitarian organizations in reuniting missing persons with their families.

Furthermore, kinship verification could play a crucial role in enhancing security measures. In public spaces like schools or hospitals, it could be employed to verify the identity of guardians accompanying children, adding an extra layer of safety [45].

Kinship verification can also be used in online user identification to combat impersonation and fraud. Also, it has the potential to help identify victims of human trafficking by verifying or disproving claimed relationships between individuals.

In addition, kinship verification offers a faster and more cost-effective way to support DNA verification in establishing familial relationships [6].

1.4.2 Automatic Kinship Verification Challenges

The problem of kinship verification arises from **the small size of existing datasets** [2]. In this case, traditional hand-crafted features such as Local Binary patterns(LBP), Histogram of oriented Gradient (HOG), and Scale-invariant feature transform (SIFT) are not sufficient to adequately represent facial images. On the other hand, deep convolutional neural networks(CNN) provide an answer by enabling us to customize the feature extraction procedure to the specific needs of kinship verification. Although this method can produce more reliable features, it requires a large dataset of labeled samples for training, which is often lacking in the kinship verification space. Furthermore, unlike face recognition, kinship verification requires additional information about the persons being matched, making it difficult to annotate a big kinship dataset. Kinship data collection is further complicated by copyright and privacy issues.

Another difficulty in kinship verification is **the large intra-class variation**. Because of disparities in age, gender, and genetic inheritance, people with positive kinship relationships can have very different facial appearances from one other. Further complicating the verification process are variations in shape, such as change in posture and facial expression, or disparities in the parts of the face that facial identification systems capture.

There are also two other difficulties, **shape variations** and **illumination changes**, where the first one starts from the fact that people in facial photographs won't always be in the same posture or attitude, and there can be minute variations in the facial regions that face identification algorithms pick up on. The second factor to consider is that when extracting the faces of a child and a parent from two different source photos, the luminosity of the images may differ, and local light reflections can cause shifts in the colors of the pixels in the image.

1.5 Kinship Verification Datasets

Kinship verification datasets are crucial for training and evaluating algorithms used to determine whether two individuals are related. Here are some popular datasets to consider:

1.5.1 KinFaceW Dataset

The KinFaceW dataset, collected by Lu et al. [23], plays a vital role in studying kinship verification using unconstrained face images. Unconstrained images can vary greatly in lighting, pose, and background, making them more realistic but also more challenging for algorithms.

KinFaceW is divided into two subsets:

- **KinFaceW-I:** This subset contains 1066 images forming 533 pairs. The images in these pairs are likely captured under different conditions (different photos) but still represent kinship relationships.
- **KinFaceW-II:** This subset is larger, containing 2000 images forming 1000 pairs. Here, the images within each pair are more likely captured under similar conditions (same photo).

In the table below 1.1, the comparison between both KinFaceW databases is shown.

Table 1.1: Statistic of the KinFaceW database [44].

	KinFaceW-I	KinFaceW-II
Image	1066 images – 533 pairs	2000 images – 1000 pairs
Asian people %	64.40	8
Not Asian people %	35.60	92
Source image	Many images	One image
Image size	64×64 pixel	64×64 pixel

1.5.2 UB KinFace Dataset

The UB KinFace database [48] is a valuable resource for kinship verification research. It contains 600 images of 400 people, specifically divided into two sets:

Set 1: This set comprises 200 pairs featuring a child and their young parent.

Set 2: This set includes 200 pairs of a child with their older parent.

1.5.3 Cornell KinFace Dataset

The Cornell KinFace dataset [8] is a collection of images specifically designed for kinship verification research. It consists of 143 pairs of parents and children, totaling 286 individual cropped frontal face images. Each image has a resolution of 100 x 100 pixels.

1.5.4 TSkinFace Dataset

The TSkinFace database [35] offers a unique collection of facial images designed to study tri-subject kinship relationships. It focuses on two specific family structures:

Father-Mother-Daughter (FM-D): This category contains 502 relationships, totaling 1506 individual face images (502 sets of 3 images).

Father-Mother-Son (FM-S): This category encompasses 513 relationships, translating to 1539 individual face images (513 sets of 3 images).

1.6 Kinship Verification System

Any typical kinship verification system has to apply some steps, as it is resumed in the figure 1.4 :

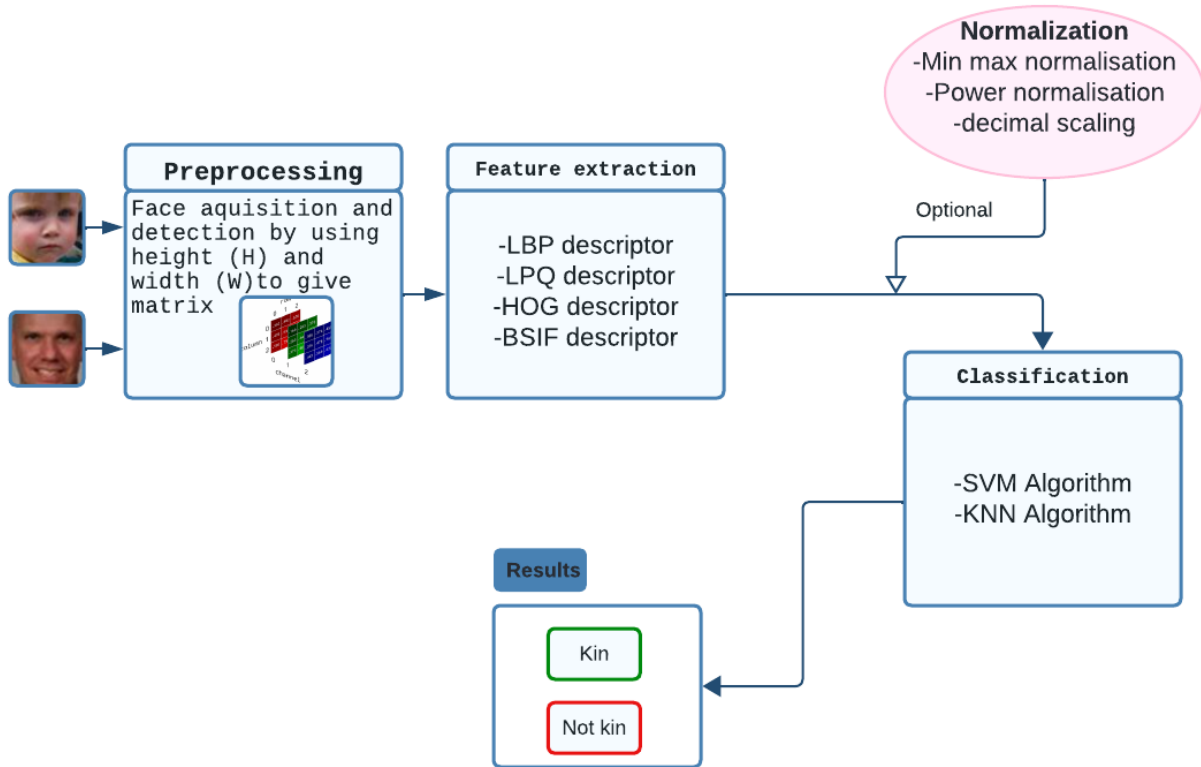


Figure 1.4: A global system structure for kinship verification

1.6.1 Preprocessing

Preprocessing is a crucial step in image processing that involves preparing the raw image data for further analysis or processing. The main goals of preprocessing are to improve the quality of the image, remove unwanted distortions or noise, and enhance specific features of interest. Common techniques include noise removal using filters, **resizing** and **resampling** images to required dimensions, and enhancing images by adjusting contrast, brightness, and sharpness. Also an important task which is **Color Space Conversion** because some algorithms may work better in specific color spaces (e.g., RGB, Grayscale, HSV, or YCbCr), and also **image registration** when we use a database so it converts each image to a vector and collects all of them in one matrix.

1.6.2 Feature Extraction

During the kinship verification process, each facial image is analyzed to extract descriptive features. These features are then converted into numerical representations, typically forming two feature vectors - one for each face being compared. These feature vectors capture essential characteristics of the facial images, essentially acting as a condensed yet informative summary of the visual data [43].

Several well-established techniques are commonly used for feature extraction in this context. We will delve deeper into these methods, including Local Binary Patterns LBP, Local Phase Quantization LPQ, Histogram LBP Hist LBP, Histogram of Oriented Gradients HOG, Binarized Statistical Image Features BSIF. We will detail more in the next chapter.

1.6.3 Feature Normalization

After feature extraction, the resulting data often contains values across varying ranges. To ensure all features contribute equally to the kinship verification process, we employ a technique called feature normalization. This mathematical approach transforms or rescales the extracted features into a specific range, commonly between 0 and 1 or -1 and 1 (similar to Figure 1.5).

Normalization ensures that all features operate on a comparable scale, preventing any single feature from dominating the analysis due to its original range [27]. This is crucial for achieving fair and accurate kinship verification results, as we will see in the results chapter.

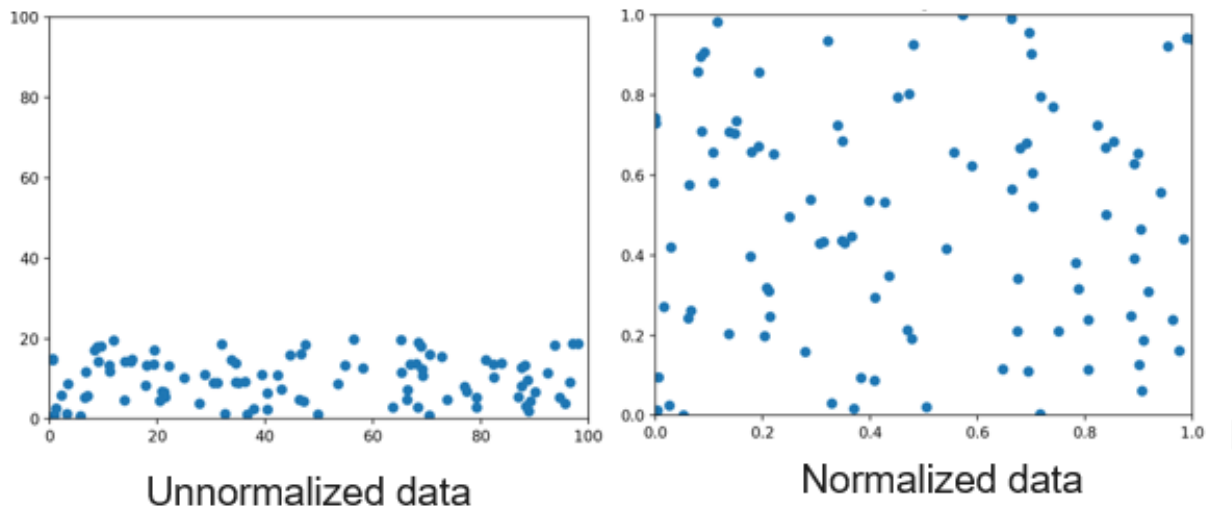


Figure 1.5: Normalization example

1.6.4 Classification

Classification, is the process of assigning a label or category to an image or regions within an image based on their visual characteristics or content. Image classification is commonly used in applications such as object detection (identifying and classifying objects within an image), facial recognition (classifying images of faces into identities), and other computer vision tasks such as car driving [36, 25].

Classification algorithms analyze the pixel values, textures, shapes, colors, and other characteristics of an image to determine which class or category it belongs to. Image classification algorithms come in various flavors, each with its own strengths. Choosing the right technique depends on the complexity of the task and the available data [43].

The effectiveness of image classification, like any learning process, depends on several key ingredients [43]:

- **High quality training data:** An image classification algorithm thrives on a large dataset of well-labeled images. The more high-quality, accurately labeled images the algorithm is trained on, the better it learns to recognize patterns and make accurate classifications.
- **Task difficulty:** Not all classification tasks are created equal. Distinguishing between a parent and a child is an easier task than identifying the relationship between them. The complexity of the classification problem affects the accuracy achievable by the algorithm.
- **Feature extraction and model selection:** Selecting the right features and employing powerful models are critical to optimal performance. These factors determine how well the algorithm recognizes patterns for accurate classification.

1.7 Conclusion

This chapter has provided a comprehensive overview of kinship verification, emphasizing its significance in various domains. We have highlighted the importance of biometrics and facial recognition in kinship verification, as well as the challenges and potential applications of this process. It sets the stage for further exploration of the technical aspects of kinship verification in subsequent chapters.

Chapter 2

Techniques and Innovations in Kinship Verification Systems

2.1 Introduction

Kinship verification is an important problem in computer vision and machine learning, with applications in areas like family photo organization, social network analysis, and forensics. This chapter discusses the key techniques and innovations in kinship verification systems. It covers the main steps involved, including face acquisition, feature extraction, normalization, and classification.

We started the chapter by explaining the face acquisition process, which involves detecting and extracting faces from images. The bulk of this chapter focuses on feature extraction techniques that have been applied to kinship verification, such as Local Phase Quantization (LPQ), Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and Binarized Statistical Image Features (BSIF). These methods capture different types of facial information that can be useful for distinguishing between related and unrelated individuals.

The chapter also covers normalization techniques like z-score, decimal scaling, and square root normalization, which are important steps to ensure the features are on a common scale.

Finally, we discuss classification approaches for relationship verification, including support vector machines.

2.2 Facial Analysis

Facial analysis is divided into two parts, **face detection** and **face recognition** [5], face detection is a necessary initial step to localize and extract the face region from the background. The solution of the problem involves segmentation and extraction of faces and possibly facial features from an uncontrolled background. The face recognition process involves performing verification and identification (age, gender...), this stage takes the probe image extracted from the scene during the face detection stage and compares it with the database of previously enrolled known faces to detect kinship between them as shown in the figure 2.1 below.

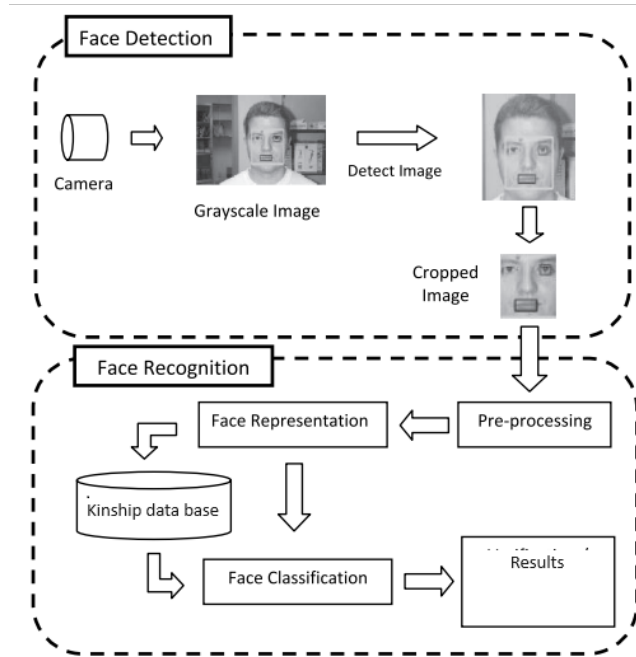


Figure 2.1: A typical facial analysis system [5]

2.3 Preprocessing

Kinship image preprocessing is a crucial step in kinship verification systems. It involves preparing the facial images before feeding them into the model for analysis. The goal is to ensure consistent data representation and improve the effectiveness of the model in recognizing kinship relationships. Different algorithms might function better in specific color spaces like RGB (standard red, green, blue), grayscale, HSV (hue, saturation, value), or YCbCr (luminance and chrominance components). Converting images to a suitable color space can improve feature extraction and model performance.

- Color Space Conversion

A color space, also known as a chromatic or colorimetric space, is a system for representing and organizing all the colors that the human eye can see. It is a three-dimensional model in which each dimension corresponds to a different attribute of color: Hue (the type of color, like red or blue), Saturation (the intensity or clearness of the color), and Brightness (how light or dark the color is, ranging from black to white) [11]. The figure 2.2 below illustrates the three color space components.



Figure 2.2: The three components of color: Hue, Value, and Saturation

- **Grayscale Color Space:** A grayscale color space represents colors using only black, white, and gray levels. Each color is defined by a single brightness value, where 0 is black and 255 is white. Grayscale images, also called monochrome images, use only brightness and they are common in medical photography and black-and-white printing. This color space simplifies image processing by reducing the amount of data needed compared to full-color spaces [11].
- **RGB Color Space:** This color space uses the three primary colors (RGB) —Red, Green, and Blue— to represent all the colors visible to the human eye. The concept is simple: each color is defined by three values that represent the intensity of each primary color. These values are typically encoded in 8 bits (ranging from 0 to 255), allowing for the representation of 256^3 (16,777,216) different colors [11].
- **Image Conversion from RGB to Grayscale:** Converting an RGB image to grayscale removes the hue and saturation information, retaining only the luminance. The luminance, which represents the brightness of each pixel, is obtained by combining the brightness information from the red, green, and blue channels of the RGB image [11].
- **HSV Color Space:** It represents colors based on three components: Hue, which represents the pure color and is measured in degrees on a color wheel; Saturation, which measures the intensity or purity of the color; and Value, which represents the brightness of the color. This model is more intuitive to human perception, with hue specifying the type of color, saturation indicating how vivid or muted it is, and value denoting how light or dark it is [4].
- **YCbCr Color Space:** The YCbCr color space, also known as YCbCr, separates luminance (Y) from chrominance (Cb and Cr). Luminance (Y) represents the brightness or intensity of light as perceived by the human eye, independent of color, ranging from 0 (black) to 255 (white). Chrominance Blue-difference (Cb) indicates the difference between the luminance and the amount of blue in the image, with values ranging from -128

(maximum blue) to 127 (no blue). Chrominance Red-difference (Cr) shows the difference between the luminance and the amount of red in the image, with values from -128 (maximum red) to 127 (no red) [4].

- **XYZ Color Space:** The CIE (International Commission on Illumination) XYZ color model, also known as the CIE 1931 model, is a three-dimensional coordinate system for representing colors. It was defined by the International Commission on Illumination (CIE) in 1931 and is the reference for many other color models. The CIE XYZ model uses three components, X, Y, and Z, called tristimulus values, which measure the amounts of red, green, and blue light reflected or emitted by an object, weighted by the human eye's spectral sensitivity. The X component represents the red light contribution to color perception, Y corresponds to luminance or perceived brightness, and Z represents the blue light contribution [12].
- **LAB Color Space:** The LAB color space is characterized by its perceptually uniformity, which ensures that equal variations in this space result in consistent color perception for the average human observer. This property makes it a valuable tool for color management and the accurate color reproduction across various domains. It consists of three main components: Luminance (L), representing brightness from absolute black to absolute white; the A-axis, indicating the hue from pure red to pure green along a scale from -128 to 128; and the B-axis, representing saturation from pure blue to pure yellow the on a similar scale [12].

The preprocessed data is then fed into the feature extraction stage, where it is transformed into a more compact and informative representation.

2.4 Feature Extraction

Feature extraction is one of the most important fields in artificial intelligence. It involves transforming the raw facial image data into a set of informative features [17]. It is a special form of dimensionality reduction. The main goal of feature extraction is to obtain the most relevant information from the original data and represent that information in a lower dimensionality space, when the input data to an algorithm is too large to be processed, and it is suspected to be redundant (much data, but not much information) then the input data will be transformed into a reduced representation set of features (also named features vector).

In simpler terms, feature extraction is like identifying the key details in an image. When classifying images for kinship verification, these details are crucial for grouping similar faces together. The accuracy of the system heavily relies on extracting the most discriminative features, essentially the elements that differentiate between related and unrelated individuals. Various techniques exist for feature extraction, each with its own advantages and limitations. Choosing the right approach allows us to capture the most valuable information from facial images for kinship verification tasks [28].

Facial image features can be broadly categorized into four main types:

- **Geometric Features:** These features focus on the spatial relationships between facial landmarks such as eyes, nose, and mouth.
- **Statistical Features:** Statistical features capture information like average pixel intensity or standard deviation within specific facial regions.

- **Color Features:** Color features analyze the distribution of colors within the facial image.
- **Texture Features:** Texture features describe the spatial variation of pixel intensities, often used to represent wrinkles, skin roughness, or other textural patterns. This is the focus of the following chapter.

This section will explore specific image descriptors commonly used in kinship verification research. It's important to note that while these descriptors were originally developed for texture classification, they have also demonstrated effectiveness in representing facial images for kinship verification tasks.

2.4.1 Local Phase Quantization LPQ

The Local Phase Quantization (LPQ) descriptor, introduced by Ojansivu et al. [2], was originally designed for classifying blurred textures (refer to the figure 2.3). This descriptor computes the local 2D Discrete Fourier Transform within local neighborhoods N_x of a specified size $M \times M$ at each position x in the image f using the equation (2.1)

$$F(u, x) = \sum_{X_x \in N_x} f(x - y) e^{-i2\pi u^\top y} = w_u^\top f_x \quad (2.1)$$

Where :

x is the central position of the neighborhood in the image.

y is the relative offset within that neighborhood.

$i^2 = -1$, w_u is the basis vector of the 2D DFT at frequency u , and f_x is a vector containing all samples in N_x .

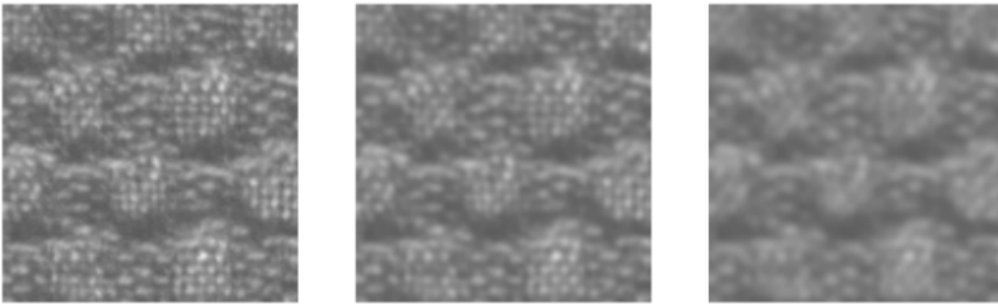


Figure 2.3: An example of textures with increased degrees of blurring from left to right [2]

The Fourier coefficients of only four low 2D frequencies $\{u_1, u_2, u_3, u_4\}$ are used by the LPQ to form an eight-dimensional complex vector F_x as indicated by the equation (2.2).

$$F_x = [\text{Re}(F(u_1, x)), \dots, \text{Re}(F(u_4, x)), \text{Im}(F(u_1, x)), \dots, \text{Im}(F(u_4, x))]^\top \quad (2.2)$$

Where, $\text{Re}(c)$ and $\text{Im}(c)$ are the functions that return the real and imaginary parts, respectively, of the complex number c .

In the subsequent step, the phase components of the four 2D DFT complex coefficients in F_x are decorrelated and quantized into an eight-dimensional binary vector. Finally, the binary vector is converted into an 8-bit integer code as shown in the figure 2.4

The histogram of the LPQ 8-bit integer codes serves as the feature vector representing the input image.

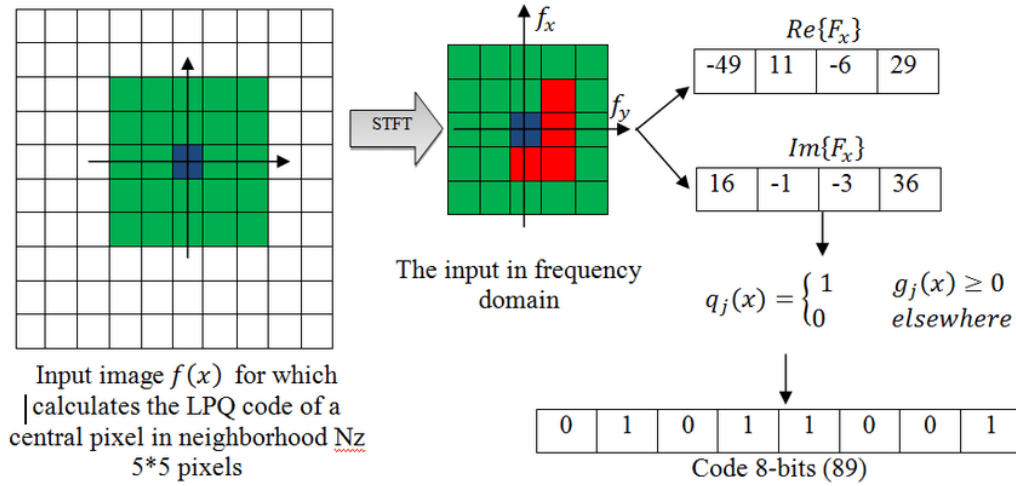


Figure 2.4: The necessary steps to build the LPQ descriptor

2.4.2 Histogram of Oriented Gradients HOG

A Histogram of Oriented Gradients (HOG) is a tool used in computer vision to detect objects in images. It works by looking at how the direction of color changes (gradients) are distributed in different parts of the image. HOG is good at recognizing shapes because it considers both the strength and direction of these color changes. It creates histograms, or graphs, of these color changes in different areas of the image to understand the object's structure better. to calculate the HOG, we follow steps as shown in the figure 2.5.



Figure 2.5: The main steps to calculate the HOG descriptor

1. **Preprocessing:** To resize the input image, it is recommended to use dimensions of **128x64** pixels. This recommendation stems from the need to combine two images, each of size 64x64 pixels, resulting in a total width of 128 pixels.
2. **Compute Gradient and Orientation:** The image is divided into small cells, such as 4x4 cells. Each cell is then processed individually to extract features.

First, the algorithm calculates the gradient in the x and y directions (G_x and G_y) for each pixel in the cell. G_x represents the difference in pixel values between the right and left neighbors, while G_y represents the difference between the top and bottom neighbors. These gradients are calculated using the formulate (2.3).

$$\begin{aligned} G_x(r, c) &= I(r, c + 1) - I(r, c - 1) \\ G_y(r, c) &= I(r - 1, c) - I(r + 1, c) \end{aligned} \quad (2.3)$$

Next, the algorithm combines these gradients to compute the total gradient magnitude and orientation for each pixel in the cell. The magnitude is calculated using the

Pythagorean theorem, which gives the square root of the sum of the squares of G_x and G_y . The orientation is calculated as shown in equation (2.4)

$$\begin{aligned} |G(r, c)| &= \sqrt{G_x(r, c)^2 + G_y(r, c)^2} \\ \theta(r, c) &= \arctan\left(\frac{G_y(r, c)}{G_x(r, c)}\right) \end{aligned} \quad (2.4)$$

This process is repeated for all pixels in the cell, resulting in a set of gradient magnitudes and orientations. For edge pixels, padding techniques are used to ensure that all pixels can be processed in the same way.

2.4.3 Local Binary Patterns LBP

The Local Binary Patterns (LBP) is a way to describe textures in pictures [33]. It looks at each pixel and compares it to its nearby pixels to decide if it's light or dark, creating a code for that pixel. This method is popular because it's good at telling different textures apart and is fast to compute. It's like a common ground between different ways of analyzing textures. One big advantage is that it can handle changes in brightness well, like when the lighting in a picture changes. Also, it's quick, so it can be used to look at images in real-time, even in tricky situations.

For each pixel $I(x,y)$, we calculate LBP following these steps:

1. Comparing a pixel's intensity to its eight neighbors in a small square around it (3x3 neighborhood).
2. Thresholding each neighbor: if the neighbor's intensity is **greater than** the center pixel's intensity, it becomes a 1 in a binary code. Otherwise, it becomes a 0, according to (2.5).

$$f(d) = \begin{cases} 1 & \text{if } d \geq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (2.5)$$

Where, d is the difference between the currently sampled pixel and the central pixel.

3. Combining the binary codes of all the neighbors into a single number, essentially creating a label for the center pixel's texture.
4. The binary code is converted to a decimal value, which represents **the LBP code** for that pixel.
5. Creating a histogram that shows how many times each label appears in the image. This histogram acts as a fingerprint for the image's texture (2.6).

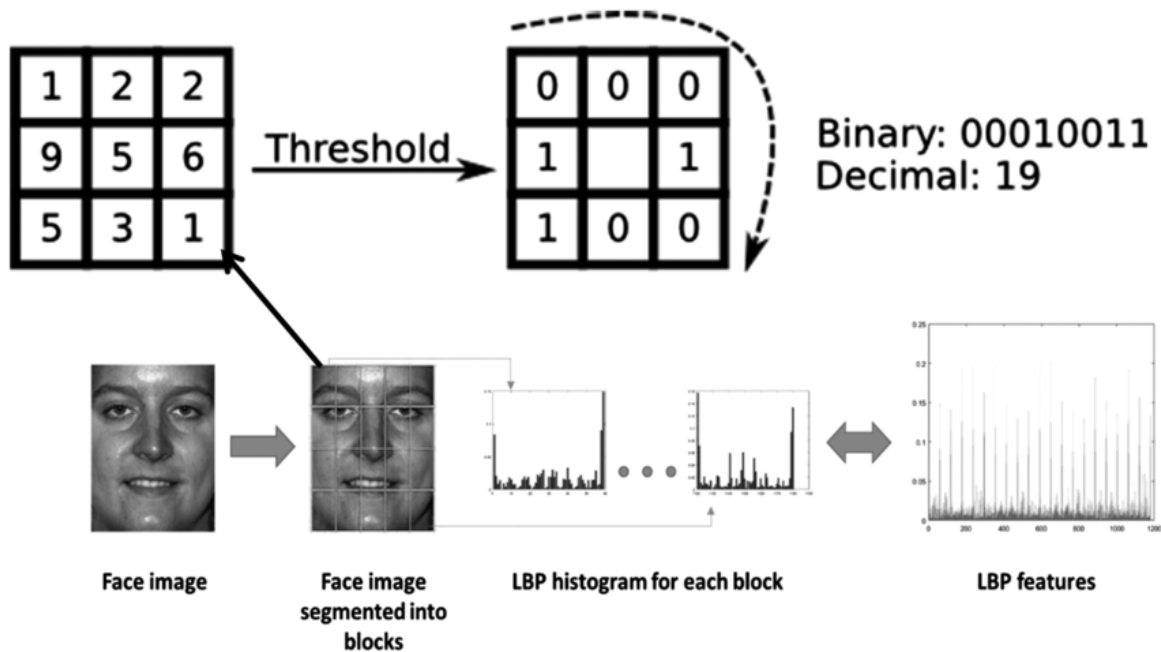


Figure 2.6: Local binary patterns method with gray image

2.4.4 Histogram of Local Binary Patterns Hist LBP

The Histogram of Local Binary Patterns (Hist LBP) is a texture-color feature extraction technique initially proposed for the person re-identification domain [49]. After calculating the LBP descriptor of an image, the LBP histogram counts how many times each unique LBP code appears in the entire image. So, each bin in the histogram represents a specific texture pattern, and its height shows how prevalent that pattern is in the image. By analyzing this histogram, we can gain insights into the dominant textures present in the image. We can explain the difference between simple LBP and Hist LBP on the table 2.1 below:

Table 2.1: Difference between LBP and Hist LBP

	LBP	Hist LBP
Representation	local texture patterns for individual pixels	global texture patterns
Information	detailed information about texture patterns	abstract view of the overall texture distribution

When the histogram of LBP codes is computed, after capturing the frequency of occurrence of each LBP code. The LBP histograms of individual cells are optionally normalized (e.g. L2-norm) to account for illumination variations. The normalized histograms of all cells are concatenated to form the final LBP histogram feature vector for the image or region of interest. The dimensionality of the LBP histogram feature vector depends on the number of distinct LBP codes possible, which is determined by the neighborhood size and the number of neighboring pixels considered.

For example, with a 3x3 neighborhood, there are $2^8 = 256$ possible LBP codes, resulting in a 256 dimensional histogram. The LBP histogram is effective at capturing texture information because it encodes the local spatial structure of the image while being invariant to monotonic

gray-level transformations (e.g., changes in illumination). However, it is sensitive to rotation and scaling, which can be addressed by incorporating rotation-invariant and multi-scale LBP variants.

2.4.5 Color Local Binary Patterns CLBP

In the color LBP method, we first separate each color channel (like red, green, and blue) and apply the LBP (Local Binary Patterns) algorithm to each of these channels separately. After that, we combine the histograms of these separate LBP computations into a single histogram. in 3.3.1.2 here is more explanation.

2.4.6 Binarized Statistical Image Features BSIF

The Binarized Statistical Image Features BSIF is a type of descriptors similar to LBP and LPQ, it calculates a binary code string for the pixels of a given image by thresholding the responses of a set of n linear filters [16]. The main method consists of create a binary code for each pixel in an image by using different filters. Each filter calculates a value for the pixel, which is then converted to either 0 or 1 based on whether it's above or below zero. The number of filters used depends on how long we want the binary code to be. These filters are learned from a training set of natural image patches, ensuring that they give us different values for different parts of the image. This process relies on the statistical properties of real images to create descriptors, therefore we call them binarized statistical image features BSIF.

We can express the process mathematically as follows [18]:

Let's assume an image patch of size $l \times l$ represented by a vector $x \in \mathbb{R}^{l^2}$ and a filter $w_i \in \mathbb{R}^{l^2}$, the response s_i is calculated by the equation (2.6).

$$s_i = w_i^\top x \quad (2.6)$$

Then, the i th bit b_i of the bit string is calculated by the equation (2.7).

$$b_i = \begin{cases} 1 & \text{if } s_i > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.7)$$

The set of filters w_i is created by maximizing the statistical independence of their responses s_i . Initially, these filters w_i for $i = 1, 2, \dots, n$ can be arranged into a matrix $W \in \mathbb{R}^{n \times l^2}$. We can then express equation (2.6) as equation (2.8).

$$s = Wx \quad (2.8)$$

As shown, the vector s contains all filter responses. Then, the matrix W is decomposed into two matrices, $U \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{n \times l^2}$ as $W = UV$. The matrix V is derived as a preprocessing step learned from a set of natural images in the following manner: Firstly, a set of image patches x is randomly sampled from the training images. Then, the covariance matrix C of these patches is computed. After that, the eigenvalue decomposition of C is calculated as $C = EDE^\top$, where D contains the eigenvalues of C in descending order along its main diagonal. Next, V is defined as the first n rows of the matrix $D^{-\frac{1}{2}}E^\top$. By setting $z = Vx$ (thus $s = Uz$), the matrix U can be obtained using the standard Independent Component Analysis (ICA) of the preprocessed samples z .

NB: This process allows us to decompose the original filter responses into statistically independent components. Figure 2.7 shows the application of BSIF code in different conditions.

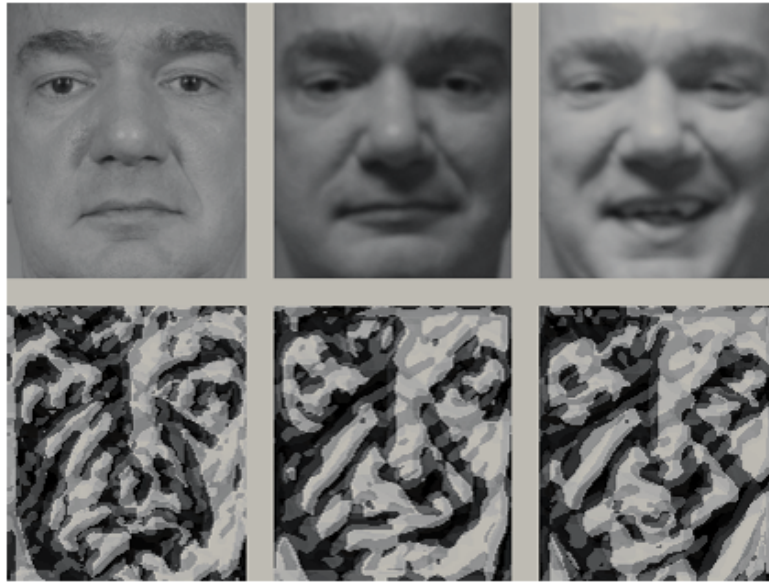


Figure 2.7: BSIF code person's image under different condition, lightness, blur, gesture

To enhance the discriminative power of the feature vector extracted in the feature extraction step, a normalization step is often applied.

2.5 Feature Normalization

As mentioned before, normalization is a step in preparing data where we adjust the values of different features to be on a similar scale. This prevents any one feature from having too much influence just because of its range, making machine learning models work better and more reliably.

- **Min-Max Normalization:** Or Re-scaling is one of the most famous methods to normalize data, For every feature, the minimum value of that feature gets transformed into a 0, the maximum value gets transformed into a 1, and every other value gets transformed into a decimal between 0 and 1 [38], following the formulate 2.9 below.

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2.9)$$

Algorithm 1 Min-Max Normalization [37]

Require: X (dataset with m instances and n features)

Ensure: $X_{\text{minmax_scaled}}$ (Min-Max Scaled dataset)

```

1: for  $j = 1$  to  $n$  do
2:    $\text{min\_value}_j \leftarrow \min(X[:, j])$            ▷ Find the minimum value of feature  $j$ 
3:    $\text{max\_value}_j \leftarrow \max(X[:, j])$        ▷ Find the maximum value of feature  $j$ 
4:    $\text{range}_j \leftarrow \text{max\_value}_j - \text{min\_value}_j$    ▷ Calculate the range of feature  $j$ 
5:   if  $\text{range}_j == 0$  then
6:     for  $i = 1$  to  $m$  do
7:        $X_{\text{minmax\_scaled}}[i, j] \leftarrow 0$            ▷ Handle cases where range is zero
8:     end for
9:   else
10:    for  $i = 1$  to  $m$  do
11:       $X_{\text{minmax\_scaled}}[i, j] \leftarrow \frac{X[i, j] - \text{min\_value}_j}{\text{range}_j}$    ▷ Scale each value to the range  $[0, 1]$ 
12:    end for
13:  end if
14: end for
15: return  $X_{\text{minmax\_scaled}}$ 
    
```

- **Z-Score Normalization:** It also called as standardization, in purpose to transform the data such that it has a mean of 0 and a standard deviation of 1 [38], see figure 2.8. The z-score value of each feature is calculated as:

$$z = \frac{x - \mu}{\sigma} \quad (2.10)$$

Where:

- x is the original data point
- μ is the mean of the data distribution
- σ is the standard deviation of the data distribution

Here is the algorithm of Z-score normalization:

Algorithm 2 Z-score Normalization [37]

Require: X (dataset with m instances and n features)

Ensure: $X_{\text{z_scored}}$ (Z-score normalized dataset)

```

1: for  $j = 1$  to  $n$  do
2:    $\mu_j \leftarrow \text{calculateMean}(X[:, j])$            ▷ Calculate mean of feature  $j$ 
3:    $\sigma_j \leftarrow \text{calculateStandardDeviation}(X[:, j])$    ▷ Calculate standard deviation of feature  $j$ 
4:   for  $i = 1$  to  $m$  do
5:      $X_{\text{z\_scored}}[i, j] \leftarrow (X[i, j] - \mu_j) / \sigma_j$    ▷ Z-score normalization formula
6:   end for
7: end for
8: return  $X_{\text{z\_scored}}$ 
    
```

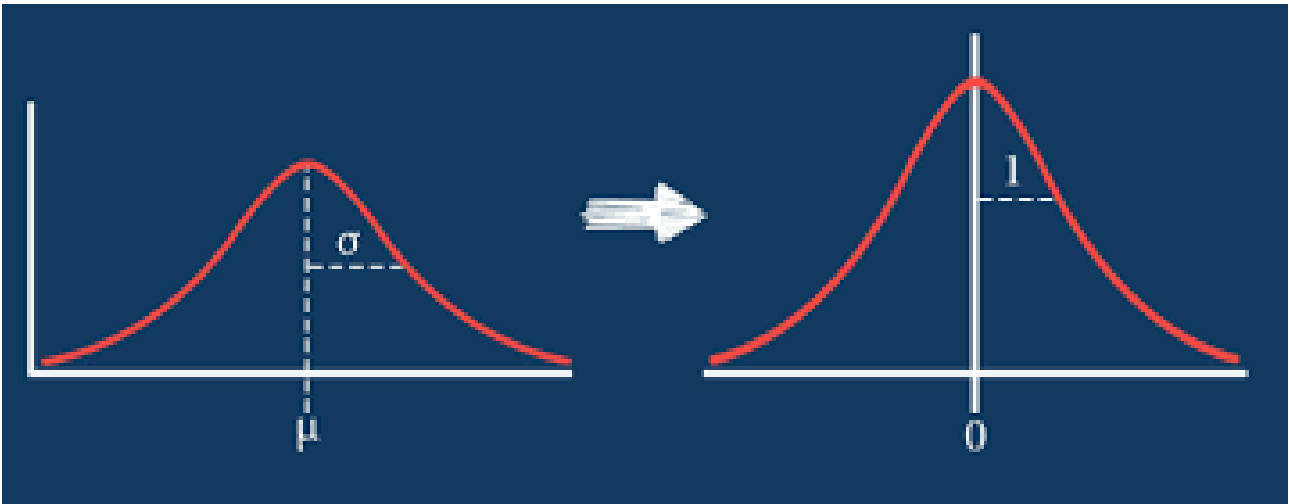


Figure 2.8: Z-score normalization

- **Decimal Scaling:** This technique normalizes the data by adjusting its decimal point. To apply this method, each data value is divided by the maximum absolute value in the dataset [38]. A data value x_i is normalized to x'_i using the following formula 2.11:

$$x_i = \frac{x'_i}{10^n} \quad (2.11)$$

Where:

- x is the original data point.
- n is the smallest integer such that the maximum absolute value of the data is less than or equal to 1.
- **L2 Normalization:** It is called Euclidean normalization, it normalizes vectors or matrices by scaling their elements such that their Euclidean norm (L2 norm) becomes equal to 1 [38]. Here is how we define L2 norm of a vector:
Let's suppose we have a vector $x = [x_1, x_2, \dots, x_n]$, the Euclidean norm is defined as 2.12

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2} \quad (2.12)$$

For the vector x , the L2 normalized vector $x_{norm} = [x_{1norm}, x_{2norm}, \dots, x_{nnorm}]$ is calculated as 2.13

$$x_{norm} = \left[\frac{x_1}{\|x\|}, \frac{x_2}{\|x\|}, \dots, \frac{x_n}{\|x\|} \right] \quad (2.13)$$

- **Square Root Normalization:** Square root normalization, or Sqrt normalization, is a data transformation technique that adjusts data to a more appropriate range for specific machine learning algorithms or data analysis tasks. This method is especially beneficial for handling skewed or heavy-tailed data distributions. The process involves applying the square root to each data point in the dataset [32].
for a vector $X = [x_1, x_2, \dots, x_n]$.

$$x_{sqrt_normalized} = [\sqrt{x_1}, \sqrt{x_2}, \dots, \sqrt{x_n}] \quad (2.14)$$

The square root transformation has several useful benefits. It helps stabilize the variance of data, which is important when the variance tends to increase with the mean, like in count data or Poisson distributions. This makes the data more consistent and easier to work with. It also reduces skewness, making skewed data distributions more symmetric

and closer to a normal distribution. Additionally, it compresses the range of the data, bringing large values closer to smaller ones. This is helpful when dealing with data that has a wide range or outliers.

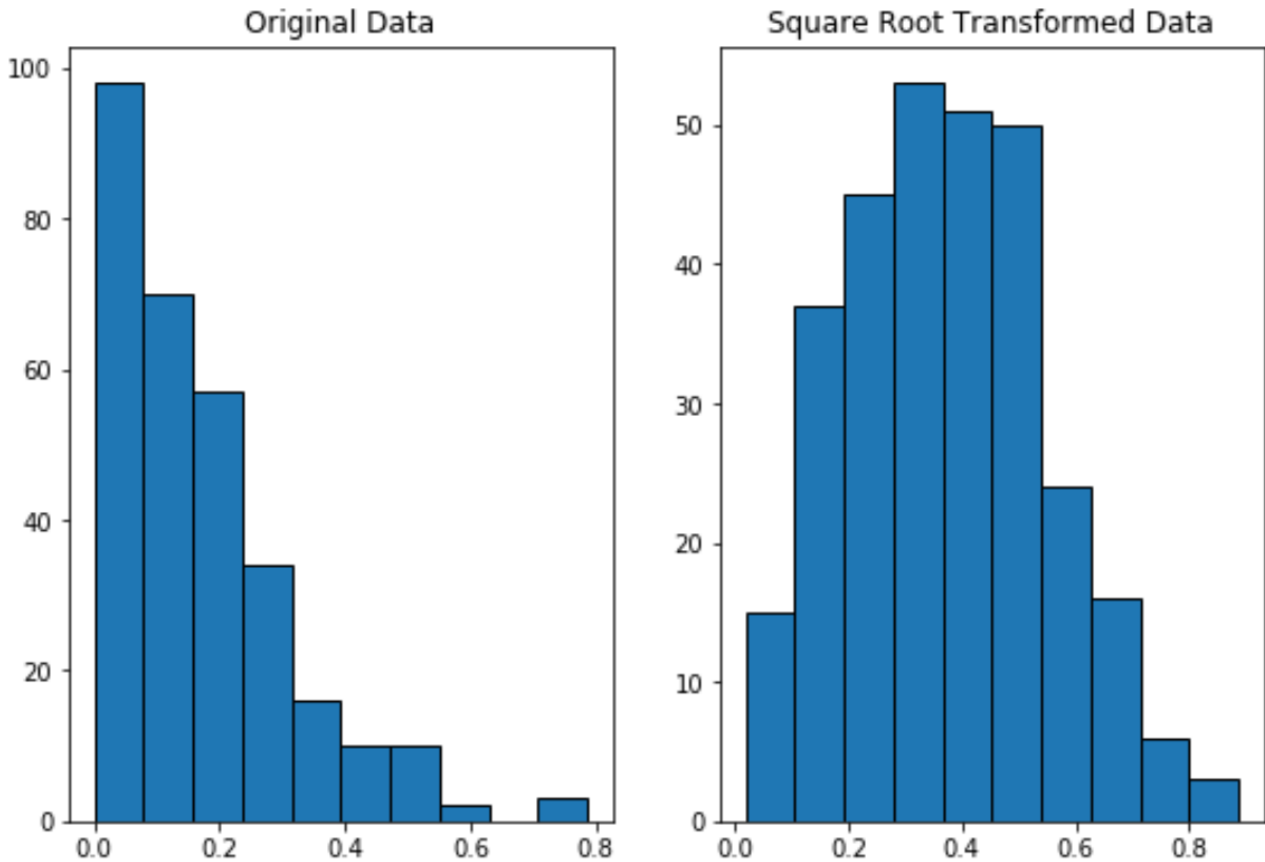


Figure 2.9: Comparison of data before and after being square root transformed [41]

- **Logistic Sigmoid:** It is a non-linear function that maps any real-valued input to a value between 0 and 1. It is defined as 2.15:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.15)$$

Where:

- x is the input value.

- **Pareto Scaling:** It works by adjusting the data so that it centers around the average and is scaled based on the square root of the standard deviation. This approach helps keep the overall pattern of the data intact while minimizing the effect of very large values and outliers [38].

The following describes the algorithm for computing Pareto scaling.

Input: X (dataset with m instances and n features)

Output: $X_{\text{pareto_scaled}}$ (Pareto Scaled dataset)

Algorithm 3 Pareto Scaling Normalization [37]**Require:** X (dataset with m instances and n features)**Ensure:** $X_{\text{pareto_scaled}}$ (Pareto Scaled dataset)

```

1: for  $j = 1$  to  $n$  do
2:    $\mu_j \leftarrow \text{calculateMean}(X[:, j])$  ▷ Calculate mean of feature  $j$ 
3:    $\sigma_j \leftarrow \text{calculateStandardDeviation}(X[:, j])$  ▷ Calculate standard deviation of feature  $j$ 
4:   for  $i = 1$  to  $m$  do
5:      $x_{\text{centered}} \leftarrow X[i, j] - \mu_j$  ▷ Center the data by subtracting the mean
6:     if  $\sigma_j == 0$  then
7:        $X_{\text{pareto\_scaled}}[i, j] \leftarrow 0$  ▷ Handle cases where standard deviation is zero
8:     else
9:        $X_{\text{pareto\_scaled}}[i, j] \leftarrow x_{\text{centered}} / \sqrt{\sigma_j}$  ▷ Scale the centered data by dividing by square root of standard deviation
10:    end if
11:  end for
12: end for
13: return  $X_{\text{pareto\_scaled}}$ 

```

- **Power Transformation:** It is a combination of min-max normalization and square root transformation on the input data. The following details the calculation of the normalization procedure [38]:

1. Input:
 - train: A two-dimensional matrix representing the training data.
2. Initialization:
 - ntrain: Create a matrix of zeros with the same dimensions as train to store the normalized data.
 - tmp_{train} : Create another matrix of zeros with the same dimensions as train for temporary calculations.
 - mn: Find the minimum value for each feature in train and store it in a vector mn.
 - mx: Find the maximum value for each feature in train and store it in a vector mx.
3. Preprocessing for Negative Values: remove the negative values with absolute.
4. Square Root Transformation: Apply a square root transformation to all elements in tmp_{train} .
5. Calculate mean centering for each feature

The resulting normalized feature vectors are then fed into a classification stage for kinship verification evaluation.

2.6 Classification

The last stage is to categorize the individual pairings according to the feature vectors that were taken from the matrices that took from the preprocessing phase. Each pair in this classification procedure is given a label that indicates whether the persons in the pair are related or not.

A binary classifier is employed to accomplish this classification. For this, a number of methods binary classifiers can be used, such as Random Forest, Decision Trees, Multilayer Perceptrons (MLP), Support Vector Machines (SVM), and others. Based on the patterns seen in the data, these classifiers examine the collected feature vectors and learn to differentiate between related and unrelated pairs.

Support Vector Machines (SVMs) have been a popular choice for kinship verification due to their effectiveness on various datasets [47, 46, 30]. Building upon this established approach, this study utilizes the KinFaceW-II dataset for training the SVM classifier.

- **Support Vector Machines SVM**

Support Vector Machines (SVM) is a supervised learning algorithm used in machine learning for classification and regression tasks [31]. The main idea behind SVM is to find the optimal hyperplane that maximally separates the different classes in a dataset. This hyperplane is chosen in such a way that it has the maximum margin, which is the distance between the hyperplane and the nearest data points from each class, known as support vectors. SVMs is about to transform input data into a higher-dimensional feature space using kernel functions, it is about to map the input data points into a higher-dimensional space where the separation between the two classes becomes easier. This allows SVM to solve complex non-linear problems as well, enabling effective classification even for complex, nonlinear data 2.10. SVMs are adept at handling high-dimensional data efficiently and are resistant to overfitting, making them suitable for various classification application.

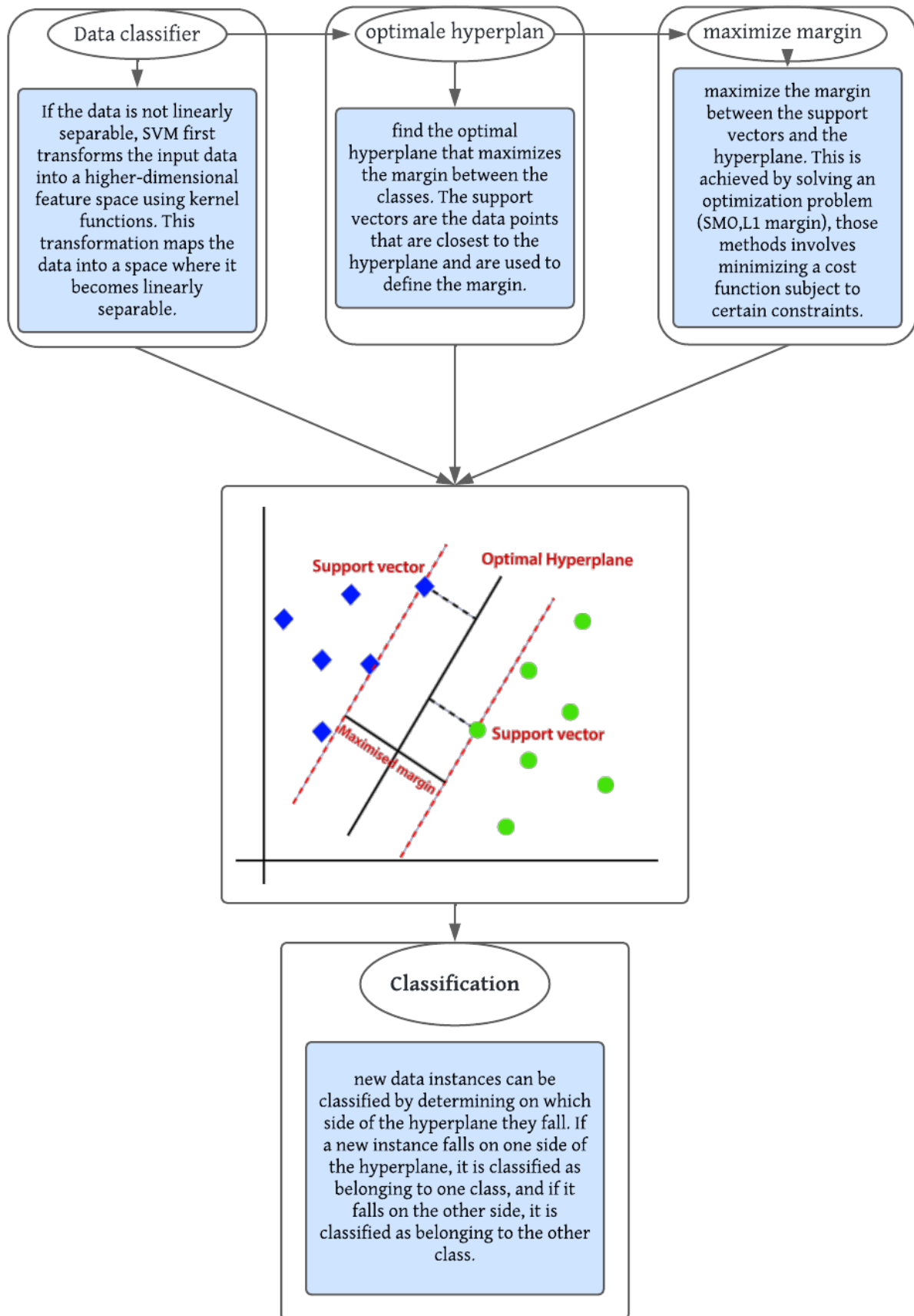


Figure 2.10: Processing steps for SVM classification

2.7 Conclusion

In conclusion, this chapter has provided an overview of the key techniques and innovations in kinship verification systems. It has covered the main steps involved, from face acquisition to preprocessing, feature extraction, normalization, and classification. We have highlighted some of the popular feature descriptors we used in our experiments, which we will discover in the next chapter, as well as important normalization and classification methods.

Chapter 3

Implementation and Results

3.1 Introduction

This chapter explains how we built and tested our kinship verification system. We begin by introducing the dataset chosen to evaluate our proposed model’s performance. Following that, we present the details of the proposed model. We leverage various texture descriptors to extract key features from the images. We employ a Support Vector Machine (SVM) to analyze the extracted features and determine kinship relationships. Furthermore, we present the results of our experiments, focusing on the system’s accuracy and its performance as visualized by ROC curves. Finally we compares the proposed model with state-of-the-art techniques.

3.2 KinFaceW-II Dataset

In our experiment, we tested our classification model using the widely recognized KinFaceW-II dataset [23], which is commonly used for studying kinship verification. The KinFaceW-II dataset is divided into four groups, each containing 250 pairs of color images. These image pairs depict different types of family relationships: Father-Son (FS), Mother-Son (MS), Father-Daughter (FD), and Mother-Daughter (MD). For each kinship relation, the 250 positive pairs (kin) are used to generate 250 negative pairs (not kin). This strategy ensures that both classes (kin and not kin) are equally represented within the data, leading to a total of 500 image pairs for each relation. This involves splitting the balanced data into five equal folds, each containing 100 image pairs (50 kin and 50 not kin).

The table 3.1 illustrates the structure of the KinFaceW-II dataset for each family type and the table 3.2 presents the statistic of this dataset:

Table 3.1: Contributions of different Kinface-II source

Fold(i)	matches	Pairs	
i	Kin/non-kin	image 1	image 2

Table 3.2: The statistic of the KinFaceW-II dataset [44]

Class	FD	FS	MD	MS
Pairs of images	250	250	250	250
Asian people %	6	7.2	7.6	11.2
Not Asian people%	94	92.8	92.4	88.8

To provide a visual representation of the dataset, Figure 3.1 and Figure 3.2 showcase example images pairs.

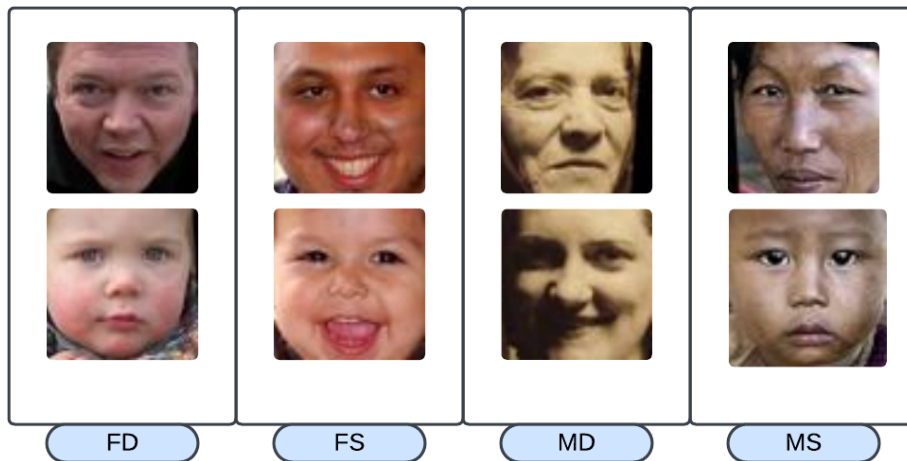


Figure 3.1: Example of images from the KinFaceW-II dataset as kin relationship

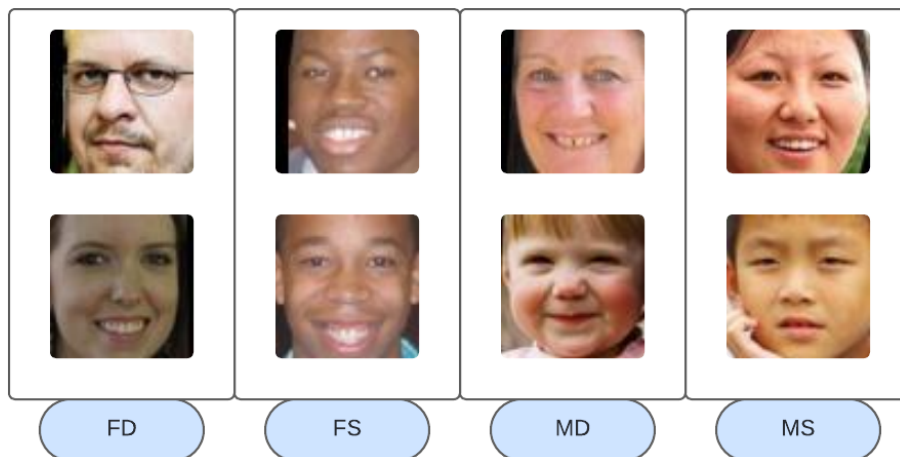


Figure 3.2: Example of images from the KinFaceW-II dataset as Non-kin relationship

3.3 Main process of Kinship Verification System

In the figure below 3.3, we provide an overview of the global system and illustrate how it operates. This will give us a broad understanding of the system as a whole. In the subsequent

sections, we will break down each process, explaining their roles and how they contribute to the overall functionality of the system.

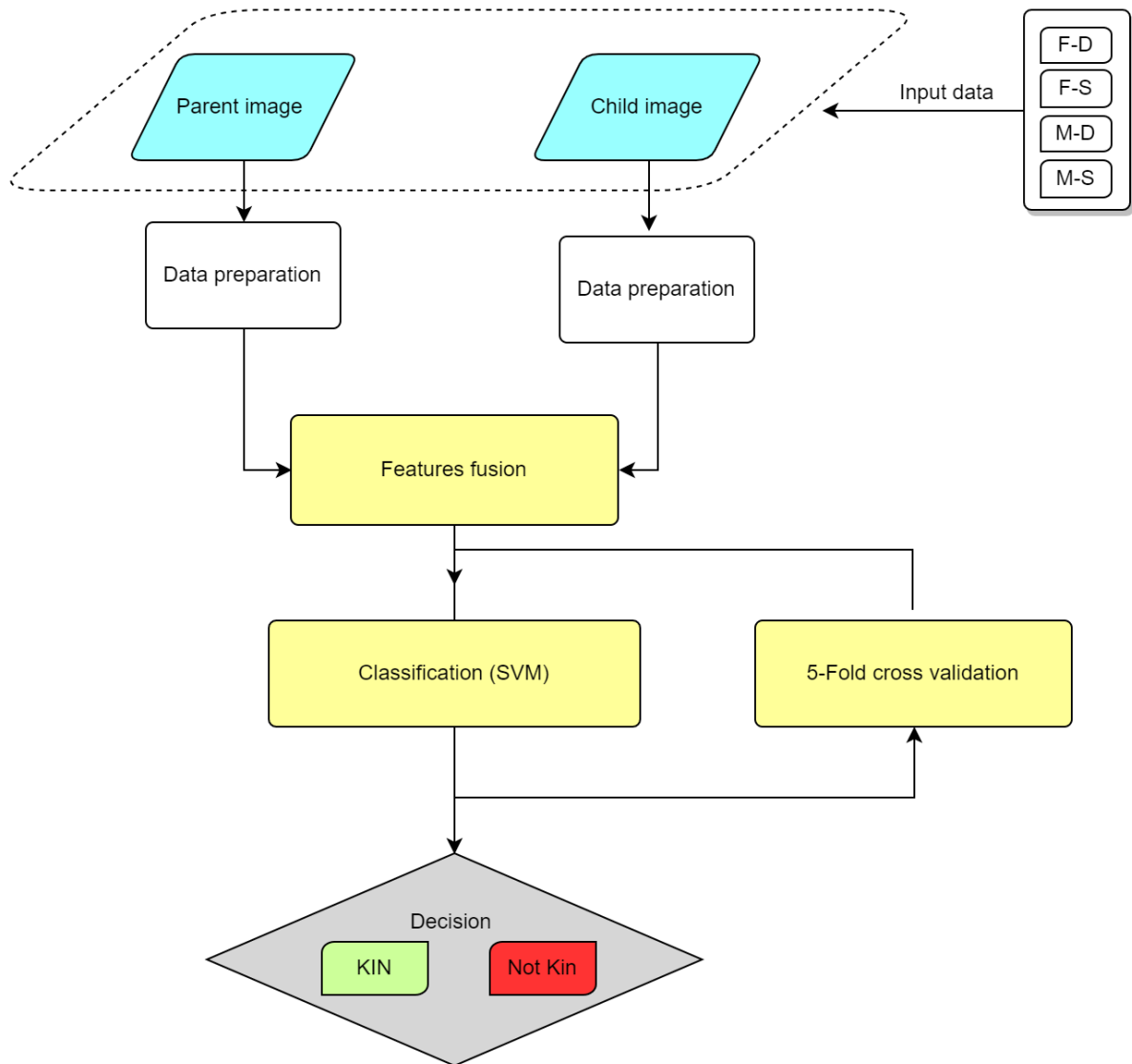


Figure 3.3: Kinship Verification Process Schema

3.3.1 Data Preparation

This thesis delves into the field of kinship verification, aiming to develop a method that automatically determines whether two individuals in images are related. Before feeding the data into a classifier, it undergoes several steps to ensure optimal performance. As illustrated in the figure 3.4, the first step typically involves color space conversion, followed by feature extraction and finally, feature normalization.

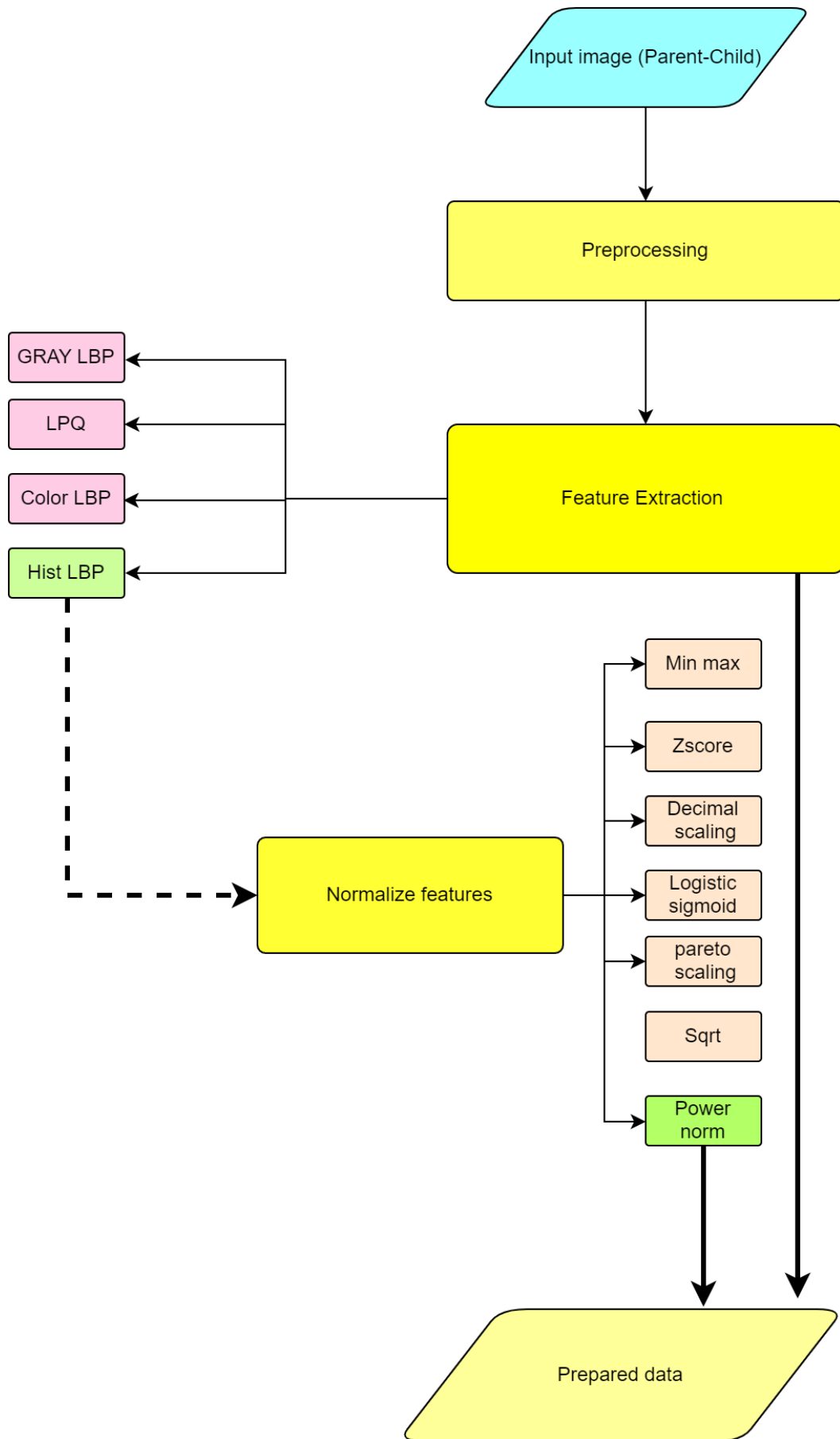


Figure 3.4: Pipeline for feature extraction and data preparation from input images

3.3.1.1 Data Preprocessing

This section explores the concept of color space conversion, focusing on how an RGB image, the workhorse of digital photography, transforms when different conversion techniques are applied. In the figure 3.5 below, we'll look at how an RGB image, a common format for digital images, is transformed when different color conversion techniques are applied.

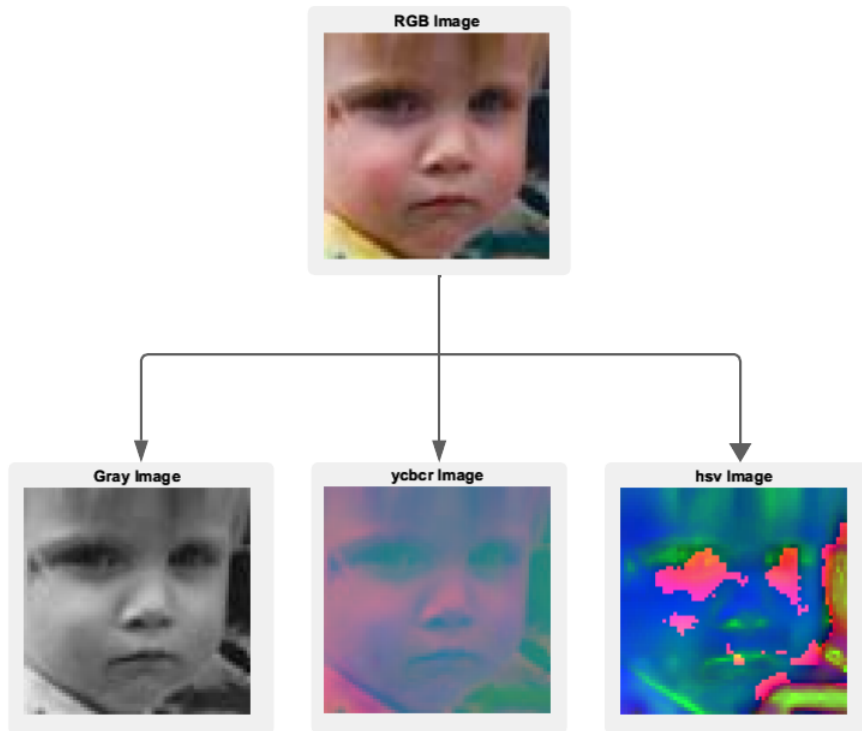


Figure 3.5: RGB image conversion to different color spaces

In this algorithm, we show the logic of data preprocessing before starting the feature extraction.

Algorithm 4 Database Preparation

```

1: procedure DATABASE PREPARATION
2:   for each image  $I$  do
3:     Read images
4:     Convert images to HSV, YCbCr, and Grayscale color spaces
5:   end for
6: end procedure

```

3.3.1.2 Feature Extraction

Following data preprocessing, feature extraction plays a critical role in the kinship verification process. This step aims to identify and extract essential characteristics from the images that can be used to determine potential kinship relationships. This work utilizes various texture descriptors: Local Phase Quantization (LPQ), Local Binary Patterns (LBP) in grayscale, RGB, and HSV color spaces, and Histogram of Local Binary Patterns (Hist LBP).

The figure below 3.4 shows the detailed steps involved in data preparation, as mentioned in

the previous figure 3.3. It breaks down each stage, showing how we transform raw data into a format ready for the system.

- **Local Phase Quantization:** We leverage Local Phase Quantization (LPQ) to extract features from the dataset. This process involves first converting the original RGB images to grayscale format. The LPQ algorithm is then applied to these grayscale images, resulting in a new feature representation for each image. This new representation captures textural information that can be beneficial for classification tasks.
- **Local Binary Patterns:** We employ Local Binary Patterns (LBP), as detailed in section 2.4.4, LBP, to extract features from the dataset. This process begins by converting the original RGB images to grayscale format. The grayscale images are then subjected to LBP analysis, resulting in a new feature representation that encodes the textural details of each face image. The figure 3.6 shows the main steps of an LBP descriptor.

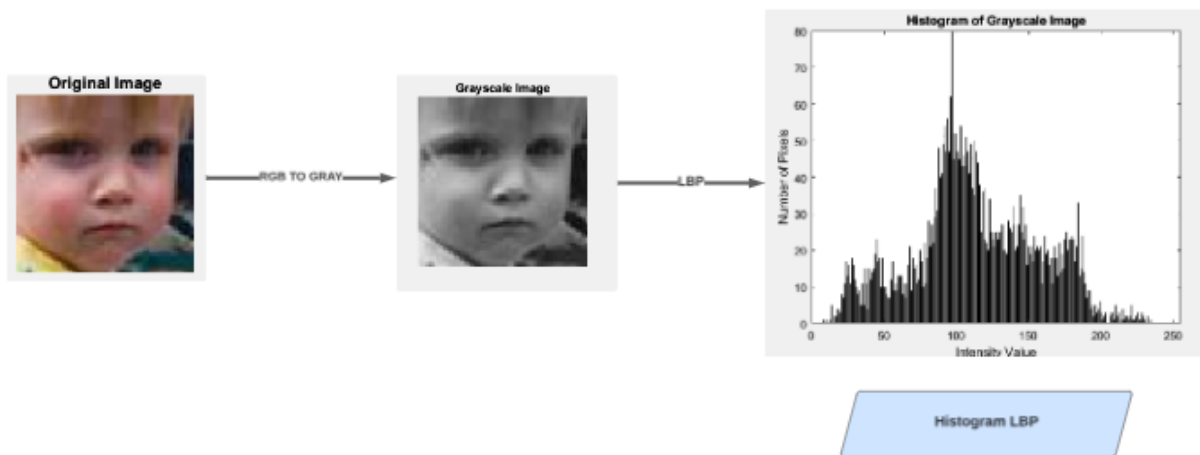


Figure 3.6: Local binary patterns features

- **Color Local Binary Patterns in RGB Color Space:** We utilize Color Local Binary Patterns (Color LBP (RGB)) to capture both color and texture information from the images. Unlike LBP, which operates on grayscale images, Color LBP leverages the inherent color information within the data. To achieve this, we first decompose the original RGB image into its individual red, green, and blue channels, effectively creating three separate grayscale images. Subsequently, we apply LBP to each of these grayscale channels, resulting in three distinct feature vectors. Finally, these three feature vectors are concatenated to form a single, combined feature representation. This final representation incorporates both the spatial textural patterns and the color information from the original RGB image. Figure 3.7 summarizes the calculation of the Color LBP (RGB) features.



Figure 3.7: Color local binary patterns steps

- Color Local Binary Patterns in HSV Color Space:** In addition to applying Color LBP to the RGB color space, we also explore its application in the HSV (Hue-Saturation-Value) color space. This approach aims to potentially capture different aspects of color information compared to the RGB representation. The process begins by converting the original RGB image to the HSV color space. We then separate the HSV image into its individual channels (Hue, Saturation, and Value). Similar to Color LBP in RGB, we apply the LBP descriptor to each of these separate channels, resulting in three distinct feature vectors. Finally, these three feature vectors are concatenated to form a unified feature representation. This approach leverages the potential benefits of the HSV color space representations for feature extraction.
- Histogram of Local Binary Patterns:** The Hist LBP is a technique originally developed for person re-identification. Hist LBP aims to capture a more nuanced and informative representation of facial features by combining texture and color information into a single descriptor. This work aim to demonstrate the effectiveness of Hist LBP in improving the accuracy of kinship verification compared to traditional methods.

Hist LBP is a combination between HOG and LBP [2.4.2](#), [2.4.4](#), for the purpose to improve performances, before we go to algorithm, here is definition of some variables:

- Patches:** Patches determine the size of the local regions or blocks from which features will be extracted.
- Step size:** It is the distance between the center of neighboring patches when dividing the image, it is used to determine the level of overlap between patches. See the figure below [3.8](#) :

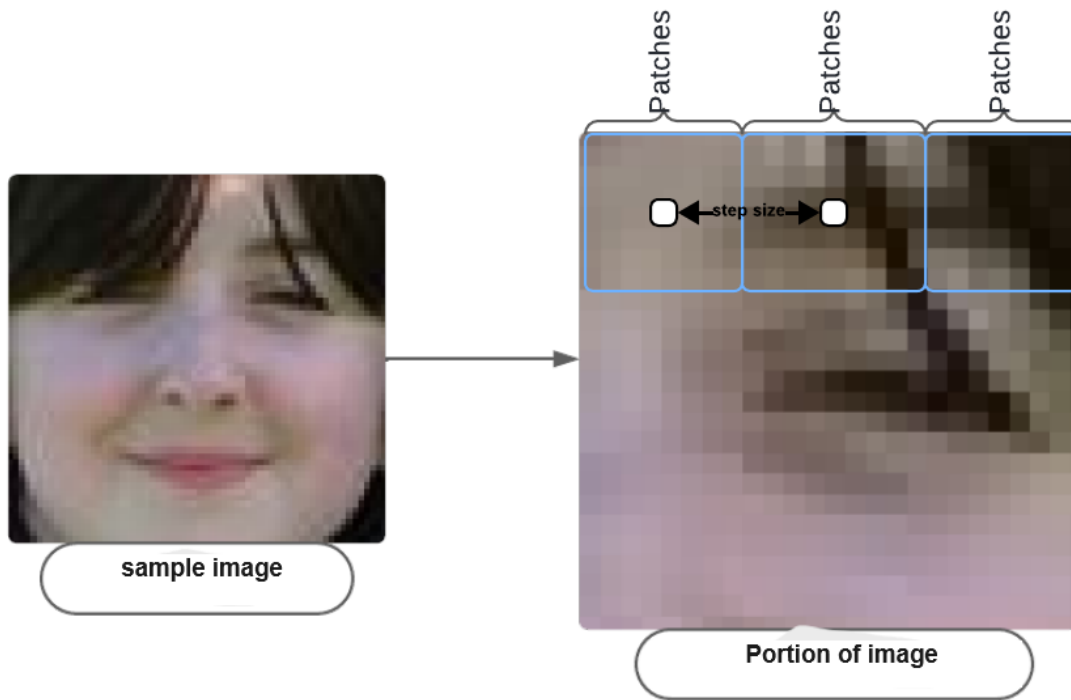


Figure 3.8: Difference between patches and step size

- **Bins:** They are the range of pixels values in the histogram,

The algorithm 5 shows Hist LBP explained after data preprocessing.

Algorithm 5 Feature Extraction with Hist LBP

```

1: procedure FEATURE EXTRACTION( $I$ , patchSize, numBins, stepSize)
2:   imsz  $\leftarrow$  size( $I$ ) ▷ Get image size
3:   patches  $\leftarrow$  GeneratePatches(imsz, patchSize, stepSize) ▷ Generate patches
4:   for patch  $\in$  patches do
5:     for  $c \in \{R, G, B, H, S, V, Y, Cb, Cr\}$  do ▷ For each color channel
6:       hist  $\leftarrow$  ComputeHistogram( $I_c$ (patch), numBins) ▷ Compute histogram
7:       hist  $\leftarrow$  hist/sum(hist) ▷ Normalize histogram
8:       colorHist[patch,  $c$ ]  $\leftarrow$  hist ▷ Store color histogram
9:     end for
10:    convert  $I$  to grayscale
11:    lbpFeat  $\leftarrow$  LBP( $I$ (patch), 1, patch) ▷ LBP
12:  end for
13:  features  $\leftarrow$  Concatenate(colorHist, lbpFeat) ▷ Concatenate color histogram and LBP features
14: end procedure

```

3.3.1.3 Feature Normalization

Each extracted feature vector for the parent image and the child image extracted from the Hist LBP descriptor is separately subjected to normalization using different normalization methods

(section 2.5).

For each descriptor, the resulting feature vectors for the parent and child are then merged to create a single feature vector representing the pair.

3.3.2 Feature Fusion

The feature fusion technique calculates a new feature vector to represent a pair of individuals (parent and child). This is achieved for each feature dimension by computing the absolute difference between the corresponding feature values of the parent and child. In simpler terms, for each feature, we take the absolute value of the subtraction between the parent's and child's values for that feature.

The merged feature vector is subsequently utilized as input for the classification stage.

3.3.3 Classification

In our experiment, the prepared data is split into **80% for the training phase and 20% for the testing phase**. This split is utilized for cross-validation. The training data is then fed into a linear SVM classifier because it is the best choice in this field to generate a trained model. Subsequently, the system uses this model to predict outcomes for the remaining 20% of the data reserved for testing. The predicted results are compared with the actual test data to calculate the accuracy of each k-fold ($k=1,2,3,4,5$), and the mean accuracy across the 5 folds is determined. This process is repeated for each type of relationship, resulting in four distinct models: **model-FD**, **model-FS**, **model-MD**, **model-MS**. the figure 3.9 below resume the training process for each model.

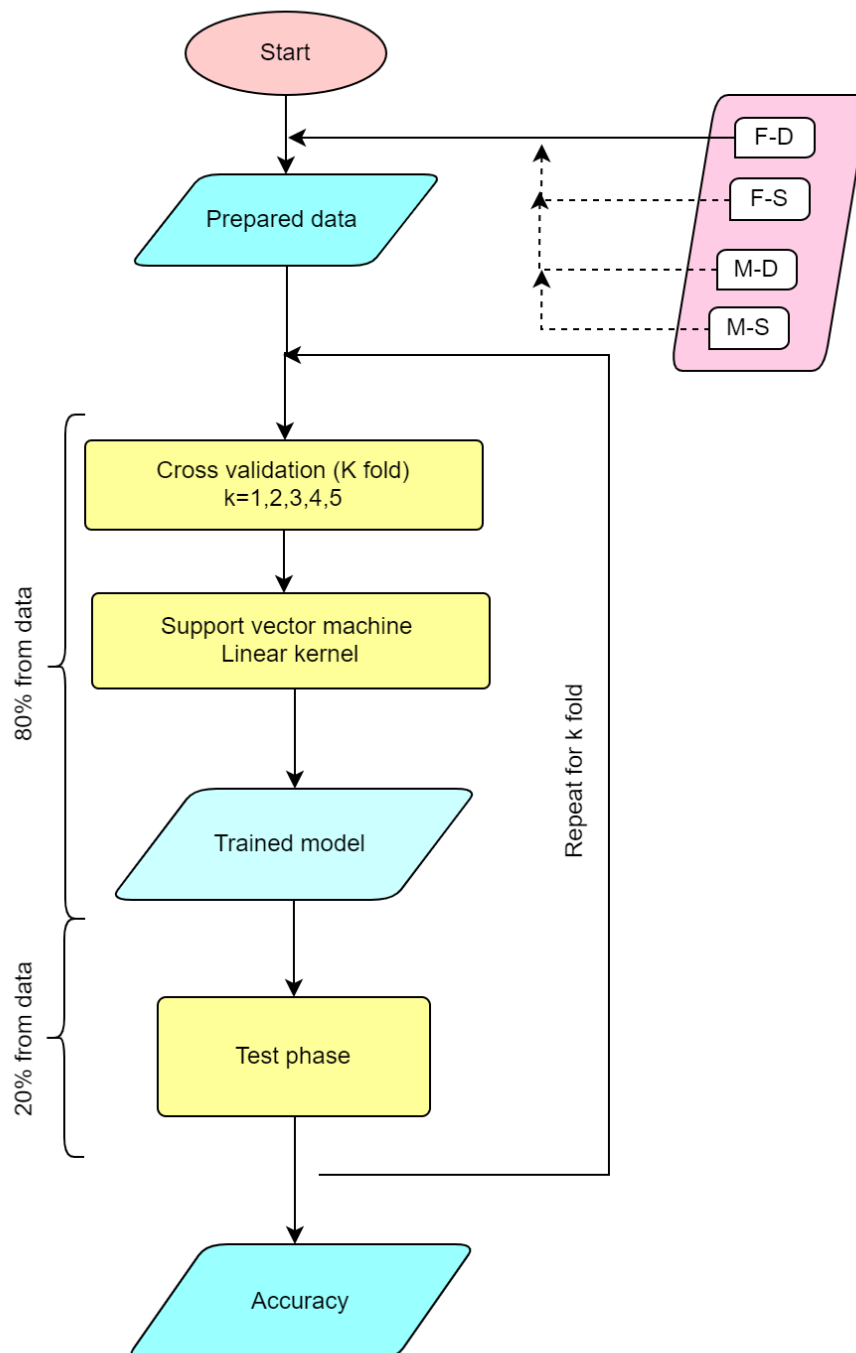


Figure 3.9: The training process for the system to get four models, model-FD, model-FS, model-MD, model-MS

We employ a 5-fold cross-validation technique to train and evaluate the SVM classifier. This process helps ensure the model's generalizability and reduces the risk of overfitting to the specific training data.

Cross-Validation

Cross-validation is a resampling technique used to evaluate our model on an unseen dataset and estimate their performance on new, unseen data. It is a way to assess how well a model

will generalize to new data, and it helps to reduce overfitting, which occurs when a model fits the training data too closely and fails to generalize well to new, unseen data [3].

The basic idea of cross-validation is to divide the available dataset into two parts: a training set and a validation set (or test set), as presented in the figure 3.10. The model is trained on the training set, and its performance is evaluated on the validation set. This process is repeated multiple times, with different portions of the data used as the training and validation sets each time. The final performance measure is typically the average of the performance measures across all iterations.

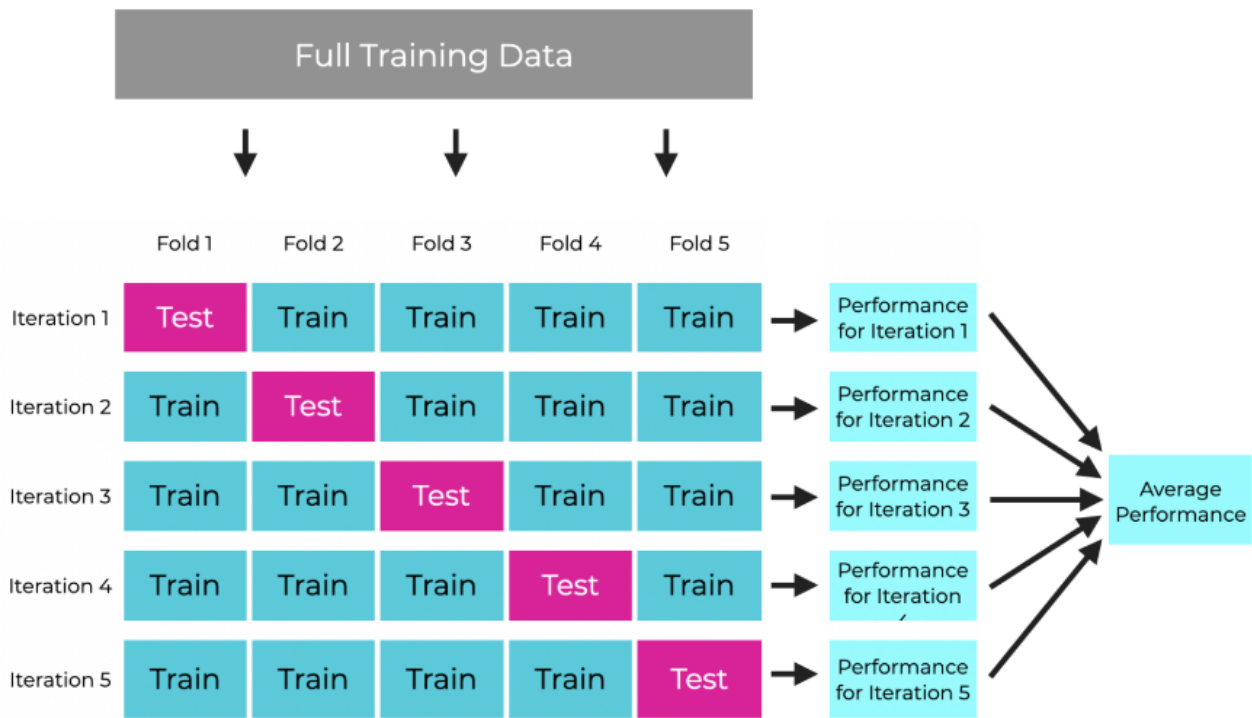


Figure 3.10: 5-fold cross validation explained [42]

We assess the performance of the SVM classifier using various evaluation metrics.

3.4 Evaluation Metrics

To comprehensively evaluate the performance of our kinship verification model, we employed three key metrics: Accuracy, Receiver Operating Characteristic (ROC) curve analysis, and Area Under the ROC Curve (AUC).

3.4.1 Accuracy

Accuracy in machine learning is a metric that measures the proportion of correct predictions made by a model out of the total number of predictions. It is one of the most widely used evaluation metrics for image classification tasks, where the goal is to assign the correct class

label (or category) to each instance in the dataset. It is calculated as below :

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad (3.1)$$

Or in another term by using **confusion matrix** where it summarizes the performance of a classification model by comparing the predicted class labels with the true class labels (ground truth) for a set of instances.

Confusion matrix		<i>Reality</i>	
		Negative : 0	Positive : 1
<i>Prediction</i>	Negative : 0	True Negative : TN	False Negative : FN
	Positive : 1	False Positive : FP	True Positive : TP

Figure 3.11: Confusion matrix for machine learning

And with that we define accuracy as :

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (3.2)$$

Accuracy is often expressed as a percentage, ranging from 0% (all predictions are incorrect) to 100% (all predictions are correct). The importance of accuracy in machine learning lies in its ability to provide a simple and intuitive measure of a model's overall performance. It is easy to interpret and understand the results of our work. By offering a clear indication of how well a model is performing in making correct predictions, accuracy serves as a fundamental benchmark for evaluating and comparing different models. Its straightforward nature allows for quick assessments and facilitates communication of results to stakeholders who may not have a deep technical background. However, while accuracy is a fundamental metric for evaluating model performance, ROC (Receiver Operating Characteristic) curves offer a more comprehensive view of a classifier's performance.

3.4.2 Receiver Operating Characteristic

ROC curves (Receiver Operating Characteristic curves) are a crucial tool for assessing the performance of machine learning models, They provide an effective means to see how any predictive model can distinguish between the true positives and negatives, particularly in binary classification tasks [1], where there are two distinct output classes, in our case we have **Kin** or **No kin**. An ROC curve illustrates the relationship between the true positive rate (TPR) and the false positive rate (FPR). The TPR indicates the proportion of actual positives correctly identified by the model, while the FPR represents the proportion of actual negatives incorrectly classified as positives. An ideal classifier achieves a TPR of 1 and an FPR of 0.

3.4.3 Area Under the Curve

In the context of a ROC curve, the AUC represents the area under the ROC curve. The AUC's value ranges from 0 to 1, with the following interpretations:

- AUC = 0.5: The classifier is no better than random guessing.
- AUC < 0.5: The classifier is worse than random guessing.
- $0.5 < \text{AUC} < 1$: The classifier is better than random guessing, with higher values indicating better performance.
- AUC = 1: The classifier is perfect, with no false positives or false negatives.

3.5 Experimental Protocol for Classification

In this section, we've outlined the classification system we created to verify kinship, or family relationships. This system is carefully designed to be thorough and adaptable, allowing it to handle different types of data and scenarios. By using this system in our tests, we were able to develop a reliable method for confirming whether people are related, based on features like facial similarities or genetic information.

3.5.1 Evaluation of Different Descriptors

We used accuracy 3.4.1 as a primary tool to evaluate the performance of the classification model because it provides a straightforward and intuitive measure of how well the model is performing overall. Accuracy is easy to understand and interpret, making it a convenient metric for quickly assessing the effectiveness of the model. This section presents the evaluation results for various feature extraction techniques employed in our kinship verification system. These techniques include Local Phase Quantization (LPQ), Local Binary Patterns (LBP) in grayscale, RGB, and HSV color spaces, and Histogram of Local Binary Patterns (Hist LBP). The evaluation focuses on the performance of each technique in terms of accuracy. The results presented in the following table demonstrate that the Histogram of Local Binary Patterns (Hist LBP) descriptor achieved the most favorable results compared to the other descriptors evaluated in this study.

Table 3.3: Evaluation of different feature descriptors based on accuracy

	F-D%	F-S%	M-D%	M-S%	Mean accuracy%
LBP gray	73	69	69	68	70
Color LBP RGB	72	72	71	70	71
Color LBP HSV	79	80	82	84	81
LPQ	58	64.8	54.8	55.2	58.2
Hist LBP	84	88	90	92	88.5

To further illustrate the effectiveness of Hist LBP, Figure 3.12 presents a comparison of its performance against other feature descriptors. This visual representation allows for a clearer understanding of how Hist LBP compares to other descriptor.

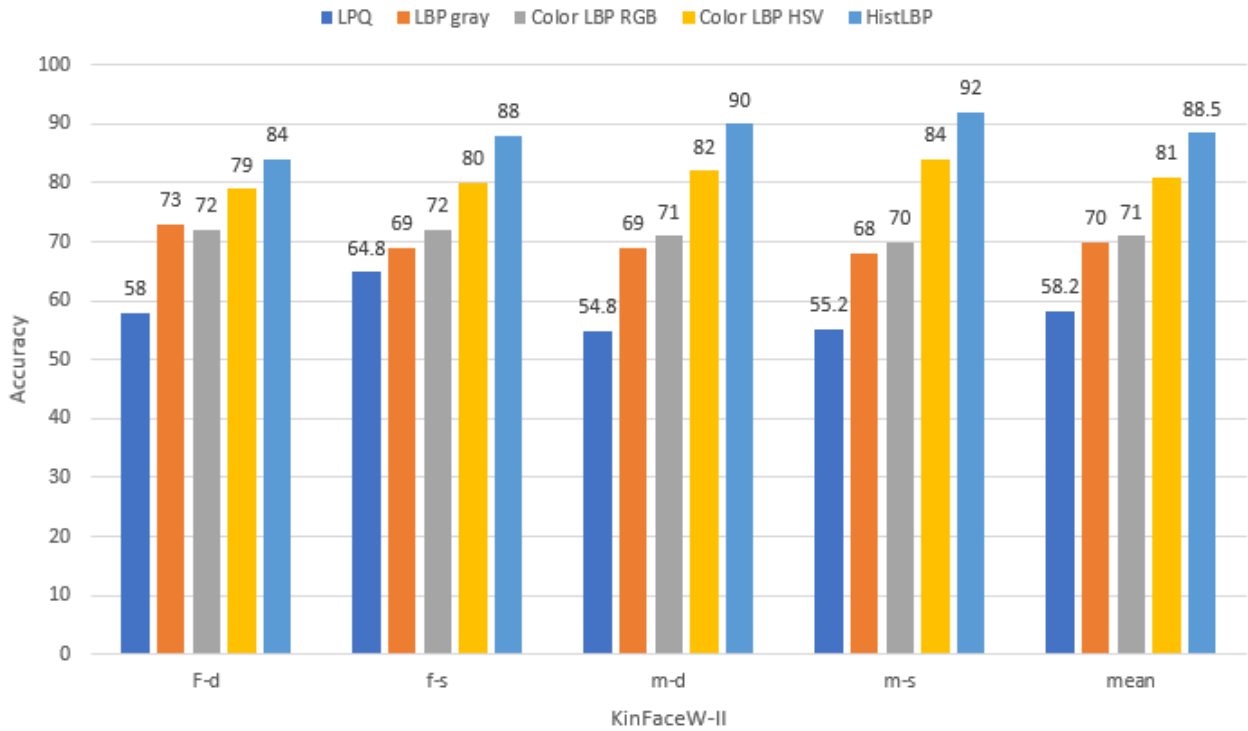


Figure 3.12: Accuracy estimation using SVM with different feature extractors

As shown in the 3.12, Hist LBP is the feature extractor that provides the best accuracy with a default configuration (nbins=16, Paches=32, step size=8). Based on this, we focus on it further and enhance its development.

3.5.2 Impact of Different Parameters on the Hist LBP Descriptor

To identify the optimal configuration for the Hist LBP extractor, we conducted a grid search experiment. This experiment systematically evaluated the impact of different parameter values on the extractor’s performance. Specifically, we focused on tuning the parameters related to: Patches and bins.

The grid search explored various combinations of these parameter values. The best performing combination, as determined by the accuracy achieved using 5-fold cross-validation (described in section 3.3.3), was then used for further experiments. Table 3.4 summarizes the accuracies obtained with different parameter settings while maintaining a step size of 16 and exploring patch sizes specifically set to 64.

Table 3.4: Accuracy results using Hist LBP descriptor with change of patches and bins

bins	patches	F-D %	F-S %	M-D %	M-S %	Mean accuracy %
8	64	83.8	85.6	91.2	92	88.15
16	64	84.4	85.8	90.6	94	88.7
32	64	84.8	86.4	92.2	88.8	88.05
64	64	84.6	86.4	91.6	91	88.45

Table 3.4 reveals that variations in the number of bins have a minimal effect on accuracy. Based

on this observation, we opted to proceed with a fixed bin value of 16.

We will now focus on adjusting the patch parameters to assess their influence on performance. Notably, the step size will be set equal to the patch size, eliminating any overlap between extracted patches. Following this evaluation, we will investigate the impact of introducing overlap between patches. We will analyze how incorporating overlap affects the results and potentially identify a further improvement in accuracy. The table 3.5 presents the obtained results.

Table 3.5: Accuracy results with step size variation

patches	step size	F-D%	F-S%	M-D%	M-S %	Mean accuracy%
32	32	79.8	86.2	90.4	87.4	85.95
32	16	83	88	90.2	87.8	87.25
32	8	83.8	87.6	91.2	87.2	87.45
32	4	83.8	87.8	91.4	87.2	87.55
32	2	83.2	88.8	91.4	92	88.85

When we reduce the step size in data analysis, we improve accuracy because we take more frequent samples. This increased frequency means that the data segments overlap more, so we capture more detailed information and lose less data. As a result, important patterns are less likely to be missed, making our analysis more precise and reliable. In simpler terms, smaller steps mean we get a clearer and more data.

Prior to plotting the ROC curve, we calculated the Area Under the Curve (AUC) metric. The results of this calculation are presented below.

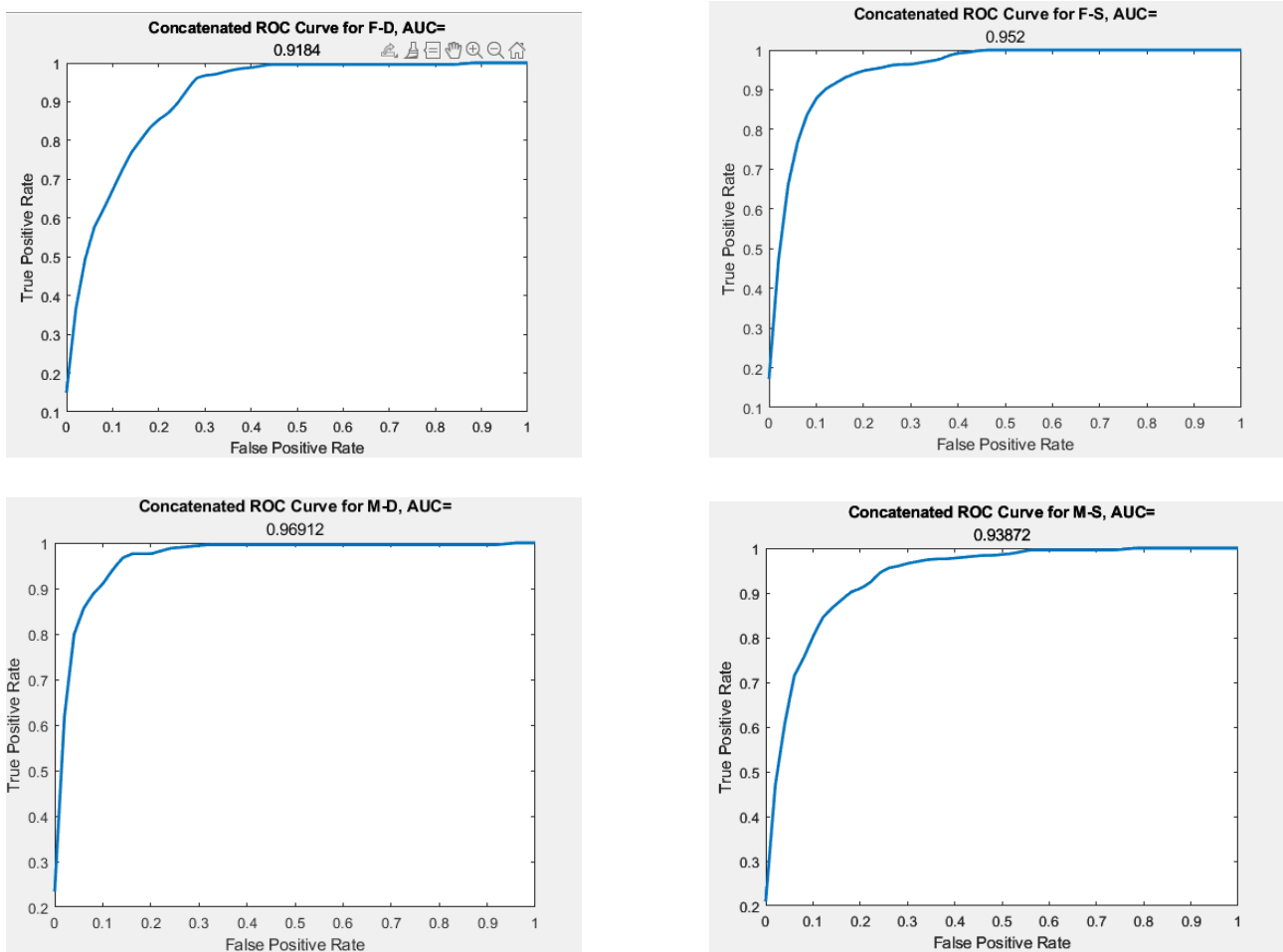


Figure 3.13: ROC curves with Hist LBP for **patches=32, bins=16, step size=2**

From the ROC curves, we can see that the AUC is above 0.5, which means our SVM classifier is performing well, and our parameters are on point. To make our system even better, we should consider adding normalization (refer to section 2.5). This technique can improve the classifier by making sure all features are on the same scale, which helps the model perform more accurately. By normalizing our data, we can achieve even better and more reliable results, making our classification model stronger and more effective.

3.5.3 Impact of Normalization on the Hist LBP Descriptor

As discussed in section 2.5, normalization plays a crucial role in enhancing the performance of machine learning models. To understand its effect more completely, we experimented with different types of normalization techniques. The goal was to observe whether and how these normalization methods influence the model's accuracy and to look for the best classification model. The table 3.6 below presents the results of these experiments while step size=8:

Table 3.6: Mean accuracy after different normalization techniques, step size=8

normalization	Patches	Bins	F-D %	F-S %	Md %	M-S %	mean accuracy %
min max	16	8	82.2	87	91.4	93	88.4
	32	8	82.8	85.4	91.2	92	87.85
	32	64	85.6	87.6	92.6	94	89.95
zscore function	16	8	81	85.8	90.4	89	86.55
	32	8	81.4	85.4	91	91	87.2
	32	64	85	86.8	92	95	89.7
decimal scaling	16	8	82	86.6	91.6	93	88.35
	32	8	80.6	87.2	87.8	91	86.65
	32	64	82.4	87.4	91.2	94	88.75
sqrt	16	8	84.4	89.2	93.6	96	90.8
	32	8	85.4	88.4	93.8	96	90.9
	32	64	86.2	89	93.2	96	91.1
logistic sigmoid	16	8	83	87.2	92	91	88.3
	32	8	83.8	86.2	91.2	90	87.8
	32	64	86	86.4	92.8	93	89.55
pareto scaling	16	8	82	87.2	92.6	95	89.2
	32	8	83.6	85.8	92	93	88.6
	32	64	84.2	85.8	92.6	94	89.15
power norm	16	8	84.4	89.2	93.6	96	90.8
	32	8	85.4	88.4	93.8	96	90.9
	32	64	86.2	89	93.2	96	91.1

The findings presented in the table highlight the significant influence of normalization techniques on the accuracy of our kinship verification models. Notably, most normalization methods, except for Pareto scaling, achieve favorable results when using a patch size of 32. This suggests that patch size plays a crucial role in performance, while the impact of bin values appears to be less pronounced.

Based on these insights, we have chosen to focus further investigation on the power normalization method (refer to section 2.5 for details).

3.5.4 Impact of Power Normalization on the Hist LBP Descriptor

We will explore different parameter settings for power normalization while maintaining a bin value of 16, as suggested by the results presented in the table 3.5. Our objective is to identify the configuration that maximizes model accuracy. The following table presents the mean accuracy results obtained when applying power normalization with varying step size parameters.

Table 3.7: Mean accuracy results from power normalization by adjusting step size parameter

patches	step size	F-D %	F-S %	M-D %	M-S %	mean accuracy %
32	1	84.6	89.4	93.8	95	90.9
32	2	84.6	90.8	94	97	91.6
32	3	83.6	89.6	94	96	90.8
32	4	84.2	90.40	93.4	96	91
32	6	84.6	90.4	93.2	96	91.05
32	8	84.6	90.2	93.6	96	91.2

By analyzing the data presented in the table 3.7, we can evaluate the influence of the step size parameter on the performance of the kinship verification model. Notably, the table reveals that a step size of 2 with a patch of 32 and number of bins of 16 yields the most favorable outcome among the tested values.

To further assess the performance of our classification model equipped with the optimal Hist LBP configuration, we will generate Receiver Operating Characteristic ROC curves. This visual representation allows us to analyze how variations in the power normalization parameters affect the model's ability to discriminate between kinship classes (kin vs. non-kin). By examining the ROC curves, we can gain valuable insights into the model's effectiveness in separating the positive and negative classes.

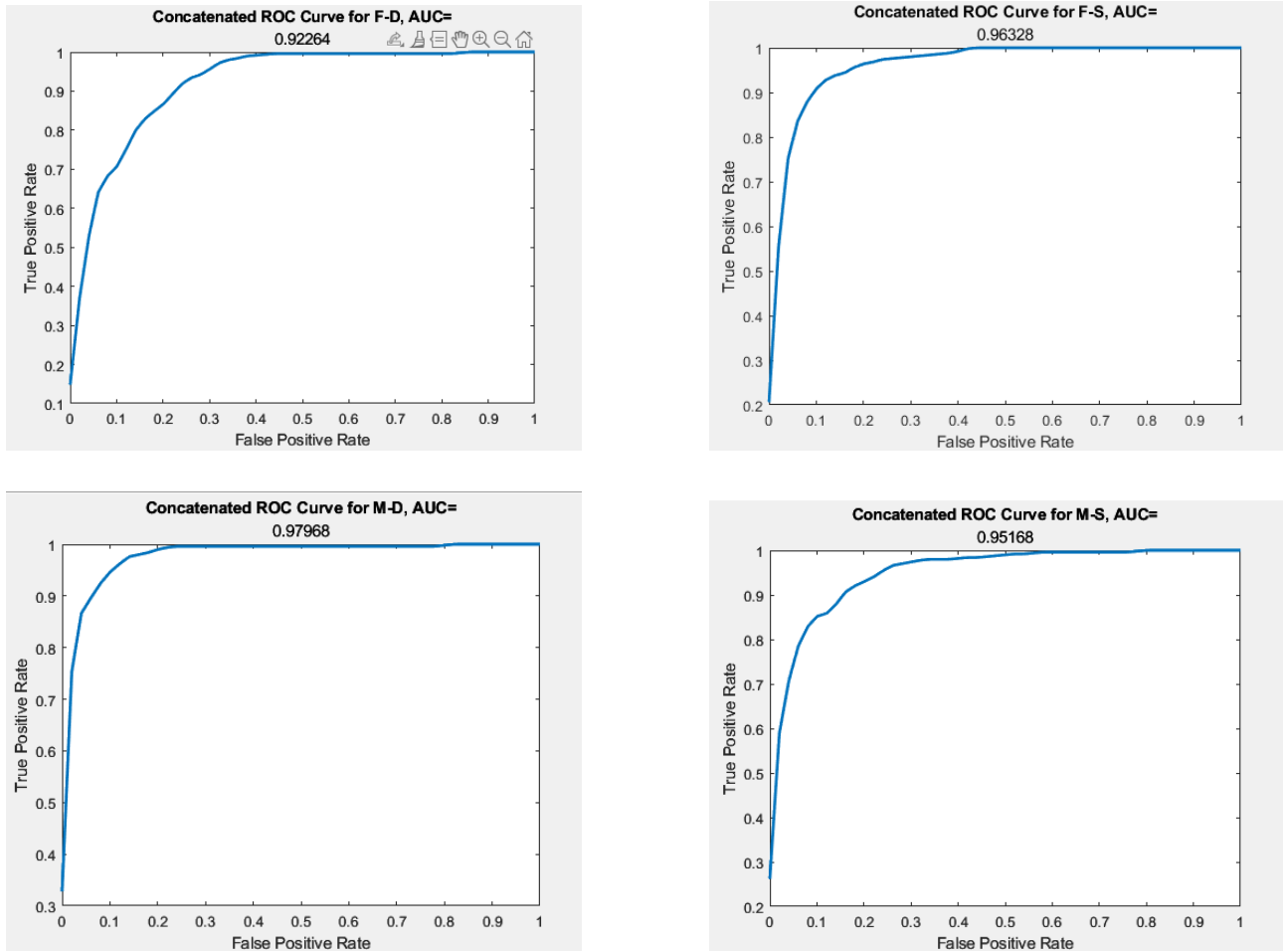


Figure 3.14: ROC curves for **patches=32**, **bins=16**, **step size=2**, with power norm normalization

3.5.5 Impact of Power Normalization on Best Hist LBP Configuration

This section explores the influence of power normalization on the performance of the best Hist LBP configuration. We present the results for both accuracy (table 3.8) and Area Under the ROC Curve (AUC) (table 3.9).

Technique	F-D %	F-S %	M-D %	M-S %	Mean Accuracy %
without normalization	83.2	88.8	91.4	92	88.85
with power normalization,	84.6	90.8	94	97	91.6

Table 3.8: Comparison of Hist LBP Accuracy: Power Normalization vs. No Normalization, **patches=32**, **bins=16**, **step size=2**

Technique	F-D %	F-S %	M-D %	M-S %	Mean AUC %
without normalization	91.65	95.28	96.912	93.872	94.45
with power normalization,	92.26	96.33	97.97	95.17	95.43

Table 3.9: Comparison of Hist LBP AUC: Power Normalization vs. No Normalization, **patches=32, bins=16, step size=2**

The analysis reveals that applying power normalization, compared to no normalization, leads to an improvement in AUC values and accuracy. While the specific increase is ($\sim 1\%$ for AUC and $\sim 3\%$ for accuracy), this enhancement is considered significant for our kinship verification system.

This observation suggests that scaling the data using power normalization before training the SVM classifier likely contributes to improved performance in correctly classifying data instances.

3.6 Comparison with State-of-the-Art Techniques

The optimal configuration identified in the preceding section is evaluated in comparison to the state-of-the-art results for the KinFaceW-II database. As presented in the following table 3.10, our model achieves superior results on KinFaceW-II dataset, surpassing all previously reported methods.

Authors	Classifier	Feature	FD %	FS %	MS %	MD %	Mean Accuracy %
Lu & al [22]	SVM	HOG	71.6	79.6	73.3	69.6	73.5
Faraki & Al [10]	-	LBP	78.4	75.6	68.6	73.2	74
Lui & Zhu [29]	SVM	LBP	-	-	-	-	74.7
Lian & Zhou [20]	-	HOG	71.2	81.4	74.8	73	75.1
Qin & al [19]	-	-	71.6	77.2	79	73.4	75.3
Lu & al [34]	SVM	-	-	-	-	-	75.4
Yan & al [24]	SVM	LBP + TPLBP + SIFT + LE	74.3	76.9	77.4	77.6	76.5
Duan & Tan [52]	SVM	LBP + SPLE + SIFT	74.7	77.3	77.8	78	77
Yan & al [7]	-	LBP	76.2	82.4	76.6	73.2	77.1
H. Yan [51]	SVM	LBP + SPLE + SIFT	76.5	78.5	78.5	79.5	78.3
Hu & al [50]	SVM	LE	76.1	79.8	79.8	80	78.7
J. Hu & al [15]	-	LBP + TPLBP + SIFT + LE	78.2	82.4	78.8	80.4	80
Fang & al [9]	-	HOG	77	85	80.4	78.4	80.2
Tan & al [14]	-	LBP + TPLBP + SIFT + LE	79.8	80.4	78.8	81.8	80.2
Lu & al [26]	SVM	-	73.8	82.6	82.8	84	80.8
Liu & al [21]	SVM	SIFT	77	85.4	81.6	81.6	81.4
Fang & Houang [44]	SVM	LBP	82	87	71	87	81.8
Our approach	SVM	Hist LBP with power normalization	84.6	90.8	94	97	91.6

Table 3.10: Comparison of Hist LBP descriptor with State-of-the-Art Techniques

3.7 Application: Graphical user interface

To ensure the proposed kinship verification model is intuitive and easy to use, we developed a logic that helped us to realize a graphical interface.

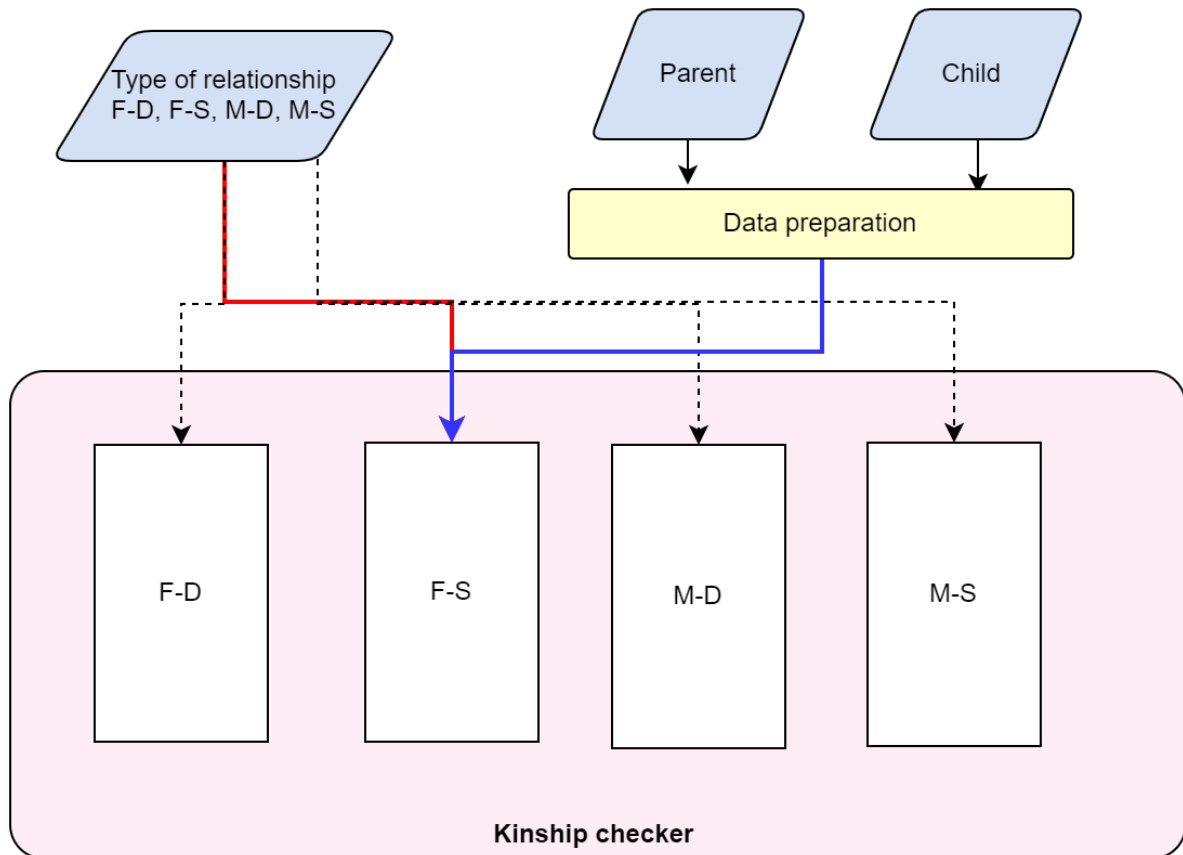


Figure 3.15: Kinship Verification Framework

The figure below 3.16, shows the first interface that the user find when he wants to use it.

Our kinship verification system utilizes a graphical interface developed in MATLAB to guide the user through the process. Here's a step-by-step explanation:

- **Initial Interface** (Figure 3.16): The user encounters this first screen upon launching the application. It likely displays essential instructions or system information.
- **Selecting Parent Image** (Figure 3.17): When the user clicks the "Parent Image" button (Figure 3.16), a file selection window appears (Figure 3.17a) allowing them to choose the parent's image.
- **Parent Image Display** (Figure 3.17b): Once selected, the chosen parent's image is displayed on the main interface.
- **Selecting Child Image**(Figure 3.16): Following the same principle, clicking the "Child Image" button (Figure 3.16) prompts a file selection window (Figure 3.18a) for choosing the child's image.

- **Child Image Display** (Figure 3.18b): The selected child's image is then displayed alongside the child's image on the main interface.
- **Relationship Selection** (Figure 3.19): The user chooses the relationship type they want to verify (likely kin or not kin) through an appropriate interface element as shown in Figure 3.19.
- **Verification Result:** After processing the images and the selected relationship type, the system displays the final result. Figure 3.20 might illustrate a successful verification (images are likely kin), while Figure 3.21 could represent an unsuccessful verification (images are likely not kin).

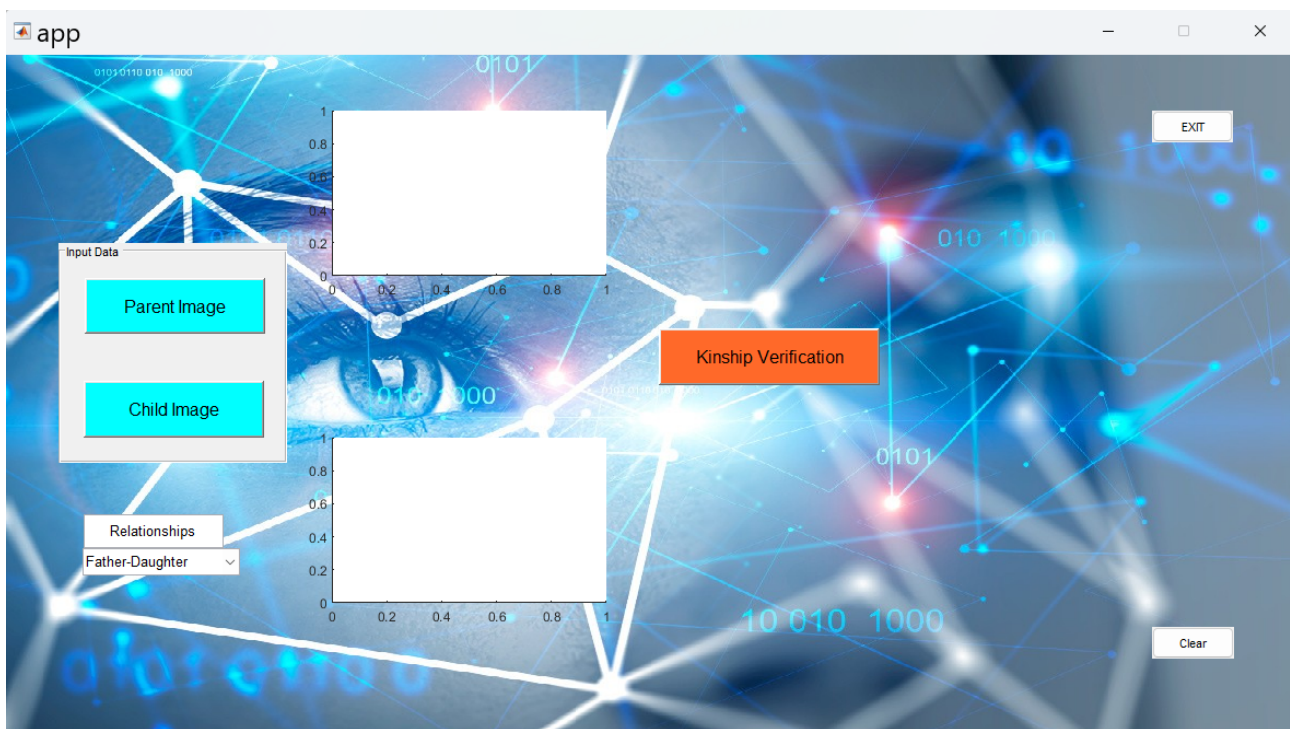
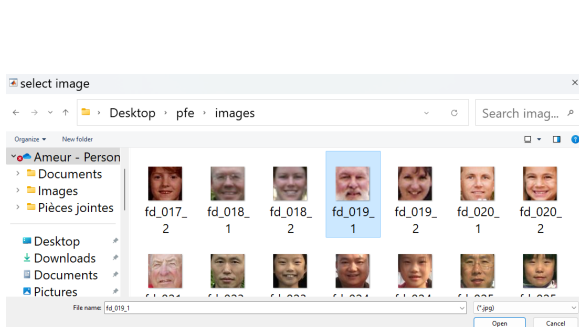
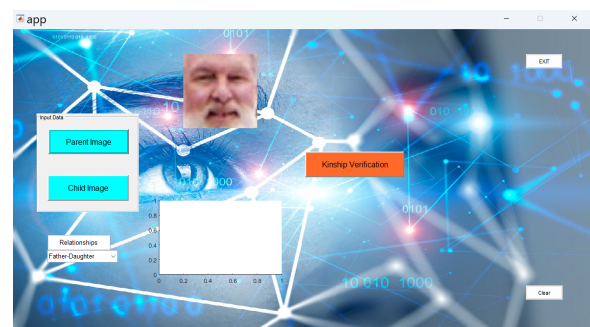


Figure 3.16: The basic user interface

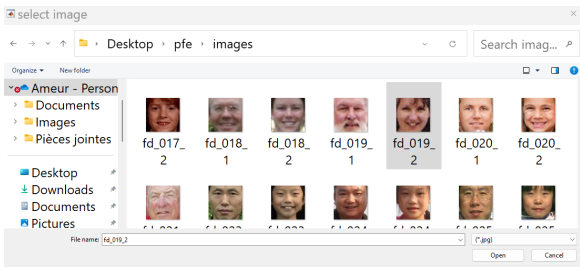


(a) When the user clicks on Parent image, it shows the files that we want to choose from

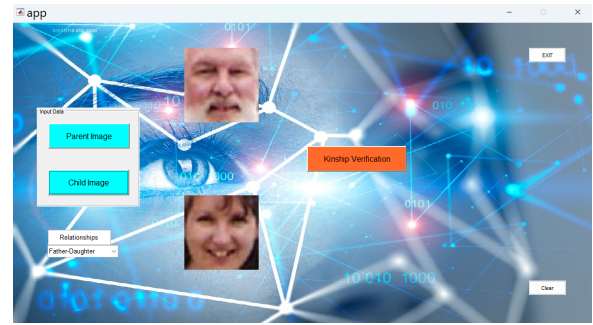


(b) The parent image displays on the basic interface

Figure 3.17: Selecting Parent Image



(a) When the user clicks on child image, it shows the files that we want to choose from



(b) The child image displays on the basic interface

Figure 3.18: Selecting Child Image

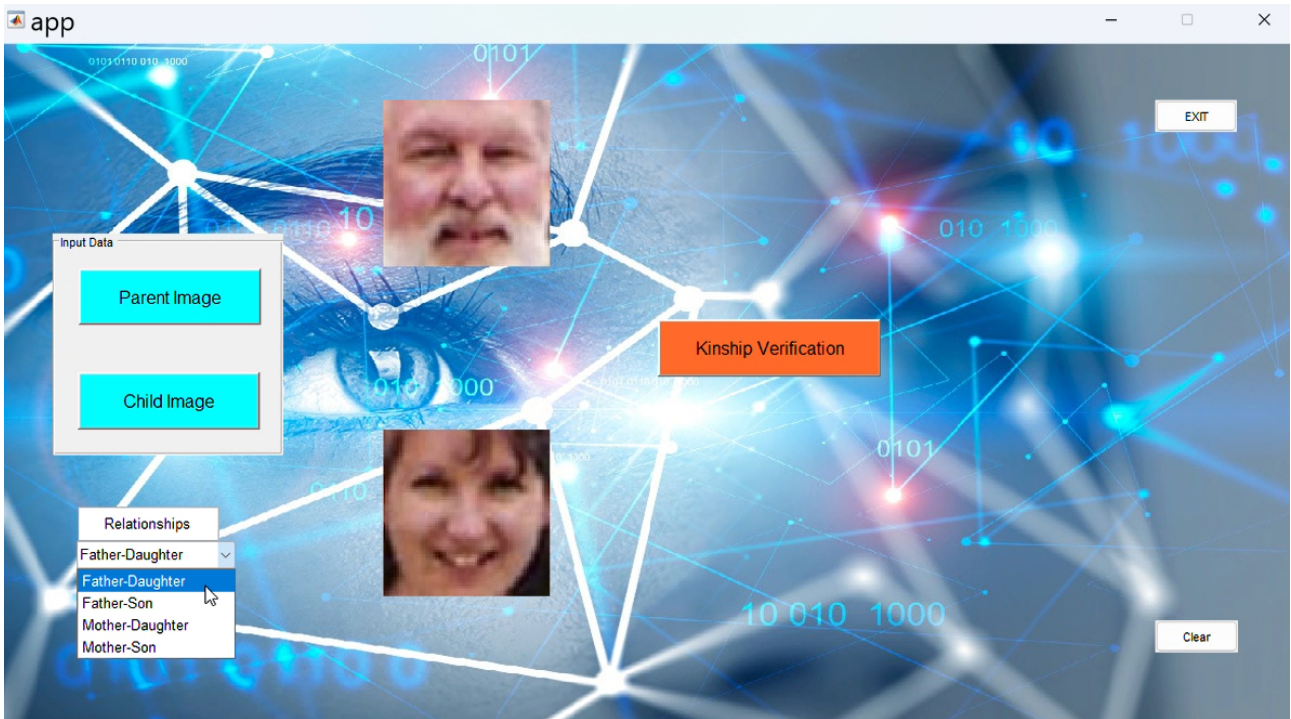


Figure 3.19: User Selection of Relationship Type for Verification

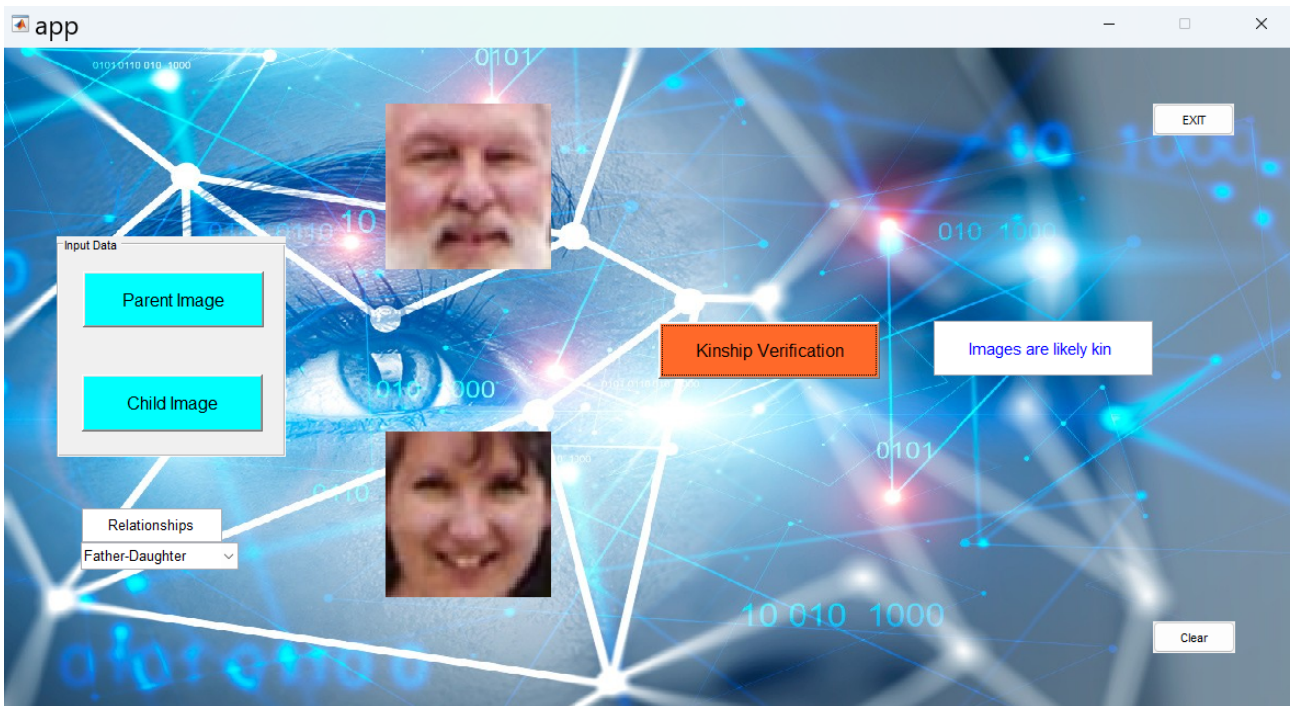


Figure 3.20: The final result (images are likely kin)

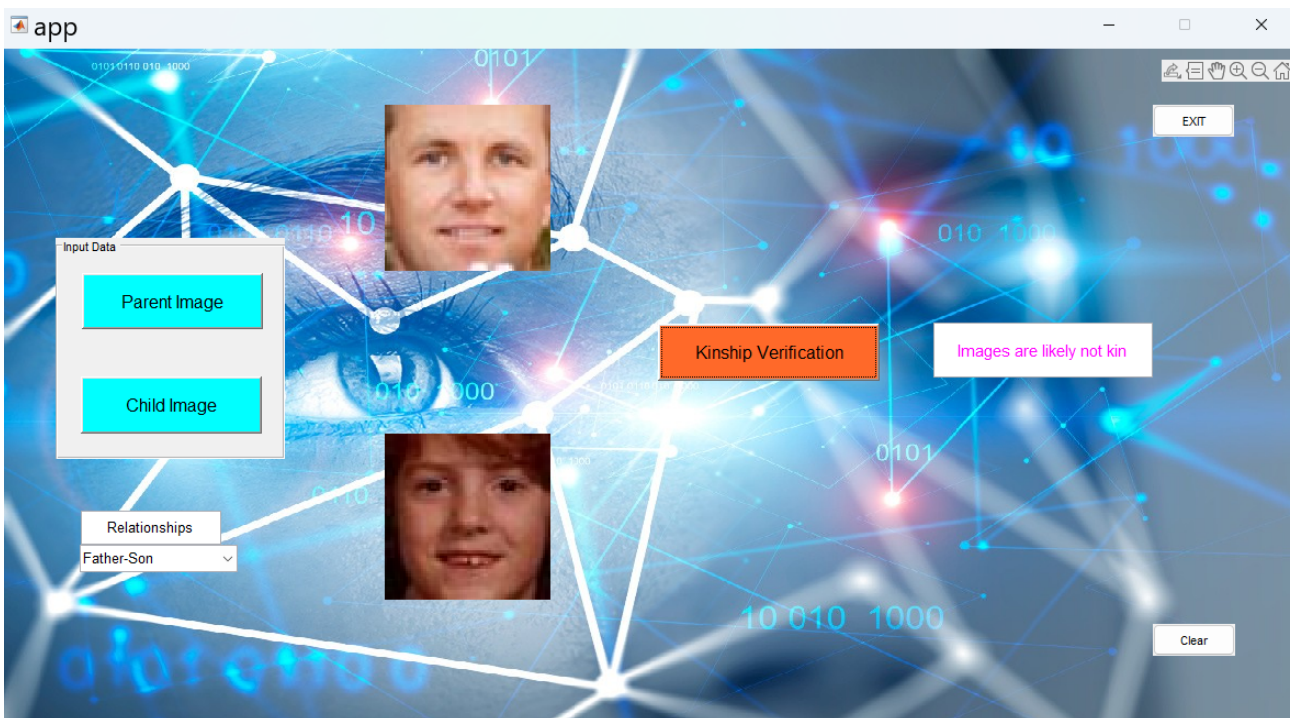


Figure 3.21: The final result (images are likely not kin)

3.8 Conclusion

The chapter wraps up by summarizing how we built and tested the kinship verification system. Our system proved to be effective, achieving 91.6% accuracy rate using the Hist LBP descriptor to analyze facial images. It also demonstrated robustness, handling variations in the number of patches and bins used during feature extraction well. We highlight the potential uses for this system in areas like law enforcement and social media, and we suggest some future improvements to make the system even better.

General Conclusion

Facial kinship verification uses computer vision techniques to analyze facial images and determine if two individuals belong to the same family. This project introduces a novel approach to kinship verification that leverages the strengths of the Histogram of (Hist LBP) descriptor. Hist LBP offers a significant advantage by capturing both texture and color information, potentially leading to a richer feature representation compared to traditional methods that rely on individual features like shapes, textures or colors. The extracted features are then fed into a Support Vector Machine (SVM) classifier, a robust machine learning algorithm trained using 5-fold cross-validation for reliable model evaluation. This approach helps mitigate overfitting and ensures the generalizability of the system.

To demonstrate the effectiveness of the explored approach, we conduct a comprehensive evaluation on the KinFaceW-II benchmark, a widely used dataset for kinship verification. This analysis goes beyond just accuracy by comparing our system's performance with established descriptors like Local Phase Quantization (LPQ) and (LBP) across various color spaces (grayscale, RGB, and HSV). We further investigated the impact of different normalization techniques applied to the Hist LBP descriptor by adjusting parameters to identify the optimal settings for future experiments. By analyzing the performance with various normalization approaches, we gain insights into how data scaling affects the effectiveness of feature extraction for kinship verification. The proposed model with power norm normalization technique achieved the highest accuracy of 91.6% for the optimal configuration parameters of Hist LBP descriptor. This finding demonstrates the potential of Hist LBP for kinship verification, especially when combined with appropriate normalization techniques. Based on accuracy, we compared our system's performance against state-of-the-art descriptors. This comparison demonstrates that the proposed model surpassing many comparative state-of-the-art techniques presented in this work.

Currently, the evaluation focused on the KinFaceW-II benchmark. In the future, we propose to apply the proposed model to a wider range of kinship verification datasets. This broader evaluation will provide a more comprehensive understanding of the system's performance across various facial characteristics, ethnicities, and image qualities.

While Hist LBP has demonstrated promising results for feature extraction, its potential can be further amplified by combining it with other complementary descriptors. Future work proposes investigating the fusion of Hist LBP features with other techniques like shape-based or deep learning-based feature extractors. This combined approach could potentially capture a richer and more informative representation of facial features, potentially leading to improved kinship verification accuracy.

Bibliography

- [1] Understanding roc curve.
- [2] S. Abdellah. *Image-based Facial Kinship Verification*. PhD thesis, University of Science and Technology HOUARI BOUMEDIENE, 2022.
- [3] D. Berrar et al. Cross-validation., 2019.
- [4] Z. Bi and P. Cao. Color space conversion algorithm and comparison study. In *Journal of Physics: Conference Series*, volume 1976, page 012008. IOP Publishing, 2021.
- [5] A. Datta and P. K. Banerjee. *Face Detection and Recognition Theory and Practice*. 10 2015.
- [6] H. Dibeklioglu, A. A. Salah, and T. Gevers. Are you really smiling at me? spontaneous versus posed enjoyment smiles. In *Computer Vision—ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III 12*, pages 525–538. Springer, 2012.
- [7] X. Duan and Z.-H. Tan. A feature subtraction method for image based kinship verification under uncontrolled environments. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 1573–1577, Quebec City, QC, Canada, Sep. 2015. IEEE.
- [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. IEEE, 2010.
- [9] Y. Fang, Y. Y. S. Chen, H. Wang, and C. Shu. Sparse similarity metric learning for kinship verification. In *2016 Visual Communications and Image Processing (VCIP)*, pages 1–4, Chengdu, China, Nov. 2016. IEEE.
- [10] M. Faraki, M. T. Harandi, and F. Porikli. No fuss metric learning, a hilbert space scenario. *Pattern Recognition Letters*, 98:83–89, Oct. 2017.
- [11] J. D. Foley. *Computer graphics: principles and practice*, volume 12110. Addison-Wesley Professional, 1996.
- [12] H. He, C. Ning, M. Liu, and J. Zhu. Tomato disease degree recognition based on rgb and lab color space conversion method. In *2023 IEEE 6th International Conference on Industrial Cyber-Physical Systems (ICPS)*, pages 1–6. IEEE, 2023.
- [13] L. Hong. *Automatic personal identification using fingerprints*. Michigan State University, 1998.
- [14] J. Hu, J. Lu, and Y.-P. Tan. Sharable and individual multi-view metric learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(9):2281–2288, Sep. 2018.

- [15] J. Hu, J. Lu, Y.-P. Tan, J. Yuan, and J. Zhou. Local large-margin multi metric learning for face and kinship verification. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(8):1875–1891, Aug. 2018.
- [16] 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.
- [17] G. Kumar and P. K. Bhatia. A detailed review of feature extraction in image processing systems. In *2014 Fourth international conference on advanced computing & communication technologies*, pages 5–12. IEEE, 2014.
- [18] O. Laiadi. *Kinship verification between two people by machine learning*. PhD thesis, Université Polytechnique Hauts-de-France ; Université Mohamed Khider (Biskra, Algérie), Sept. 2021.
- [19] R. Lan and Y. Zhou. Quaternion-michelson descriptor for color image classification. *IEEE Transactions on Image Processing*, 25(11):5281–5292, Nov. 2016.
- [20] H. Liu and C. Zhu. Status-aware projection metric learning for kinship verification. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 319–324, Hong Kong, Hong Kong, Jul. 2017. IEEE.
- [21] Q. Liu, A. Puthenputhussery, and C. Liu. A novel inheritable color space with application to kinship verification. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, Lake Placid, NY, USA, Mar. 2016. IEEE.
- [22] J. Lu, J. Hu, and al. The fg 2015 kinship verification in the wild evaluation. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–7, Ljubljana, May 2015. IEEE.
- [23] J. Lu, J. Hu, X. Zhou, Y. Shang, Y.-P. Tan, and G. Wang. Neighborhood repulsed metric learning for kinship verification. In *IEEE International Conference on Computer Vision and Pattern Recognition, 2012*, pages 2594–2601. IEEE, 2012.
- [24] J. Lu, X. Zhou, Y.-P. Tan, Y. Shang, and J. Zhou. Neighborhood repulsed metric learning for kinship verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):331–345, Feb. 2014.
- [25] Y. Lu, Z. Xu, and T. Ebrahimi. Towards visual saliency explanations of face verification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4726–4735, 2024.
- [26] Z. Lu, X. Jiang, and A. Kot. An effective color space for face recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2019–2023, Shanghai, Mar. 2016. IEEE.
- [27] M. Mazziotta and A. Pareto. Everything you always wanted to know about normalization (but were afraid to ask). *Rivista Italiana di Economia Demografia e Statistica*, 75(1):41–52, 2021.
- [28] S. A. Medjahed. A comparative study of feature extraction methods in images classification. *International journal of image, graphics and signal processing*, 7(3):16, 2015.
- [29] A. Moujahid and F. Dornaika. A pyramid multi-level face descriptor: application to kinship verification. *Multimedia Tools and Applications*, Aug. 2018.

- [30] N. Nader, F. E.-Z. A. El-Gamal, and M. Elmogy. Enhanced kinship verification analysis based on color and texture handcrafted techniques. *The Visual Computer*, 40(4):2325–2346, 2024.
- [31] W. S. Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.
- [32] R. Peh. 16 data feature normalization methods using python with examples (part 3 of 6), 2020. Accessed: 2024-05-27.
- [33] M. Pietikäinen. Local binary patterns. *Scholarpedia*, 5(3):9775, 2010.
- [34] X. Qin, D. Liu, and D. Wang. Heterogeneous similarity learning for more practical kinship verification. In *Neural Processing Letters*, volume 47, pages 1253–1269. Springer, Jun. 2018.
- [35] X. Qin, X. Tan, and S. Chen. Tri-subject kinship verification: Understanding the core of a family. *IEEE Transactions on Multimedia*, 17(10):1855–1867, 2015.
- [36] L.-D. Quach, K. N. Quoc, A. N. Quynh, and H. T. Ngoc. Evaluating the effectiveness of yolo models in different sized object detection and feature-based classification of small objects. *Journal of Advances in Information Technology*, 14(5):907–917, 2023.
- [37] D. Singh and B. Singh. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 2019.
- [38] D. Singh and B. Singh. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97:105524, 2020.
- [39] M. Soltane and M. Bakhti. Multi-modal biometric authentications: concept issues and applications strategies. *International Journal of Advanced Science and Technology*, 48, 2012.
- [40] A. K. J. Stan Z. Li. *Handbook of Face Recognition*. Springer London Dordrecht Heidelberg New York, 2011.
- [41] Statology. How to transform data in python (log, square root, cube root), 2024. Accessed: 2024-06-19. Available at: <https://www.statology.org/transform-data-in-python/>.
- [42] M. Terribile. Understanding cross-validation’s purpose, 2020. Accessed: 2024-05-29. Available at: <https://medium.com/@mtterribile/understanding-cross-validations-purpose-53490faf6a86>.
- [43] K. D. Toennies. *An Introduction to Image Classification: From Designed Models to End-to-End Learning*. Springer, 2024.
- [44] T. N. Van and V. T. Hoang. Kinship verification based on local binary pattern features coding in different color space. In *2019 26th international conference on telecommunications (ICT)*, pages 376–380. IEEE, 2019.
- [45] M. D. Vigeland, F. L. Marsico, M. Herrera Piñero, and T. Egeland. Prioritising family members for genotyping in missing person cases: A general approach combining the statistical power of exclusion and inclusion. *Forensic Science International: Genetics*, 49:102376, 2020.
- [46] W. Wang, S. You, S. Karaoglu, and T. Gevers. A survey on kinship verification. *Neuro-computing*, 525:1–28, 2023.

- [47] H. Wu, J. Chen, X. Liu, and J. Hu. Component-based metric learning for fully automatic kinship verification. *Journal of Visual Communication and Image Representation*, 79:103265, 2021.
- [48] S. Xia, M. Shao, J. Luo, and Y. Fu. Understanding kin relationships in a photo. *IEEE Transactions on Multimedia*, 14(4):1046–1056, 2012.
- [49] F. Xiong, M. Gou, O. Camps, and M. Sznai. Person re-identification using kernel-based metric learning methods. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part VII 13*, pages 1–16. Springer, 2014.
- [50] H. Yan. Kinship verification using neighborhood repulsed correlation metric learning. *Image and Vision Computing*, 60:91–97, Apr. 2017.
- [51] H. Yan, J. Lu, W. Deng, and X. Zhou. Discriminative multimetric learning for kinship verification. *IEEE Transactions on Information Forensics and Security*, 9(7):1169–1178, Jul. 2014.
- [52] H. Yan, J. Lu, and X. Zhou. Prototype-based discriminative feature learning for kinship verification. *IEEE Transactions on Cybernetics*, 45(11):2535–2545, Nov. 2015.

Abstract

Facial Kinship Verification, the process of identifying family relationships through facial images, is gaining more interest due to its many applications. This work proposes the use of feature extraction technique for kinship verification: the Histogram of Local Binary Patterns (Hist LBP). Hist LBP offers a richer representation of facial features by capturing both texture and color information. We evaluate its effectiveness against established methods like Local Phase Quantization (LPQ) and Local Binary Patterns (LBP) across various color spaces to assess its potential for improvement. Furthermore, we explore the impact of different normalization techniques applied to the Hist LBP descriptor in order to further enhance performance. The normalized Hist LBP features are then fed into a Support Vector Machine (SVM) classifier with 5-fold cross-validation for robust model training and evaluation. We evaluate the system's performance using the KinFaceW-II benchmark, allowing us to analyze the impact of Hist LBP on kinship verification accuracy compared to existing approaches. Ultimately, this work strives to contribute to the advancement of facial kinship verification technology by providing a robust and effective solution.

Keywords: Features, color space, classification, descriptors

Resume

La vérification de la parenté faciale, qui consiste à identifier les relations familiales à partir d'images faciales, suscite un intérêt croissant. Ce travail présente une nouvelle technique d'extraction de caractéristiques appelée Histogramme des motifs binaires locaux (Hist LBP), qui capture les informations de texture et de couleur des visages. L'efficacité de Hist LBP est comparée à des méthodes établies comme la quantification de phase locale (LPQ) et les motifs binaires locaux (LBP) dans divers espaces colorimétriques. Des techniques de normalisation sont également explorées pour améliorer les performances. Les caractéristiques normalisées sont ensuite utilisées dans un classificateur SVM avec validation croisée en 5 parties. Les performances du système sont évaluées avec le référentiel KinFaceW-II pour analyser l'impact de Hist LBP sur la précision de la vérification de la parenté. L'objectif est de contribuer à l'avancement de cette technologie en proposant une solution robuste et efficace.

Mot cle: Caractéristiques, espace colorimétrique, classification, descripteurs

ملخص

هناك اهتمام متزايد باختبار قرابة الوجه، والذي يتضمن تحديد العلاقات العائلية من صور الوجه. يقدم هذا العمل تقنية جديدة لاستخراج الملامح تُدعى المدرج التكراري للأنماط الثنائية المحلية (Hist LBP)، والتي تلتقط معلومات اللمس واللون من الوجوه. تتم مقارنة فعالية Hist LBP بالطرق المعمول بها مثل تحديد الطور المحلي (LPQ) والأنماط الثنائية المحلية (LBP) في مساحات لونية مختلفة. يتم أيضًا استكشاف تقنيات التطبيع لتحسين الأداء. يتم بعد ذلك استخدام الميزات المعيارية في مصنف SVM مع التحقق التبادلي المكون من 5 أجزاء. يتم تقييم أداء النظام باستخدام KinFaceW-II لتحليل تأثير Hist LBP على دقة التحقق من النسب. الهدف هو المساهمة في تطوير هذه التقنية من خلال اقتراح حل قوي وفعال. الكلمات المفتاحية: الخصائص، مساحة الألوان، التصنيف، الواصفات