Larbi Ben M'hidi University, Oum El Bouaghi, Algeria
Faculty of Exact Sciences, Nature Sciences and Life
Department of Mathematics and Computer Science

Universidad de Las Palmas de Gran Canaria
Doctorado en Empresa, Internet y Tecnologías
de las Comunicaciones (EmITIC)

THESIS

Offline handwritten signature verification and forgery detection

Submitted, in partial fulfilment
of the requirements for the degree
of Doctor in Computer Science

Author:

Walid Bouamra

Supervised by:

Pr. Miguel. A. Ferrer	Professor	ULPGC University, Las Palmas de Gran Canaria, Spain
Pr. Brahim Nini	Professor	Larbi Ben M'hidi University, Oum El Bouaghi, Algeria
Dr. Moises Diaz	A. Professor	ULPGC University, Las Palmas de Gran Canaria, Spain
Dr. Chawki Djeddi	A. Professor	Larbi Tebessi University, Tebessa, Algeria

Jury members:

President:

Pr. Mekhlouf Derdour	Professor	Larbi Ben M'hidi University, Oum El Bouaghi, Algeria

Examinators:

Pr. Khadir Mohamed Tarek	Professor	Badji Mokhtar University, Annaba, Algeria
Dr. José Juan Quintana	A. Professor	ULPGC University, Las Palmas de Gran Canaria, Spain
Dr. Cristina Carmona-Duarte	A. Professor	ULPGC University, Las Palmas de Gran Canaria, Spain

September 2022

Part of the development of this work using Run Lengths Algorithms was awarded with the second and third place in the "Name Component and Signature Verification Tasks" in the ICFHR2018 Competition on Thai Student Signatures and Name Components Recognition and Verification.

The author was the winner of the ICFHR 2020 Competition on short answer assessment and Thai student signature and name components recognition and verification in the Task 3 - Short Answer Recognition.

I would like to dedicate this thesis,
To my dear parents, under whose feet resides heaven,
To my darling wife and precious children,
To the pure souls of my dear brothers "Tarek" and "Soufiane," May God's mercy and blessings be upon them ...

Walid

Acknowledgements

"Coming together is a beginning, Keeping together is progress, Working together is success."
Edward Everett Hale

Throughout the completion of this cycle, I have received a great deal of support and assistance.

I would like to express my infinite appreciation to my dear supervisors, who were the light that illuminates my path throughout my PhD career.

I'd like first to express my gratitude to Professor Miguel Angel Ferrer for accepting the scientific partnership and twinning between the Spanish and Algerian universities, as well as for the tremendous effort he put in to overcome all obstacles and barriers to the cotutelle's success, which provided me with new horizons and knowledge. Similarly, I applaud him for considering me as a member of his prestigious university. I am deeply grateful to him for the warm reception and unparalleled welcome that I received at the level of the EMITIC laboratory throughout my studies and during my stay in this laboratory. Pr. Ferrer provided me with all the material and moral support, as well as all the means necessary to work comfortably and progress to complete this phase in the best circumstances.

Secondly, I would also like to extend my heartfelt thanks to my dear supervisor Dr. Moises Diaz. Who has always been my tutor, adviser, teacher, and friend. He was continuously earnest about the progress of the work, tireless, and persistent. I always resorted to him when facing difficulties and problems, and he spared no effort to help me. His support went beyond the scientific and academic aspects; he supported me even outside work time. The presence of Dr. Moises Diaz has been pivotal to me, given his flexible approach, extraordinary diligence, and insight.

I equally would like to show my thankfulness to my respectable professor Brahim Nini for his unconditional support and assistance. He was patient and meek; his careful observations and sound guidance enlightened me and removed the ambiguity from all the dark spots during the period he supervised me. Mr. Nini has an insight into phases and events; he can wisely determine the most appropriate step at the right time, moving the student forward with confident steps and successful planning.

I'd like also to express my appreciation to Dr. Chawki Jeddi, my professor and friend, for his significant help in both theoretical and practical matters. His knowledge and comprehension of the subject were important in allowing me to finish the topics in less time and with higher efficiency.

Mr. Fateh Boutekkouk, the project leader, has always been humble and generous, and has listened to my interests and ambitions. I am grateful to him for allowing me and my colleagues to participate in this once-in-a-lifetime scientific opportunity.

My thanks belong to Mr. Tawfik Marir, the head of the Scientific Council. I thank him for his sincere support for me and my supervisors at various stages and that he was always by my side, side by side; his great support accelerated the pace of work.

Without forgetting all the professors and administrators whose contributions had a significant impact on my reaching this stage, I would like to mention them:

Mr. Samir Gramma, for his unconditional support and response to all my relied requests during the last period. Mr. Samir was like his brother's faithful brother, especially in difficult and delicate stages. His directives were sound, and his advice constructive and useful; he was very loyal to his promises and sincere in his diligence.

Mme. Cristina Carmona Duarte, is a respected colleague with whom I shared the laboratory throughout the study and research period in EDITIC laboratory. She was always supportive and helpful, and her advice was like precious pearls that removed all questions and confusion.

In addition to the executive staff of the two universities, Larbi Ben M'hidi in Algeria and ULPGC in Spain. Educators and administrators, whether in the university administration, the laboratory, or the offices where I received help and asset.

Finally, I would not have been able to finish this thesis without the help of many others. I appreciate them all on their own behalf, as well as on behalf of my fellow PhD students, friends, instructors, and everyone else who helped in any way.

Abstract

Handwritten signature is widely accepted as a means of verifying a person's identity. As a result, an automatic signature verification system would allow for more efficient verification and lowering the possibility of falsification. Several approaches have been presented in order to provide as much reliable signature verification as possible.

This thesis attempts to shed the light on the difficult issue of distinguishing a genuine signature from a forged one. In this context, it presents a novel offline verification system for handwritten signature verification that uses handcrafted techniques as signature textural descriptors.

The dissertation introduces three techniques for the signature verification: (i) run-length features, (ii) the multidirectional run-length features and (iii) the spiral run-length features.

Run-length method is a technique applied on signature images based on counting neighboring pixels having the same value; the set of these pixels constitutes a run. Then, the number of runs with the same length will be regrouped in a matrix M, which composes the run-length matrix. This technique has been applied to binary images, and the computation of the runs is done in the four main directions by counting the black pixels as well as the white ones.

The other two techniques are inspired from the run-length distributions. The multidirectional run-length features reinforce the use of run-length features by adding to the four main directions four other directions and supporting each one by its direct neighborhood, ultimately comprising eight composite directions. Therefore, the spiral run-length features are presented as a fifth direction of the classic run-length distributions, following spiral browsing of the signature, including the horizontal and vertical directions permanently.

For a concrete evaluation, we performed many experimentations. We applied our algorithms on well-known databases such as GPDS960, MCYT-75, and CEDAR. For the classification phase, we employed the One Class Support Vector Machine (OC-SVM) to simulate the real case where we do not have enough genuine samples.

Compared with the state-of-the-art, the obtained results are promising. Moreover, we had satisfactory rankings against other algorithms.

Keywords: Offline signature verification, handcrafted method, run-length distributions, multidirectional run-length features, spiral run-length features.

الملخص

الإمضاء بخط اليد مقبول على نطاق واسع كوسيلة للتحقق من هوية الشخص. ونتيجة لذلك، سيسمح نظام التحقق التلقائي من التوقيع بمزيد من الفعالية في التحقق من صدقيته وتقليل احتمالية التزوير. تم تقديم العديد من الأنظمة من أجل توفير أكبر قدر ممكن من التحقق من التوقيع الموثوق. تحاول هذه الأطروحة إلقاء الضوء على المسألة الصعبة المتمثلة في التمييز بين التوقيع الحقيقي والتوقيع المزور. في هذا السياق، نقدم نظامًا جديدًا للتحقق دون اتصال من صحة التوقيع بخط اليد والذي يستخدم تقنيات يدوية كوصف نصي للتوقيع. تقدم الأطروحة ثلاث تقنيات للتحقق من صحة التوقيع: (1) توزيعات أطوال القطع، (2) توزيعات أطوال القطع متعددة الاتجاهات و (3) توزيعات أطوال القطع اللولبية. طريقة توزيعات أطوال القطع هي تقنية يتم تطبيقها على صورة التوقيع بناءً على حساب وحدات البيكسل المتجاورة التي لها نفس القيمة؛ تشكل هذه المجموعة من البيكسل قطعة. بعد ذلك، يتم إعادة تجميع عدد القطع التي لها نفس الطول في مصفوفة M، والتي تشكل مصفوفة أطوال القطع. تم تطبيق هذه التقنية على الصور ثنائية اللون، وتتم عمليات الحساب في الاتجاهات الأربعة الرئيسية عن طريق حساب نقاط البيكسل السوداء وكذلك البيضاء. التقنيتان الأخريان منبثقتان من توزيعات أطوال القطع. تعزز توزيعات أطوال القطع متعددة الاتجاهات استخدام ميزات أطوال القطع من خلال إضافة أربعة اتجاهات إضافية إلى الاتجاهات الأربعة الرئيسية، ودعم كل اتجاه من خلال جواره المباشر، والذي يشتمل في النهاية على ثمانية اتجاهات مركبة. بالإضافة إلى ذلك، يتم تقديم توزيعات أطوال القطع اللولبية كاتجاه خامس لتوزيعات أطوال القطع الكلاسيكية، بعد التصفح اللولبي للتوقيع، بما في ذلك الاتجاهات الأفقية والرأسية بشكل متناوب. لتقييم ملموس، أجرينا الكثير من التجارب. طبقنا خوارزميات على قواعد بيانات معروفة مثل GPDS960، MCYT-75، وCEDAR. بالنسبة لمرحلة التصنيف، استخدمنا حامل فواصل ذو هامش واسع بفئة واحدة (OC-SVM) لمحاكاة الحالة الحقيقية حيث لا يتوفر لدينا عادة عينات أصلية كافية. بالمقارنة مع أحدث ما توصلت إليه التكنولوجيا، فإن النتائج التي تم الحصول عليها واعدة. علاوة على ذلك، حصلنا على تصنيفات مرضية مقابل الأنظمة الأخرى.

الكلمات الدلالية التحقق من التوقيع دون اتصال ، الطريقة اليدوية ، توزيعات أطوال القطع ، توزيعات أطوال القطع الحلزونية ، توزيعات أطوال القطع متعددة الاتجاهات.

Resumen

La firma manuscrita es ampliamente aceptada como un medio para verificar la identidad de una persona. Como resultado, un sistema automático de verificación de firmas permitiría una verificación más eficiente y reduciría la posibilidad de falsificación. Se han presentado varios enfoques para proporcionar una verificación de firma lo más fiable posible.

Esta tesis intenta arrojar luz sobre la difícil cuestión de distinguir una firma genuina de una falsificada. En este contexto, presenta un novedoso sistema de verificación offline para la verificación de firmas manuscritas que utiliza técnicas artesanales como descriptores texturales de la firma.

La disertación presenta tres técnicas para la verificación de firmas: (i) características de longitud del segmento, (ii) características de longitud del segmento multidireccional y (iii) características de longitud del segmento en espiral.

El método de longitud del segmento es una técnica aplicada en imágenes de firma basada en el conteo de píxeles vecinos que tienen el mismo valor; el conjunto de estos píxeles constituye una corrida. Luego, el número de corridas con la misma longitud se reagrupará en una matriz M, que compone la matriz de longitud de corrida. Esta técnica se ha aplicado a imágenes binarias, y el cálculo de las corridas se realiza en las cuatro direcciones principales contando los píxeles negros y los blancos.

Las otras dos técnicas están inspiradas en las distribuciones de longitud del segmento. Las características de longitud del segmento multidireccional refuerzan el uso de las características de longitud del segmento añadiendo a las cuatro direcciones principales otras cuatro direcciones y apoyando cada una en su vecindad directa, lo que finalmente comprende ocho direcciones compuestas. Por lo tanto, las características de longitud del segmento en espiral se presentan como una quinta dirección de las distribuciones de longitud del segmento clásicas, siguiendo la navegación en espiral de la firma, incluidas las direcciones horizontal y vertical de forma permanente.

Para una evaluación concreta, realizamos muchas experimentaciones. Aplicamos nuestros algoritmos en bases de datos conocidas como GPDS960, MCYT-75 y CEDAR. Para la fase de clasificación, empleamos One Class Support Vector Machine (OC-SVM) para simular el caso real en el que no tenemos suficientes muestras genuinas.

En comparación con el estado del arte, los resultados obtenidos son prometedores. Además, obtuvimos clasificaciones satisfactorias frente a otros algoritmos.

Palabras claves: Verificación de firma estática, método manual, distribuciones de longitud de segmento, características de longitud de segmento en espiral, características de longitud de segmento multidireccional.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Introduction

1.1.1 Overview

Most applications in our daily lives, such as legal procedures and financial transactions, require a handwritten signature at the end of legal and financial documents to determine and distinguish the individual's identity.

Signatures, as well as eye, face, voice, and hand geometry, are all biometric traits. However, the signature differs from other types of biometrics in that getting a person's signature is relatively simple and widely accepted. Correspondingly, it is a cost-effective and time-efficient method of obtaining it.

The necessity of having to sign the administrative files and documents indicates their importance in dealing between persons and various administrative and official bodies, such as the municipality, the bank, and the tax authority. In turn, these departments, and to verify the file's legal validity, deal with the following question: *"Is this signature correct or forged?"* In other words: *"How do we know if the signer of an official document is who he claims to be?"*

This issue urged the development of a method to validate signature legitimacy, which resulted in the conception of handwritten signature verification systems.

This thesis describes some handwritten signatures verification solutions to improve the recognition system of individuals using their signatures. The developed algorithms supported by experimental results indicate the benefit of applying this biometrics mode in several areas as banking and forensic fields.

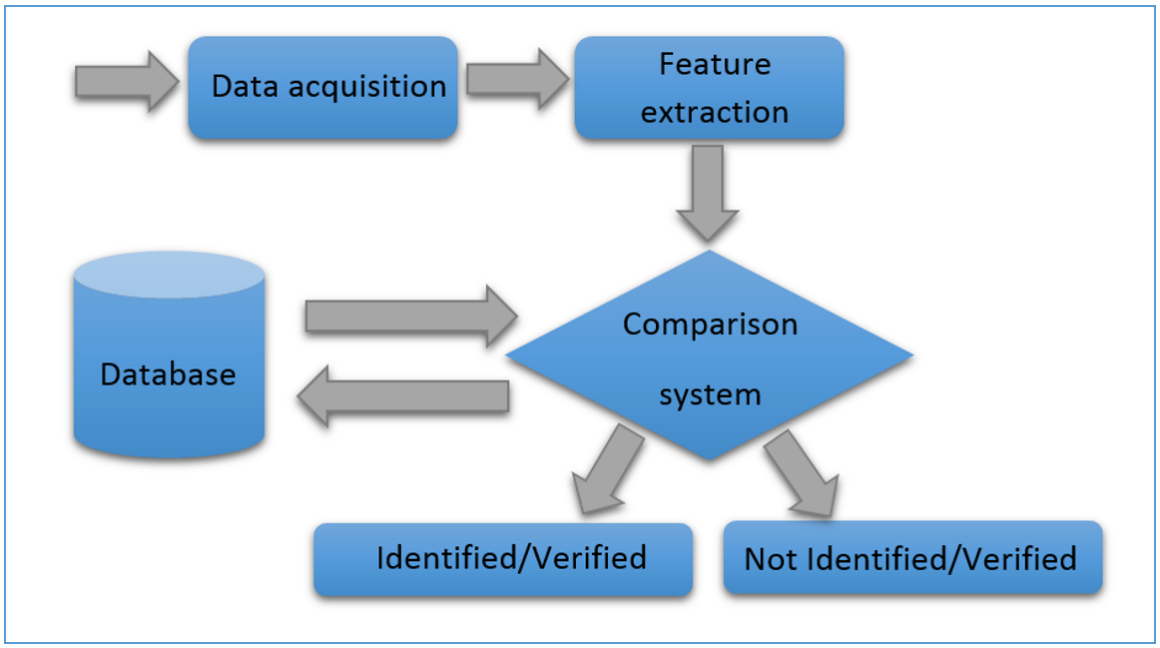


Fig. 1.1 General structure in a biometric system.

1.1.2 Biometrics

Jain, et al. define biometrics as "the science of validating the identification of an individual based on the physical, chemical, or behavioral features" (Jain et al., 2011).

Individual identification in the past was mostly focused on traditional and basic qualities such as passwords and access cards that were only known by the user. Due to the ease with which these systems may be pilfered, they were not as effective as planned. Furthermore, multiple passwords make it difficult to remember, store, and recall them when needed. Thus, biometrics confirms an individual's identity based on who he is or what he produces, rather than what he knows (password) or what he has (ID card), making it less prone to counterfeit attempts than traditional means of identification (Faundez-Zanuy, 2006; Jain et al., 2011). As a result, a biometric system can be used as a pattern recognition algorithm. As demonstrated in Fig. 2.1, it is capable of producing adequate and secure issues for identifying and recognizing individuals.

1.1.3 Modalities

The biometric system (Fig 2.3) can be established on a variety of biometric modalities, theoretically containing the majority of an individual's physiological and behavioral characteristics (Faundez-Zanuy, 2006). However, in order to be considered, these traits must satisfy a set of criteria (Jain et al., 2011).

Biometric identifier	Universality	Distinctiveness	Permanence	Collectability	Performance	Acceptability	Circumvention
DNA	H	H	H	L	H	L	L
Ear	M	M	H	M	M	H	M
Face	H	L	M	H	L	H	H
Facial thermogram	H	H	L	H	M	H	L
Fingerprint	M	H	H	M	H	M	M
Gait	M	L	L	H	L	H	M
Hand geometry	M	M	M	H	M	M	M
Hand vein	M	M	M	M	M	M	L
Iris	H	H	H	M	H	L	L
Keystroke	L	L	L	M	L	M	M
Odor	H	H	H	L	L	M	L
Palmprint	M	H	H	M	H	M	M
Retina	H	H	M	L	H	L	L
Signature	L	L	L	H	L	H	H
Voice	M	L	L	M	L	H	H

Fig. 1.2 Qualitative comparison of biometric modalities (Jain et al., 2011).

- Acceptability: public agreement for the measurement of the feature,
- Universality: each individual must have the biometric trait,
- Uniqueness: the probability of having a common feature is almost null.
- Permanence: the biometric attribute must be almost stable over time.
- Measurability: the quantification of the biometric trait in a practical way is achievable.
- Performance: the biometric characteristic must ensure speed and precision.

The distribution of these conditions across biometric modalities is depicted in Fig. 2.2. The signature, for instance, is a widely acknowledged biometric that, on the other hand, does not demonstrate a high level of circumvention resistance and shows a lack of performance and permanence (Jain et al., 2011).

Different modalities of biometric systems, to varying degrees, approve of these qualities. Biometric modalities are used in various applications. These are divided into 2 groups:

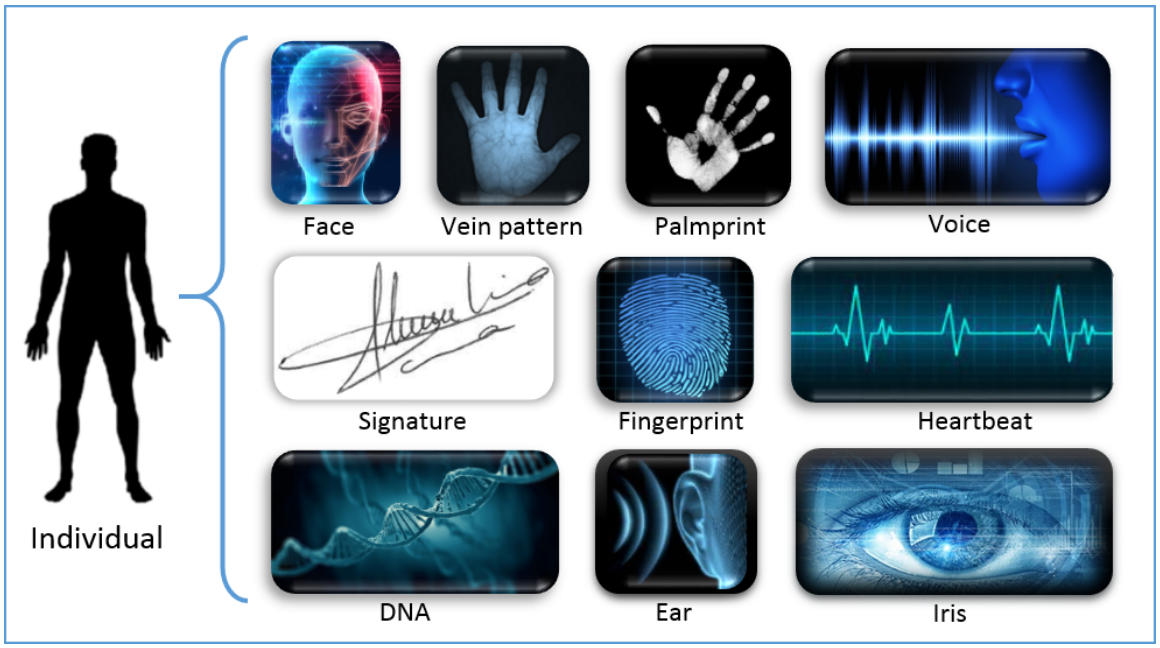


Fig. 1.3 Biometrics modalities.

- Physiological modalities: fingerprints, iris, face, hand geometry, and so forth. They are determined by morphological or biological factors.
- Behavioral Modalities: employ a unique attribute of behavior such as a signature, keystrokes, gait, or voice.

1.1.4 Handwritten signatures

The signature was an essential part of human identity for thousands of years. Since 3100 B.C, people have known the signature concept like Egyptians and Sumerians who began using a set of symbols and pictures - pictographs, the tablet containing the signature was in clay. Some centuries later, Greek and Roman cultures used signatures, they adopted the Phoenician alphabet around 1200 B.C, and within 600 years, they adopted the Latin alphabet. In 1677 the English Parliament passed an official act basing on the handwritten signature as the effective guarantee against frauds and falsification. The signature was a binding contract and used widely around the world in 1776 by the John Hancock signed America's Declaration of Independence[1].

In 1980, the signature concept was changed by the evolution of technology and the use of the fax machine. As a result, the signature has been sent electrically and replaced the

[1]The history of the Signature, legalesign.com/blog/history-of-signatures

traditional signature method. By 2000, the electronic contracts were validated by the US president Bill Clinton, which allowed to sign acts and contracts from anywhere and launched the eSignature.

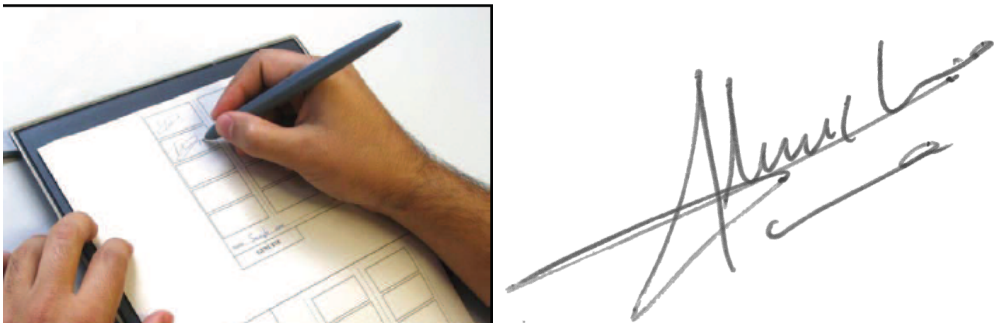
By the time John Hancock signed America's Declaration of Independence in 1776, (Bidwell, 1988), the signature was a binding contract and used widely worldwide . Nowadays, the signature is the most accepted biometric modality both on the social and legal side. On the social side, we used to apply it as a means of authentication in our daily practices, such as signing checks. Besides, in an authentication process, a person more easily accepts signing a document than taking a photo of their face, iris, or retina, which may have impediments related to time spent or the person's health. On the other hand, for legal acceptance of the signature, the laws allow a person to be accused based on their signature, such as when a person signs a check with no balance (Radhika and Gopika, 2015).

Because the handwritten signature has long been the most popular and required method of identifying and verifying persons, as shown in Fig. 2.2, it is a common biometric property shared by all modes employed for this purpose (Fairhurst, 1997; Fairhurst and Kaplani, 2003; Leclerc and Plamondon, 1994; Pirlo, 1994; Plamondon and Lorette, 1989). Signatures are also recognized as a legitimate manner for legislative, financial, and inspecting authorities to verify people's identities. On the other hand, performing an identity verification procedure using handwritten signatures does not necessitate any invasive measures because people are accustomed with this type of biometrics in everyday life (Plamondon and Srihari, 2000).

The signature, among others, is naturally affected by the signer's psychic and physical conditions, which affect the shape and details of his signature. This drawback is considered by many theories that have checked the signer's state (Plamondon, 1995, 1998) and the way to write the signature (Doermann and Rosenfeld, 1995).

Despite the large number of biometric traits adopted for the identification of persons as signature, voice, iris, hand geometry, face, and fingerprint, a single trait remains more or less not competent to define all the properties and characteristics desired for a biometric system (Jain et al., 2000). Thus, we proceed to social and cultural obstacles before discussing technical difficulties (Veeramachaneni et al., 2005; Vielhauer and Dittmann, 2006). Nevertheless, some of these traits remain more adequate then others for verifying individuals and determining their identity.

Verifying signatures is a rigorous and punctual task, mainly when this process is based on a limited number of a few reference samples to decide the authenticity or falsification of a signature. This issue is due to a set of characteristics distinguishing the handwritten signature, such as the ubiquity of its use for verifying an individual (Impedovo and Pirlo, 2008; Plamondon, 1994).



(a) Online signature (b) Offline signature

Fig. 1.4 Signature as a behavioral biometric trait.

1.1.5 On-line and off-line signature

The signature is a behavioral biometric feature that allows the owner to be identified or authenticated. As indicated in Fig. 1.4, there are two kinds of signatures: online and offline, according to their type of acquisition. A dedicated device, such as a tablet, acquires the online (or dynamic) signature and provides the device's dynamic features. The offline (or static) signature, on the other hand, merely contains information about the signature form. It is first printed on paper, then scanned or photographed and put into the system as an image. Online signatures are more difficult to counterfeit than offline signatures due to dynamic features such as pressure and speed. When analyzing a signature on paper, these dynamic factors have a more distinctive ability to discriminate between distinct signatures and are more difficult to replicate (Liwicki et al., 2011).

This thesis will be interested only in the offline signature, definition, types of forgeries, and signature verification systems.

1.1.6 Types of forgeries

In handwritten signature verification systems, forgery might be classified and categorized into three basic kinds:

- Random forgery: The forger has no access to the original signature or any information about the author's name in this type of forgery; nonetheless, the forger can duplicate the random signature (Bertolini et al., 2010).
- Simple forgery : the forger does not have access to the signature sample, but it is aware of the author's name and normally generate the signature in his own manner (**?**).

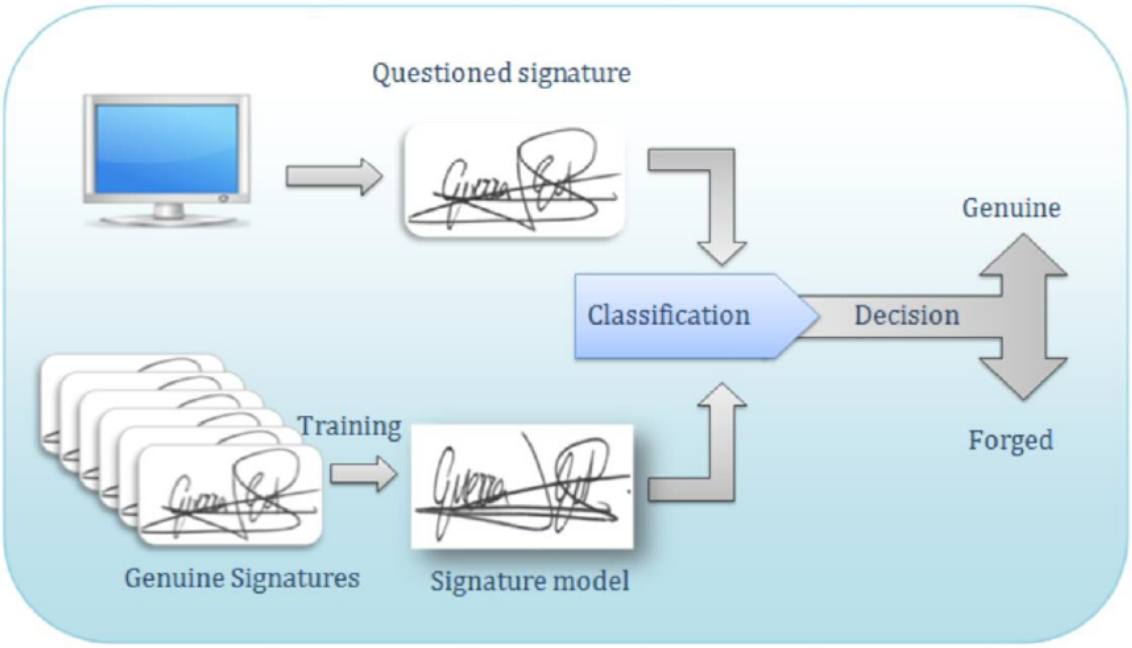


Fig. 1.5 Diagram of an offline signature verification system.

- Skilled forgery : in this kind, the skilled forger has access to authentic signature samples and may thus recreate it (Bouamra et al., 2018; Liwicki et al., 2011).

1.1.7 Signature verification

The signature verification is the process of authenticating a person based on their signature. The handwritten signature verification system must determine whether the signature is indeed produced by the original signer. The user is accepted if his signature is original; otherwise, he is considered an impostor (forger) (Bouamra et al., 2018). A general overview is depicted at Fig. 1.5.

The handwritten signature verification systems are classified into two types online and offline depending on the type of signature acquisition: online or offline signature as shown in Fig. 1.4. Offline signature verification includes a scanned document and static signature image, while the online signature verification systems use dynamic signature and are based on time functions (Diaz et al., 2019).

Thus, the steps of the verification process are the same except the first step depended on how the signature is acquired. The signature verification phases are concluded in four steps : data acquisition, preprocessing, feature extraction and classification (Diaz et al., 2019). We will show in detail the four phases description.

In an offline signature verification system, users are first introduced by registering a few samples of their signatures that serve as references. Later, when a user, claiming to be a particular enrolled in the system, presents his signature for verification, compared with the claimed individual's reference signatures. Finally, a score measuring the similarity (or dissimilarity) between the two signatures is provided at the end of this comparison. If the similarity score is greater (lower dissimilarity score, respectively) than a threshold, set in the system training step, the user is accepted; otherwise it is rejected (Bouamra et al., 2018) as shown in the Fig. 1.5.

Two main approaches are applied in the offline signature verification process are; the leaned approach and the handcrafted approach. The deep learning approach has emerged in the signature verification field; many problems were handled by deep learning using convolutional neural networks (CNN) in feature learning and classification (Dara and Tumma, 2018). This approach is present in the image processing and classification tasks as writer identification (Tang and Wu, 2016) and signature verification (Souza et al., 2018).

Meanwhile, in the second approach, many features have been manually designed or "handcrafted". The handcrafted features aim to find the right trade-off between accuracy and computational efficiency (Nanni et al., 2017). Handcrafted features are divided into two types according to the processing of the signature image; global features (treat the entire image) and local features (handles a part of the image). The widespread handcrafted feature extraction involves the use of histogram techniques (Liu et al., 2017), morphological operators, vector quantization (Agustsson et al., 2017), or the gradient (Bal et al., 2019) for texture classification. A wide variety of s-o-a handcrafted algorithms used for the signature verification are considered, such as Local Binary Pattern (LBP), Local Ternary Patterns (LTP), Local Directional Pattern (LDP), Autoregressive Coefficients (Ar-Coef), Local derivative pattern (LDerive)) and Run-Length features (RL). The handcrafted features have been implemented for decades and still serve as a powerful tool, especially for image classification and signature verification tasks.

In this thesis, we propose a system based on the Run-Length (RL) textural features; due to the effectiveness and efficiency of these features against other systems (Bouamra et al., 2018). Moreover, we participated in ICFHR18 (Suwanwiwat et al., 2018) and ICFHR20 (Das et al., 2020) competitions. The results encouraged using the RL features in offline signature verification and improving their performance by adding new directions, as this thesis will describe in detail.

1.2 Main Thesis

Several pattern recognition and image classification issues were overcome using run-length features, such as in writer recognition. (Djeddi et al., 2013) used run-length computations to characterize distinct writers' handwriting styles. They used this technique, taking into account the black and white pixels. In another contribution; (Djeddi et al., 2015) used the Run-length features in the signature verification procedure, employing the black pixels solely to characterize the signatures of individuals.

(Bouamra et al., 2018) applied the Run-Length computations in the signature verification, taking into consideration both of the black and white pixels, and considering the four main directions (Horizontal 0°, Vertical 90°, Right-Diagonal 45° and Left Diagonal 135°).

We could have been ranked among the first best systems in the two last competitions, respectively, ICFHR18 (Suwanwiwat et al., 2018) and ICFHR20 (Das et al., 2020); we were the winner in one task of the last one using the Run-Length method.

The initial RL features were based on four directions, and we introduced a new direction (fifth direction): the spiral Run-Length. This new representation for the offline signature aims to upgrade the RL distribution features. In the fifth direction, the spiral run-length direction runs the signature in a spiral way providing an outside to inside view. We combine this new direction to the standard four directions to improve the representation of the run-length features.

In another contribution, we introduce further spatial information to the Run-Length distribution; is the Multidirectional Run-Length features (Bouamra et al., 2018). These are concluding in two phases. The first one is to add more directions to the four principal RL directions. The second one is based on enforcing the information given by each direction by the information of the neighbor directions. This improvement of Run-Length features could provide more features and scan more peripheral area of the image-based signature.

1.3 Methodology

The effectiveness of Run-Length distributions is shown against the systems of the State-Of-the-Art. We employed well publicly known databases containing offline handwritten signatures; for handling this performance, they are free to download and easy to carry out. Furthermore, this community's widely consulted databases are frequently used, and other researchers can fairly compare results between the different handwritten signature verification systems.The GPDS-960 is the largest publicly known signature database (Vargas et al., 2007). It contains 881 users, with 24 genuine signatures and 30 skilled forgeries, that

were collected in grayscale. We used part of the corpus; GPDS-75 containing 75 users for the Spiral Run-Length topic, and the whole corpus GPDS-960 for the standard Run-length and the Multidirectional Run-Length distribution.

Moreover, in order to evaluate our systems; we employed another frequently used dataset, the CEDAR database (Kalera et al., 2004), including 55 singers; each one with 24 genuine signatures and 24 forged samples. The database introduced in the ICFHR18 and ICFHR20 by Das et al. (2018, 2020) for the signature verification task; was the Thai student signatures. Each of the 100 volunteers was required to sign his name 30 times. The corpus had 3000 authentic signatures in total. 12 skillfully forged signatures and 12 simple forged signatures were obtained for each genuine sample, for a total of 5400 signatures in this database.

The classification is carried out using One-Class Support Vector Machine (OC-SVM) and the Euclidian distance-based classifiers (Bouamra et al., 2020, 2018). The OC-SVM classifier employing only genuine samples in the training set matches the real-world scenario when only a limited number of authentic signatures are available. On the other hand, the Run-Length submitted systems in ICFHR competitions computed a score based on the Euclidian distance between the questioned signature and the reference signatures (Das et al., 2020; Suwanwiwat et al., 2018).

1.4 Main Contributions

The primary contributions in the thesis are:

International competitions in offline signature verification

- H. Suwanwiwat, A. Das, U. Pal, M. Blumenstein, "ICFHR 2018 Competition on Thai Student Signatures and Name Components Recognition and Verification (TSNCRV 2018)," 500-505.
- Abhijit Das, Hemmaphan Suwanwiwat, Umapada Pal, Michael Blumenstein: ICFHR 2020 Competition on Short answer ASsessment and Thai student SIGnature and Name COMponents Recognition and Verification (SASIGCOM 2020). 22-227

Our system was ranked first against the participated handcrafted systems in ICFHR18 and ICFHR 20 competitions. In addition, we were the winner in one of the proposed tasks in ICFHR20 using the Run-Length features.

Published papers

- Bouamra, W., Djeddi, C., Nini, B., Diaz, M., & Siddiqi, I. (2018). Towards the design of an offline signature verifier based on a small number of genuine samples for training. Expert Systems with Applications, 107, 182-195.

This contribution describes how to use the run-length features in offline signature verification. The standard Run-Length features are based on counting consecutive pixels with the same value, then summing the runs with the same number of pixels in an array containing the number of blocks; this operation affects black and white pixels. The calculation of the runs is done according to the four main directions, the horizontal, vertical, right-diagonal, and left-diagonal direction.

- Bouamra, W., Diaz, M., Ferrer, M. A., & Nini, B. (2020, June). Off-line Signature Verification Using Multidirectional Run-Length Features. In Proceedings of the 10th International Conference on Information Systems and Technologies (pp. 1-8).

We offer a novel use of run-length in signatures extending the number of the directions. The novelty in our system consists of defining new directions as well as the four primary directions, that is to say, four other directions, to scan more surface of the signature. After having 08 scan directions in total, we strengthen each direction by its direct neighborhood, resulting in a composite angle that enlarges the browsing of more orientations on the entire image-signature.

- Bouamra, W., Diaz, M., Ferrer, M. A., & Nini, B. (2022, June). Spiral based Run-Length Features for Offline Signature Verification. In Proceedings of the 20th International Conference on Graphonomics (pp. 1-15).

In this contribution, the classical run-length distribution is upgraded with an additional representation for off-line signatures. Furthermore, we add a fifth direction to the four standard directions of run-length features. The spiral run-length features traverse the entire image-signature in a spiral counterclockwise curve starting from the first pixel at the upper left corner of the image, moving towards the last central point. This spiral browsing rotates between the horizontal and the vertical directions.

Further, the contribution studies the combination of the novel direction with the previous standard ones at both feature and score levels. Finally, the obtained outcomes justify using the novel direction in run-length features.

1.5 Outline of the Dissertation

The thesis follows a typical and complicated framework that includes underlying theory, practical approaches, and multiple distinct experimental studies in which the techniques were applied. As a result, the structure is as:

- Chapter two involves the depiction of related works and literature analysis regarding run-length features.

  In this section, we'll illustrate the first implementation of run-length features and their basic functions, as well as the studyies that focused on improving and adapting them by adding some functions that supplemented their original ones. We'll also discuss the fields of application of these features, such as medical imaging and writer identification, before moving on to our main topic, handwritten signature verification. As far as we know, few contributions have used run-length features for the signature verification process (Bouamra et al., 2018).

- Chapter three is devoted to the use of run-length features in off-line signature verifications.

  On binary images of signatures, run-length distributions were determined, with the black pixels corresponding to the ink trace and the white pixels corresponding to the background of the signature image. The connected pixels having the same value make a 'run', the set of runs having the same length (number of pixels) are summed in a matrix P. The element $p(i, j)$ of the run-length matrix P, is defined by the color i and the position j in a given direction.

- Chapter four contributes with an extension of the run-length directions in a signature. Specifically, we extend from four directions to n directions. In addition, we had a written and published paper containing the description of the multidirectional run-Length.

- Chapter five proposes a new run-length direction based on spiral analysis of the offline signature. We already had a written and submitted paper about this contribution, the paper will be published soon.

- Chapter six closes the thesis with the concluding remarks and future work ideas.

Chapter 2

State-of-the-art on run-length features

2.1 Introduction

In numerous fields relying on texture analysis, run-length distributions have gained in popularity significantly. For example, in handwriting analysis, such as writer identification and signature verification, they are so effective. They're also quite trustworthy in biomedical imaging, such as radiography, CT, and even MR pictures.

The run-length features concept relies on dividing the signature into a series of runs. A 'run' is a collection of pixels with the same gray level value that are consecutively arranged in a line. The number of pixels in the run determines the length of the run. We can compute a run length matrix for runs in any direction for a given picture (Galloway, 1975). The run-length matrix is defined as a matrix P where the value at position (i, j) in the matrix represents the number of pixel runs of color i and length j in a given direction. The size of the matrix is $M \times K$ where M represents the number of unique colors (intensities) in the image while K is the maximum possible length of a run in a given direction. In the run length study, we consider the horizontal, vertical, left-diagonal and right-diagonal run-lengths.

The extraction of the run-length distribution features is illustrated for a 5×4 image in Fig. 2.1 having four gray levels (0-3) and the resulting gray level run-length matrices for the four principal directions.

The number of runs of different lengths that occur in the four directions is represented by the elements of the matrices in Fig. 2.2. For example, the elements in the first row of the horizontal run-length matrix (0°) indicate that for the pixel with value 0, there are 3 runs of length 1, no runs of length 2 and 4, and 1 run of length 3 for the pixel with a value of 0. Next, we get two runs of length 1 and one run of length 2 for the pixel with value 1; we get three runs of length 1 and one run of length 2 for the pixel with value 2. Finally, we receive three runs of length 1, two runs of length 2, and one run of length 3 for the pixel with the value 3.

0	3	3	2	2
2	1	1	3	3
0	3	2	1	3
2	0	3	3	3
1	0	0	0	3

Fig. 2.1 Gray level image.

Similarly, Fig 2.2 shows the run-length matrices for additional directions, such as 90°, 45°, and 135°. The image size determines the size of the run-length matrices.

2.2 History of run-length features

"A set of texture features based on gray level run lengths is described. Good classification results are obtained with these features on a set of samples representing nine terrain types." (Galloway, 1975). Although Galloway was the first introduced and employed the Run Length distributions for visual texture analysis, she used such a feature to classify a set of terrain samples. She defined a "run" as a set of consecutive, collinear picture points having the same gray level value and the "length" of a run as the number of picture points in the run.

The grey level run-length (GLRL) matrix includes the computation of runs having the same length and value according to a well-defined orientation in a given image. The matrix element (i, j) defines the number of times the image includes a run of length j, in the given direction, consisting of points with gray level i (or lying in gray level range i). The scan of the image using the run-length features is performed following four directions, horizontal (0°), vertical (90°), right-diagonal (45°), and left diagonal (135°); as a result, he specified the GLRL Matrices.

Galloway calculated five statistical functions; she defined them as the five original features of run-length distributions (Galloway, 1975). The number of calculations is precisely proportional to the number of pixels in the image. To compute the matrices, just two rows of picture values are required at any one time. The researcher employed functions similar to the gray level co-occurrence matrices to extract numerical texture measures from the matrices (Haralick et al., 1973).

0°	Length	1	2	3	4
Grey Level	0	3	0	1	0
	1	2	1	0	0
	2	3	1	0	0
	3	3	2	1	0

Horizontal direction 0°

90°	Length	1	2	3	4
Grey Level	0	4	1	0	0
	1	4	0	0	0
	2	5	0	0	0
	3	6	0	0	1

Vertical direction 90°

45°	Length	1	2	3	4
Grey Level	0	6	0	0	0
	1	4	0	0	0
	2	5	0	0	0
	3	8	1	0	0

Right-diagonal direction 45°

135°	Length	1	2	3	4
Grey Level	0	3	0	1	0
	1	2	1	0	0
	2	5	0	0	0
	3	3	2	1	0

Left-diagonal direction 135°

Fig. 2.2 Run-length matrixes.

Let $p(i, j)$ be the (i, j)th entry in the given run length matrix; Ng, be the number of gray levels in the image, Nr be the number of different run lengths that occur (so that the matrix is Ng by Nr), and P be the number of pixels in the image.

- Short Runs Emphasis (SRE)

$$SRE = \frac{\sum_{i=1}^{Ng} \sum_{j=1}^{Nr} \frac{P(i,j \vee \theta)}{j^2}}{N_z(\theta)} \tag{2.1}$$

  The objectif of this function is to emphasize short runs, it multiplies each run-length value by the run's squared length devided by the total number of runs in the image as a normalizing factor.

- Long Runs Emphasis (LRE)

$$LRE = \frac{\sum_{i=1}^{Ng} \sum_{j=1}^{Nr} P(i, j \vee \theta) j^2}{N_z(\theta)} \tag{2.2}$$

  Similar to the previous function, the LRE function emphasizes the long runs, by multiplying the length of the run squared by each run-length value. The denominator is a normalizing factor, as above.

- Gray Level Nonuniformity (GLN)

$$GLN = \frac{\sum_{i=1}^{Ng} \left(\sum_{j=1}^{Nr} P(i, j \vee \theta) \right)^2}{N_z(\theta)} \tag{2.3}$$

  This function aims to determine the image gray level nonuniformity. It squares the number of run lengths for each gray level. Then, the total number of runs in the image is divided by the sum of the square's normalizing factor.

- Run Length Nonuniformity

$$RLN = \frac{\sum_{j=1}^{Nr} \left(\sum_{i=1}^{Ng} P(i, j \vee \theta) \right)^2}{N_z(\theta)} \tag{2.4}$$

  This function measures the nonuniformity of run lengths; it twill return a low result if the runs are evenly sprereas throughout the lengths. It squares the number of runs for each length.

- Run Percentage

$$RP = \frac{N_r}{N_p} \tag{2.5}$$

If all runs had a length of 'one,' the run percentage function would operate as a ratio of the total number of runs to the total number of possible runs.

In a new study; (Chu et al., 1990) added two functions to the original ones. They stated that the five principal functions are defined by r(j) the total number of runs of length j of all possible gray values. Since for a given value of r(j), the composition of runs of different gray values can vary. Thus, the GLN measures the power of the distribution but connot detect the possible variation in the shape of a distribution of given power. Consequently, the gray level information of the image cannot be well presented by the run-length features.

The researchers introduced two new measures called Low Gray Level Run Emphasis LGRE and High Gray Level Run Emphasis HGRE; not to use only the number of runs but also the gray values associated with them.

$$LGRE = \frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Nr}\frac{P(i,j\vee\theta)}{i^2}}{N_z(\theta)} \tag{2.6}$$

$$HGRE = \frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Nr}P(i,j\vee\theta)i^2}{N_z(\theta)} \tag{2.7}$$

(Dasarathy and Holder, 1991) offered four other features in the same context, namely Short Run Low Gray-level Emphasis (SRLGE), Short Run High Gray-level Emphasis (SRHGE), Long Run High Gray-level Emphasis (LRHGE), and Long Run Low Gray-level Emphasis (LRLGE).

The utility of these new functions is to emphasize the combined distribution qualities of run lengths and gray levels rather than the individual ones. As a result, the new features can be employed in place of the older ones with no loss of information content and increased resilience.

$$SRLGE = \frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Nr}\frac{P(i,j\vee\theta)}{i^2j^2}}{N_z(\theta)} \tag{2.8}$$

$$SRHGE = \frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Nr}\frac{P(i,j\vee\theta)i^2}{j^2}}{N_z(\theta)} \tag{2.9}$$

Fig. 2.3 Terrain samples used in texture analysis study (Galloway, 1975).

$$LRLGE = \frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Nr}\frac{P(i,j\vee\theta)j^2}{i^2}}{N_z(\theta)} \tag{2.10}$$

$$LRHGE = \frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Nr}P(i,j\vee\theta)i^2j^2}{N_z(\theta)} \tag{2.11}$$

Another improvement to the typical run-length features was performed by (Tang, 1998). The study added a multilevel dominant eigenvector estimation approach that preserves much of the texture information in run-length matrices and improves image classification performance greatly over typical run-length features (Tang, 1998). The novel method produced accurate image classification and discriminatory information extraction.

To preserve all information in the matrix, the method relies on directly using the run-length matrix as the texture feature vector rather than constructing additional methods to extract texture information.

More contributions were interested in the improvement of run-length distribution. (Albregtsen and Nielsen, 2000) Constructed class distance matrices for the run-length features. The researchers discovered areas of consistently high values in the class distance matrices for a four-class problem of liver cell nuclei. They utilized the information from the normalized

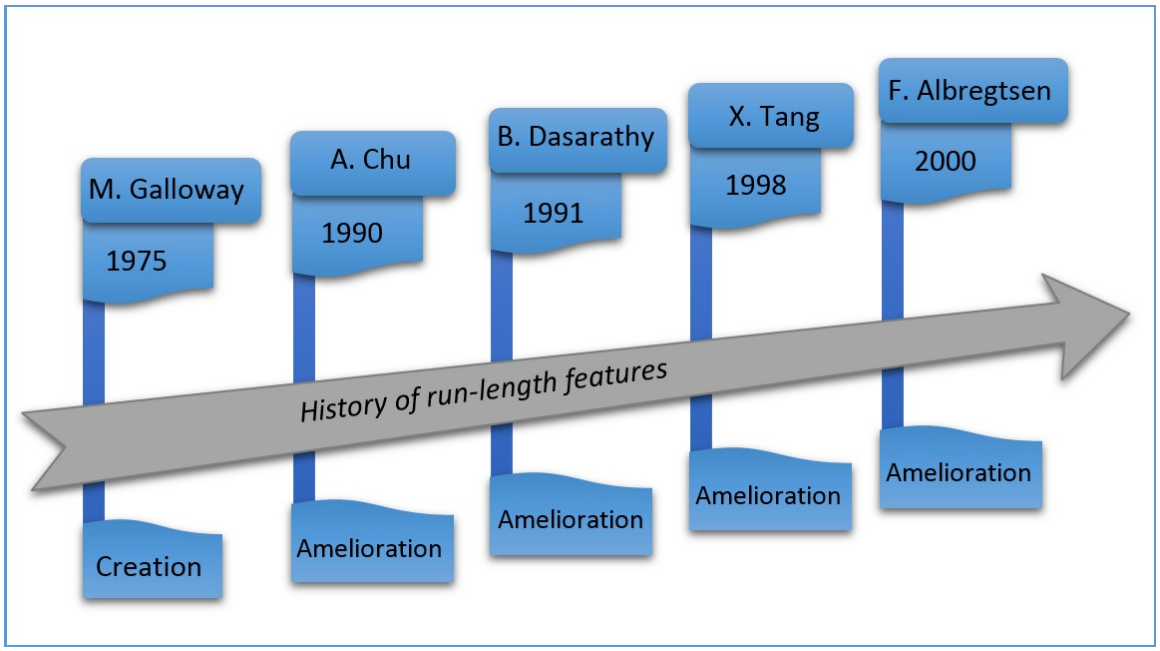


Fig. 2.4 Run-length features update.

run-length matrix entries to produce adaptive features for texture classification using the class distance matrices.

They only extracted two features using this technique, halving the classification error relative to the best pair of classical GLRLM features. The first order entropy indicates the homogeneity of the image histogram, and the run length entropy describes the irrugalrity of the pixel context.

$$\varepsilon = \sum_{i=1}^{G} \sum_{j=1}^{R} P(i,j) log_2 \left[P(i,j)\right], P(i,j) > 0 \tag{2.12}$$

The feature ε would be associated with a value-dependent weight.

$$W_{epsilon}(i,j) = log_2 \left[P(i,j)\right]. \tag{2.13}$$

2.2.1 Gray Level Run Length Matrix (GLRLM) functions description

The tables 2.1 and 2.2 describe the gray level run length matrix (GLRLM) features that quantifies gray level runs, which are defined as the length in number of pixels, of consecutive pixels that have the same gray level value. In a gray level run length matrix $P(i,j|\theta)$, the (i,j)th element describes the number of runs with gray level i and length j occur in the image (ROI) along angle θ.

Let: Ng be the number of discreet intensity values in the image.

Nr be the number of discreet run lengths in the image. Np be the number of voxels in the image $Nz(\theta)$ be the number of runs in the image along angle θ, which is equal to:

$$\sum_{i=1}^{Ng}\sum_{j=1}^{Nr} P(i,j\vee\theta), \qquad \text{and} \qquad 1 \leq N_z(\theta) \tag{2.14}$$

$P(i,j|\theta)$ be the run length matrix for an arbitrary direction θ and $p(i,j|\theta)$ be the normalized run length matrix, defined as,

$$p(i,j|\theta) = \frac{P(i,j\vee\theta)}{N_z(\theta)} \tag{2.15}$$

Table 2.1 Run-length functions *Part I*

Team, Year	**Feature name**	**Formula**	**Description**
(Galloway, 1975)	Short Run Emphasis	$SRE = \frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Nr}\frac{P(i,j\vee\theta)}{j^2}}{N_z(\theta)}$	Emphasizes short runs.
	Long Run Emphasis	$LRE = \frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Nr}P(i,j\vee\theta)j^2}{N_z(\theta)}$	Emphasizes long runs.
	Gray Level Non Uniformity	$GLN = \frac{\sum_{i=1}^{Ng}\left(\sum_{j=1}^{Nr}P(i,j\vee\theta)\right)^2}{N_z(\theta)}$	Measures the gray level nonuniformity of the image.
	Run Length Non Uniformity (RLN)	$RLN = \frac{\sum_{j=1}^{Nr}\left(\sum_{i=1}^{Ng}P(i,j\vee\theta)\right)^2}{N_z(\theta)}$	Measures the run-length nonuniformity in the image.
	Run Percentage	$RP = \frac{N_r}{N_p}$	Computes the overall number of runs to the total number of possible runs if all runs had a length of one. Values are in range $1/N_p < RP < 1$; with higher values indicating a larger portion of the ROI consists of short runs (indicates a more fine texture).
(Chu et al., 1990)	Low Gray Level Run Emphasis	$\frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Nr}\frac{P(i,j\vee\theta)}{i^2}}{N_z(\theta)}$	This function highlights the low gray level runs in the image.
	High Gray Level Run Emphasis	$\frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Nr}P(i,j\vee\theta)i^2}{N_z(\theta)}$	This function highlights the high gray level runs in the image.

Table 2.2 Run-length functions *(Part II)*

Team, Year	**Feature name**	**Formula**	**Description**
(Dasarathy and Holder, 1991)	Short Run Low Gray Level Emphasis	$\frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Nr}\frac{P(i,j\vee\theta)}{i^2 j^2}}{N_z(\theta)}$	Assesses the joint distribution of shorter run lengths with lower gray-level values.
	Short Run High Gray Level Emphasis	$\frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Nr}\frac{P(i,j\vee\theta)i^2}{j^2}}{N_z(\theta)}$	Assesses the joint distribution of shorter run lengths with higher gray-level values.
	Long Run Low Gray Level Emphasis	$\frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Nr}\frac{P(i,j\vee\theta)j^2}{i^2}}{N_z(\theta)}$	Assesses the joint distribution of longer run lengths with lower gray-level values.
	Long Run High Gray Level Emphasis	$\frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Nr}P(i,j\vee\theta)i^2 j^2}{N_z(\theta)}$	Assesses the joint distribution of longer run lengths with higher gray-level values.
(Albregtsen and Nielsen, 2000)	Run Entropy	$\sum_{i=1}^{G}\sum_{j=1}^{R}P(i,j)log_2[P(i,j)]$	Measures the uncertainty/ randomness in the distribution of run lengths and gray levels.

2.3 Run-length features in medical imaging

The run-length distributions were used for image classification in Magnetic Resonance Imaging (MRI). (Derea et al., 2019) presented a new approach for the detection and diagnosis of the brain cancer. The GLRLM technique was applied on the grayscaled images, and the image classification was performed using the segmentation based threshold; separating the pixels with high gray-level from pixels with low grey-level values. The tumour is detected by applying morphological processes and determening the number of objects. Then, the largest object of the image is localized to detect the tumour in the original image. The use of run-length features could give accurate results.

In the realm of Hepetocellular carcinoma (HCC) and liver abscess, (Xu et al., 2019) examined the use of computer-aided diagnosis (CAD) to discriminate between hepatocellular carcinoma (HCC). The researchers have chosen the ultrasound imaging (US) method among the other imaging techniques such as Computed Tomography (CT) and magnetic resonance imaging (MRI), du the advantage of this method as no radiation, easy operation and low cost. The research was based on image processing and machine learning techniques. For the textural analysis, two most effective methods were used in the feature extraction phase, the gray level run-length matrix (GLRLM) and the gray level co-occurrence matrix (GLCM), where 96 features were extracted including 52 features of GLCM and 44 features of GLRLM.

For the classification phase was carried on using the Support Vector Machine (SVM) due to its high accuracy rate. The obtained results show that the proposed models using SVM and GLRLM techniques get the highest accuracy of 87.125 % while the use of GLCM get an accuracy of 77.25 %. The final results using all the features provided an accuracy of distinguishing HCC from liver abscess with ACC up to 89.25 %.

The classification findings were compared using several texture analysis methods in extracting a characteristic of the mammography image in another medical study (Novitasari et al., 2019). Some texture analysis approaches have effectively extracted features based on their characteristics, such as first-order, GLCM, GLRLM, and GLDM. The statistical features of these methods were fed into the ECOC SVM classifier, which was built using three kernel comparisons: linear, RBF, and polynomial. Polynomial kernels with statistical features produced by GLRLM were found to be the best kernel, with a 93.98 percent accuracy value.

In the same context, Analysing medical images; (Mudda et al., 2020) contributed to the precise detection of malignancies. As a result, the requirement for a tumor detection technology outweighs the scarcity of qualified radiologists. The diagnosis and location of brain cancers are made possible by biomedical image processing employing Magnetic Resonance Imaging (MRI). Their research created a method for segmenting and detecting brain tumors that used MRI sequence images as input images to determine the tumor region. However, the vast differences hamper the procedure in tumor tissues between patients and the similarities in normal tissues. Their primary purpose was to determine whether the brain had a brain tumor or was healthy. For the Region of Interest(ROI) of the tumor region, the region-growing segmentation algorithm provides accurate boundaries. In feature extraction, the Gray Level Run Length Matrix (GLRLM) and Center-Symmetric Local Binary Patterns(CSLBP) texture features are combined for efficient brain tumor detection and classification, Adopting Neural N. Finally, the suggested method's experimental results are compared to the accuracy of several algorithms.

2.4 Run-length features in writing identification

Biometric recognition (Chawki and Labiba, 2010; **?**), personalized handwriting recognition (Nosary et al., 2004), automatic forensic document examination (Erp et al., 2003), classification of ancient manuscripts (Siddiqi et al., 2009), and smart meeting rooms are some of the applications of writing identification and writer recognition systems (Liwicki et al., 2006). These applications have sparked fresh scientific interest in the field in recent years.

A study of feature combinations was conducted by (Bulacu and Schomaker, 2007). Fusing several features (directional, grapheme, run-length PDFs) yields improved writer identification and verification performance, according to results obtained on a large dataset encompassing 900 writers. Furthermore, the average distances owing to the individual features involved in the fusion process are used to get the final unique distance between two handwritten samples. The features capture many aspects of handwriting originality and work on several degrees of analysis and scales. When textural and allographic traits are combined, excellent writer identification and verification results are obtained. The proposed fusion approach, which is based on simple distance averaging, reduces the danger of a partial solution while capturing the majority of the potential improvements in writer identification and verification performance.

Another research (Bulacu et al., 2007)studied the performance of text-independent writer identification algorithms developed and tested on the Western script in recent years on Arabic handwriting. The experiments detailed here used the IFN/ENIT data, and 350 writers participated in the testing. The findings suggest that the methodologies are pretty effective and that the conclusions reached in prior investigations apply equally well to Arabic script. Textural features (joint directional probability distributions) are combined with allographic features to obtain high performance (grapheme-emission distributions).

The contourbased angle-combination PDFs (f2, f3h, f3v) and the grapheme emission PDF are the gist of the text-independent approach to writer identification and verification (f4). As previously mentioned, these cutting-edge capabilities surpass other text-independent methods. Combining textural and allographic traits for hundreds of writers' datasets produces extremely high writer identification rates. The findings of prior studies on Western script have also been confirmed in Arabic handwriting. The statistical methods are broad in nature and produce reliable and consistent results.

(Chawki and Labiba, 2010) proposed run-length distribution as a global approach for texture analysis for Arabic writer identification and verification. The texture classification approach is adopted by considering each writer's handwriting as a distinct texture, based on a group of new features that have been developed and retrieved from GLRL matrices (Gray Level Run Length) and GLCM (Gray Level Co-occurrence Matrices). The method is applied to 650 Arabic handwriting images from 130 writers and contains feature extraction and classification steps.

The first step employs two techniques: the GLCM and the GLRL matrices. In contrast, the classification step is performed for the writer verification task by computing the Receiver Operating Characteristic (ROC) curves for all features and employing the Equal Error Rate (EER) as a performance measure. While, for the writer identification task, the classification

phase is performed using the nearest-neighbor classification by choosing one of 650 pages. It then computes the Euclidean distances between the feature vector of the chosen page and the feature vector of the remaining 649 pages.

This approach was based on the extraction of global features and was adopted on text-independent analysis; its implementation and the new set of run-length texture features could significantly improve the writer identification and verification findings.

In the same area, A text-independent writer recognition of Arabic handwritten documents was introduced in another study by (Djeddi et al., 2012). Run-length features and edge hinge features were used to classify 1375 handwritten documents from 275 distinct Arabic writers that were added to the IFN/ENIT database. The four main scanning directions—horizontal, vertical, left-diagonal, and right-diagonal—were used to apply the run-length features on binary images by counting black-and-white pixels. The nearest-neighbor classification in a leave-one-out strategy was employed to evaluate the features for the writer identification task, adhering to the same protocol as the precedent approach. Roc curves are calculated for the writer verification, and the EER metric gauges the system performance.

The results show that the run-length features have adequate discriminatory information and that combining the different directions followed in the scan by counting the black and white pixels can further increase the classification yield.

(Djeddi et al., 2013) introduced a writer recognition method in a multi-script environment in another work using different writing scripts to identify the writer of a handwritten text in one defined script from a sample of the same writer in another script, thereby confirming the idea that an individual's writing style is consistent across different scripts.

The run-length features are computed on the full image without text segmentation into characters. The binary image is first scanned in the four main orientations used on the black and white pixels. After that, a single vector identifying the document's writer is created by concatenating the four main run-length matrices already converted into vectors. The first 100 columns are maintained for the black pixels, and the first 50 columns are retained for the white pixels for each of the four directions, creating a complete vector of 600 values for each writing.

The method employed K-Nearest Neighbors (K-NN) and Support Vector Machine (SVM) for classification and Greek and English databases containing 126 writers with four examples. The run-length features in this investigation outperformed the state-of-the-art features, demonstrating their efficacy in a multi-script environment.

(He and Schomaker, 2016) proposed a general pattern run-length transform for writer identification which counting the runs of the complex patterns and can be used on the binary images or on the gray scale images. The proposed methods are more discriminative than the

traditional run-length method. They used the proposed method for writer identification on four public data sets and experimental results have demonstrated that their proposed method outperformed state-of-the-art approaches on the challenging CERUG-EN data set.

The General Pattern Run-Length Transform (GPRLT), which is the histogram of the run-length of any complex patterns. The GPRLT can be computed on the binary images (GPRLTbin) or on the gray scale images (GPRLTgray) without using any binarization or segmentation methods. Experimental results have shown that the GPRLTgray achieves even higher performance than the GPRLTbin for writer identification.

The proposed DST approach combined three information sources, edge-hinge with fragment length of 6 and 7 pixels and run-length features, where each SVM classifier works on different features. They started by converting the output of SVM classifiers into probability distributions. The second step consists in building dynamic frame of discernment. They converted the probabilistic output of each of our classifiers into a mass function and combine them. Finally, they used the pignistic transform as a decision rule to obtain the final list of writers.

The same team presented another work in (He and Schomaker, 2017), where they introduced two novel curvature-free features: the run-lengths of Local Binary Pattern (LBPruns). They can be used on binarized images and grayscale images and the Cloud Of Line Distribution (COLD), which is the distribution of the line segments from the contours of handwritten texts in the polar coordinate space quantized into a log-polar histogram.

The LBPruns is the joint distribution of the traditional run-length and local binary pattern (LBP) methods, which computes the run-lengths of local binary patterns on both binarized images and gray scale images. The COLD feature is the joint distribution of the relation between orientations and lengths of line segments obtained from writing contours in handwritten documents. The proposed LBPruns and COLD are textural-based curvature-free features and capture the line information of handwritten texts instead of the curvature information. The combination of the LBPruns and COLD features provides a significant improvement on the CERUG data set, handwritten documents on which contain a large number of irregular-curvature strokes. The proposed features evaluated on other two widely used data sets (Firemaker and IAM) demonstrate promising results. Experimental results show that the proposed methods provide very good performance on irregular-curvature handwriting. They have introduced two novel curvature-free features: the run-lengths of Local Binary Pattern (LBPruns) which is the run-lengths histogram of local binary patterns and can be used on binarized images and gray scale images, and the cloud of line distribution (COLD) which is the distribution of line segments from contours of handwritten texts in the polar coordinate space and it is quantized into a log-polar histogram.

The researchers have visualized the COLDs on both historical documents and natural images. They have shown that the COLD can capture the line structures on images which can be used, in future, for historical document retrieval and scene classification.

Examining the same issue of writer identification; (Kessentini et al., 2018) performed a work that proposed a new approach for offline writer identification based on a combination of SVM classifiers. The main contribution of this study was to propose a combination module using Dempster-Shafer Theory (DST) in an attempt to improve the overall system performance. The DST used in this contribution is an effective theoretical framework to treat uncertainty and imprecision related to information sources. The evaluation of the proposed system was carried on different publicly available databases on Arabic and Latin scripts. Experimental results reveal that the proposed combination approach outperforms the conventional combination methods and achieves interesting results as compared to those reported by the existing writer recognition systems.

The paper of (Hannad et al., 2019) investigated the problem of writer identification from handwriting samples in Arabic. The proposed technique relies on extracting small fragments of writing, characterized using two textural descriptors, Histogram of Oriented Gradients (HOG) and Gray Level Run Length (GLRL) Matrices. Several fusion rules combine similar scores realized using HOG and GLRL features. The system is evaluated on three well-known Arabic handwriting databases, the IFN/ENIT database with 411 writers, the KHATT database with 1000 writers, and the QUWI database with 1,017 writers. The results on the KHATT database are comparable to state-of-the-art, while those reported on the IFN/ENIT and QUWI databases are the highest.

Finally, the table 2.3 summarizes the main contributions.

Table 2.3 Writer identification systems using run-length features

Study	**Features**	**Number of writers**	**Writer identification Result**
(Bulacu et al., 2003)	Vertical run-length PDF	150	61.00 %
	Horizontal run-length PDF	150	66.00 %
(Bulacu et al., 2007)	Run-length, Combination of textural features	350	89.00 %
(Djeddi et al., 2012)	Run-length, Small writing fragments	33	93.93 %
(Djeddi and Souici-Meslati, 2011)	Run-length & GLCM matrices	130	82.62 %
(Djeddi et al., 2014)	run lengths, edge-hinge & edge-direction features	130	82.62 %
(Hannad et al., 2019)	GLRLwhite & GLRLblack & HOG	411	94.16 %

2.5 Automatic signature verification and Run-length features

Despite tremendous development in different biometric modalities, signatures continue to be the most widely accepted authentication mechanism in legal documents and financial transactions worldwide. However, automatic signature verification has remained a challenging pattern classification problem for several decades (Leclerc and Plamondon, 1994; Plamondon and Lorette, 1989; Plamondon and Srihari, 2000). Recent advances on this problem have been summarized in a number of survey papers (Diaz et al., 2019; Hafemann et al., 2017c).

2.5.1 Automatic Signature verification systems

Signature verification can be performed using a writer-dependent approach where a separate classifier is trained for each writer or a writer-independent approach where a single classifier is trained on genuine and forged signatures of all individuals in the database under study (Srihari et al., 2004). Signature verification techniques are distinguished into online and offline methods from the acquisition process. Online signatures are acquired on a digitizer which normally captures the trajectory, time, and pressure information while the signature is performed (Cpalka and Zalasinski, 2014; Guru et al., 2017; Maiorana, 2010). Offine signatures are images of signatures (Aubin and Mora, 2017; Justino et al., 2001; Serdouk et al., 2016) acquired through a scanner or a camera and made the subject of our study. Offline signature verification methods are further classified into static and pseudo-dynamic techniques. Static techniques rely on extracting statistical or structural features to capture the signature image's unique characteristics. Examples include geometrical (**?**) or textural measures (Ferrer et al., 2012) extracted from signature images. The pseudodynamic techniques attempt to extract dynamic properties from offline images. Examples of such features include High-Pressure Points (HPP), thickness and progression of strokes and their variations, distribution of pixels, etc. (Justino et al., 2000).

Signature modeling has been effectively carried out using hidden Markov models (HMM) (Van et al., 2007) and graph models (Wang et al., 2011). For matching, Dynamic Time Warping (DTW) (Fischer et al., 2015; Nanni et al., 2010) have been one of the most employed techniques for function- based features, while Support Vector Machines (SVMs) have been influential on parameter features (Piekarczyk, 2010).

The key task in a signature verification system is to decide whether a given signature image is genuine or forged; a two-class pattern classification problem. The typology of the forgeries, however, has remained a matter for debate in the scientific literature (Justino et al.,

2001; Malik, 2015). In an interesting study by (Malik, 2015), authors discuss the gap between the pattern recognition community and the forensic handwriting experts in terms of evaluation of signature verification systems. From the view point of pattern recognition community, authors categorize forgeries into four classes. These include: Random Forgery (any random signature other than genuine author), Simple Forgery (forger only has the knowledge of the name of the genuine signer), Simulated Forgery (signatures are forged by inexperienced forger after practice) and Skilled Forgery (signatures are forged by experienced forger after practice). From the view point of forensic experts, (Malik, 2015) identify Disguised Signatures (a genuine author makes look his/her own signature a forgery), Simple Forgery (forged without practice) and Skilled Forgery (forged after practice) as the acceptable forgery classes. Authors recommend that random forgeries should be distinguished from other types of forgeries (accepted by forensic experts) and should not affect the overall evaluation of the system.

Offine signature verification systems have been extensively researched over the last few decades from many applications especially legal validation of documents in forensic examinations. Automatic Signature Verification (ASV) systems aim to distinguish between genuine and forged signatures by enhancing the feature extraction and classification techniques, the two key components of any pattern classification system. Among well-known contributions to this problem,

Convolutional Neural Networks (CNNs) were employed in an interesting study (Zheng et al., 2021) to extract micro deformations in the max-pooling process for offline signature verification tasks. The location coordinates of the maximum values in pooling windows of max-pooling can be used to determine micro deformations. Extensive study and testing show that adding position information as a new feature for collecting micro deformations in conjunction with convolutional features can achieve state-of-the-art performance. The suggested technique made use of four publically available datasets in four different languages: English (GPDSsynthetic, CEDAR), Persian (UTSig), and Hindi (UTSig) (BHSig260). (Avola et al., 2021) offered a multi-task technique to learning a signature representation in smaller feature space using a relaxed loss by exploiting a limited generic feature space. The R-SigNet system has been successfully used to verify offline WI signatures. This network extracts compact generic features automatically so that a support vector machine (SVM) can be trained and tested in offline writer-dependent (WD) mode.

For the first time, offline signature verification literature (Zois et al., 2020) introduced their use as a parameter-free, agnostic representation for exploring global and local information. Global properties of the sparsely located content of the shape of the signature image are encoded with topological information of the whole graph. In addition, local pixel patches are

encoded by sequential visibility motifs-subgraphs of size four to a low six-dimensional motif profile vector. Finally, a number of pooling functions operate on the motif codes in a spatial pyramid context in order to create the final feature vector. The effectiveness of the proposed method is evaluated with the use of two popular datasets. The local visibility graph features are considered to be highly informative for signature verification.

Using Binary Particle Swarm Optimization (BPSO), (Souza et al., 2020) investigated the presence of overfitting when to perform the feature selection in a context of Handwritten Signature Verification (HSV). SigNet is a state of the art Deep CNN model for feature representation in the HSV context and contains 2048 dimensions. Some of these dimensions may include redundant information in the dissimilarity representation space generated by the dichotomy transformation (DT) used by the writer-independent (WI) approach. The analysis is carried out on the GPDS-960 dataset. A novel end-to-end network was introduced by (Lu et al., 2021), named Cut and-Compare, which can learn discriminative and informative regions automatically by a modified STN model and the regions are compared by an ARC model. To address the intrapersonal variability problem, they designed a smoothed double-margin loss to train the network. The proposed cut-and-compare network and smoothed double-margin loss are shown to be effective in experiments, it is achieved on several datasets, including CEDAR, GPDS Synthetic, BHSig-H and BHSig-B, which are of different languages.

(Maergner et al., 2018a) introduce two structural methods for offline signature verification. The methods were proposed for handwriting analysis, for which efficient matching methods were available: keypoint graphs with approximate graph edit distance and inkball models. Inkball models, in particular, have never been used for signature verification before. They investigated both approaches individually and proposed a combined verification system performed on the MCYT and GPDS benchmark data sets. In the same context, (Maergner et al., 2018b) proposed to complement the recent structural approach to offline signature verification based on graph edit distance with a statistical approach based on metric learning with deep neural networks. The MCYT and GPDS benchmark datasets demonstrate that combining the structural and statistical models leads to significant improvements in performance, profiting from their complementary properties. The structural model based on approximate graph edit distance achieved better results against skilled forgeries.

In comparison, the statistical model based on metric learning with deep triplet networks achieved better results against a brute-force attack with random forgeries. In order to obtain a dissimilarity representation, (Stauffer et al., 2019) intended to use different graph embedding approaches in conjunction with a recent graph-based signature verification framework. Signature graphs are not directly matched but first compared with a set of predefined prototype

graphs. They used two strategies to define these prototypes, Reference Embedding (RE) and Prototype Embedding (PE). In the case of RE, a graph is mapped into a feature space by comparing it with a user-specific set of reference graphs. In the case of PE, the mapping is based on an independent set of prototype graphs. The experimental evaluation was utilized on two datasets, GPDS-75 and MCYT-75.

A work of (Hafemann et al., 2019) investigate the impact of adversarial examples on biometric systems, in particular by identifying threats to Offline Handwritten Signature Verification under the point of view of Adversarial Machine Learning. They characterized this phenomenon under an existing taxonomy of threats to biometric systems, in particular identifying new attacks for Offline Handwritten Signature Verification systems. They conducted an extensive set of experiments on four widely used datasets: MCYT-75, CEDAR, GPDS-160 and the Brazilian PUC-PR, considering both a CNN-based system and a system using a handcrafted feature extractor (CLBP). They found that attacks that aim to get a genuine signature rejected are easy to generate, even in a limited knowledge scenario, where the attacker does not have access to the trained classifier nor the signatures used for training. Attacks that get a forgery to be accepted are harder to produce, and often require a higher level of noise in most cases, no longer "imperceptible" as previous findings in object recognition. They also evaluated the impact of two countermeasures on the success rate of the attacks and the amount of noise required for generating successful attacks.

Another study (Souza et al., 2018) introduces an approach for writer-independent offline signature verification that uses the dissimilarity representation of the deep CNN features and a single SVM as a writer-independent classifier to authenticate handwritten signatures. This work investigated whether the use of these CNN features provides good results in a writer-independent (WI) HSV context, based on the dichotomy transformation combined with the use of an SVM writer-independent classifier. The experiments were performed on the Brazilian and GPDS datasets.

(Guerbai et al., 2015) propose a writer-independent signature verification system using curvelet transform and One- Class Support Vector Machine (OC-SVM). Models trained using only genuine signatures realized promising results on CEDAR and GPDS signatures. Likewise, (Zois et al., 2016) propose to employ grid- based template matching scheme with SVM for signature verification. The method evaluated on four different signature databases reported state-of-the-art results.

In another interesting work, (Okawa, 2016b) attempt to mimic the cognitive processing of forensic experts to extract features for verification of signatures. The technique relies on vector of locally aggregated descriptors (VLAD) with KAZE features and is evaluated on the CEDAR signature database realizing an error rate of 1.0 %. Similar work by same

author (Okawa, 2016a) adapts a bag- of-visual-words (BoVW) model with KAZE features reporting an error rate of 1.6 % on the CEDAR dataset. Another recent work by (Dutta et al., 2016) exploits a combination of visual codebooks generated using BRISK points. Each point is represented by the histogram of oriented gradients as descriptor. The first codebook represents local features of order one whereas the second type paris the local features based on the graph edges created upon a Delaunay triangulation. Experiments on CEDAR and GPDS300 datasets report error rates of 0 % and 11.21 % respectively. An investigation of sparse dictionary learning and coding to serve as feature space for offline signatures is carried out by (Zois et al., 2017). Authors employ the K-SVD dictionary learning followed by an Orthogonal Matching Pursuit algorithm and report results on three well-known signa ture databases (CEDAR, MCYT75 & GPDS300). Authors also introduced archetypal analysis for signature verification (Zois et al., 2017). Such an analysis creates a signer model based on a set of learned archetypes obtained from the training samples. The effectiveness of the proposed approach is demonstrated through experiments on CEDAR and MCYT75 databases.

Among other recent studies, (Das et al., 2016) compared the performance of single-script and multi-script scenarios for signature verification using nine different databases in five scripts. Another interesting work (Diaz et al., 2016a) presents a technique to duplicate signatures utilizing a set of nonlinear and linear transformations simulating the human spatial cognitive map and motor system during the signing process. The technique is evaluated by increasing the training data using the duplicator and evaluating four state-of-the-art signature verification algorithms. In another notable contribution, Artificial Immune System (AIS) is proposed for offline signature verification (Serdouk et al., 2016) using gradient and topological features. Experiments on CEDAR and GPDS-100 datasets realized promising results.

Several studies have investigated verification techniques based on deep learning in recent years. (Soleimani et al., 2016) for instance, employed Histogram of Oriented Gradients (HOG) and Discrete Radon Transform (DRT) with Deep Multitask Metric Learning (DMML) to enhance the signature verification performance. (Hafemann et al., 2016) proposed a writer-independent approach based on Deep Convolutional Neural Networks (DNN) to learn features. The main idea is to learn features in a writer-independent mode and train writer-dependent classifiers. The technique evaluated on GPDS960 and Brazilian PUC-PR databases realized encouraging results. In another recent study, (Rantzsch et al., 2016) propose a writer-independent approach based on deep metric learning to learn embedding signatures into a high-dimensional space. This method compares triplets of two genuine and one forged signature in order to enhance the verification performance on the ICDAR SigWiComp

2013 challenge database. Likewise, (Hafemann et al., 2017a) applied Convolutional Neural Networks to learn effective representations from signature images in a writer-independent mode. The technique involves learning features from a sub-set of skilled forgeries. The system evaluated on four datasets (GPDS, MCYT, CEDAR, and Brazilian PUC-PR) reported competitive performance in terms of equal error rate. A summary of notable contributions to offline signature verification discussed in the preceding paragraphs is presented in Table 4. A critical aspect of signature verification systems is the number of signatures in the training set.

For most of the practical applications, the number of reference signatures is fairly limited and in many cases only a single specimen per individual is available making it difficult to infer the intra-signature variations. This issue has been addressed in few of the recent studies on online signature verification (Diaz et al., 2016c, 2015; Galbally et al., 2009). In case of offline signature verification systems, we can identify the ICDAR 2009 Signature Verification Competition (Blankers et al., 2009) where the participants were required to design Single Reference Signature Systems (SRSS). The winning system of the competition reported an EER of 9.15 % while rest of systems realized error rates of more than 15.00 % showing that SRSS are much more challenging and require considerable research investigations to realize the error rates acceptable for practical applications. The organization of similar competitions on signature verification (Blumenstein et al., 2010; Liwicki et al., 2012; Malik et al., 2015, 2013) has become a regular activity in conjunction with ICDAR and ICFHR, the two most notable platforms for document analysis and recognition community. The increasing number of participants in these competitions from all over the world speaks of the tremendous research attention being paid to this problem. The tables 2.4, 2.5 and 2.6 show some notable siganture verification systems published last years.

2.5.2 Signature verification systems using run-length features

Signatures have retained the most frequently accepted authentication method (Leclerc and Plamondon, 1994). For decades (Leclerc and Plamondon, 1994; Plamondon and Lorette, 1989; Plamondon and Srihari, 2000). Therefore, automatic signature verification has remained an appealing pattern classification problem. Several survey publications (Impedovo and Pirlo, 2008; Impedovo et al., 2012) have reviewed recent progress on this subject. The goal of the signature verification method is to determine if a signature is authentic or forged. The run-length distribution's effectiveness in various texture classification domains, including writing analysis and biomedical imaging, has prompted academics to apply these features to the signature verification process.

A paper of (Serdouk et al., 2014), where the authors proposes new data features to improve the off-line handwritten signature verification. The proposed features combined

Table 2.4 Summary of notable signature verification systems (*Part I)*

Study	Features	Classifier	Database	Training	Performance
(Guerbai et al., 2015)	Curvelet transform features	OC-SVM	CEDAR	4,8,12	AER: 8.70%, 7.83%, 5.60%
			GPDS-300	4,8,12	AER: 16.92%, 15.95%, 15.07%
(Zois et al., 2016)	Poset-oriented grid features	SVM	CSD1	5,10	EER: 3.92%, 0.96%
			CEDAR	5, 10	EER: 4.12%, 3.02%
			MCYT75	5,10	EER: 6.02%, 4.01%
			GPDS300	5,10,12	EER: 5.48%, 3.53%, 3.24%
(Soleimani et al., 2016)	Histogram of Oriented Gradients	Deep Multitask Metric Learning	UTSig	12	EER: 17.45%
			MCYT75	5,10	EER: 13.44%, 9.86%
			GPDSsynthetic	10	EER: 12.80%
	Discrete Radon Trans.		UTSig	12	EER: 20.28%
			MCYT75	5,10	EER: 15.89% , 10.06%
(Hafemann et al., 2016)	Deep CNN Features	SVM	GPDS-160	14	EER: 10.70%
			GPDS-300	14	EER: 12.83%
			Brazilian PUC-PR	30	EER : 4.17%
(Rantzsch et al., 2016)	Representation Learning Features	SVM	SigWiComp 2013-Offline Dutch	12	ACC: 81.76%
	DNN Features	Euclidean distance	SigWiComp 2013- Offline Japanese	12	ACC: 93.39%
(Das et al., 2016)	Geometrical features	HMM	GPDS100	5	EER: 22.50%
			MCYT100	5	EER: 19.98%
			SUSIG Visual	5	EER: 31.95%
	Zernike Moments	Euclidean Distance	GPDS100	5	EER: 35.16%
			MCYT100	5	EER: 35.51%
			SUSIG Visual	5	EER: 44.73%
	LBP	SVM	GPDS100	5	EER: 18.80%
			MCYT100	5	EER: 16.07%
			SUSIG Visual	5	EER: 28.81%
(Diaz et al., 2016b)	Textural Features	SVM	MCYT75	2,5	EER: 16.59%, 11.67%
			Bengali100	2,5	ERR: 10.67%, 6.06%
			Devanagari100	2,5	ERR: 11.88%, 9.01%

advantages of LBP histograms with a reduced size, with a topological descriptor that is called longest run features. The verification task is achieved by SVM classifiers and the performance assessment is conducted comparatively to the basic LBP descriptors. Results obtained on both GPDS 300 and CEDAR datasets show that the proposed features improve the verification accuracy while reducing the data size. The features aim was to improve the off-line handwritten signature verification. Specifically, the features take advantage from

Table 2.5 Summary of notable signature verification systems (*Part II)*

Study	Features	Classifier	Database	Training	Performance
(Diaz et al., 2017)	Geometrical features	HMM	GPDS300	2,5,8	EER: 32.01%, 27.86%, 26.60%
			MCYT75	2,5,8	EER: 19.03%, 15.27%, 12.02%
	Single Grid-based Features	Boostied Feature Selection	GPDS300	2,5,8	EER: 28.55%, 24.04%, 20.39%
			MCYT75	2,5,8	EER: 23.67%, 16.58%, 15.26%
	Texture Features	SVM	GPDS300	2,5,8	EER: 21.63%, 17.19%, 14.58%
			MCYT75	2,5,8	ERR: 16.06%, 11.90%, 9.12%
	Pose-orientated Grid Features	SVM	GPDS300	2,5,8	EER: 25.01%, 21.68%, 18.66%
			MCYT75	2,5,8	ERR: 16.50%, 14.02%, 11.57%
(Hafemann et al., 2017b)	SigNet-F	SVM	GPDS160	5,12	EER: 2.41%, 1.71%
			GPDS300	5,12	EER: 2.42%, 1.69%
			MCYT75	5,10	EER: 3.70%, 3.00%
			CEDAR	4,8,12	EER: 5.92%, 4.77%, 4.63%
			Brazilian PUC-PR	5,15,30	EER: 5.11%, 4.03%, 3.44%
	SigNet	SVM	GPDS160	5,12	EER: 3.23%, 3.63%
			GPDS300	5,12	EER: 3.92%, 3.95%
			MCYT75	5,10	EER: 3.58%, 2.87%
			CEDAR	4,8,12	EER: 5.87%, 5.03%, 4.76%
			Brazilian PUC-PR	5,15,30	EER: 2.92%, 2.07%, 2.01%
(Serdouk et al., 2016)	Gradient LBP & Longest Run	Artificial Immune Recognition	CEDAR	16	AER: 3.54%
			GPDS100	16	AER: 12.52%
(Zheng et al., 2021)	CNN features	SVM	GPDSsynthetic	5,10,12	EER: 6.78(0.37), 5.29(0.47), 4.59(0.41)%
			CEDAR	5,10,12	EER: 3.89(0.45), 2.95(0.38), 2.76(0.43)%
			UTSig	5,10,12	EER: 7.86(0.47), 6.62(0.58), 6.14(0.32)%
			BHSig260 - Bengali	5,10,12	EER: 9.87(0.34), 8.92(0.41), 8.21(0.38)%
			BHSig260 - Hindi	5,10,12	EER: 1.42(0.18), 10.53(0.45)%

Table 2.6 Summary of notable signature verification systems (*Part III)*

Study	Features	Classifier	Database	Training	Performance
(Avola et al., 2021)	R-signet	SVM	CEDAR	12	EER:0.00%
			Bengali	12	EER:0.57%
			Hindi	12	EER:0.53%
			MCYT-75	12	EER:2.25%
(Zois et al., 2020)	Average (SVV)	SVM	CEDAR	10	EER:0.51%
			MCYT-75	10	EER:1.54%
(Souza et al., 2020)	BPSO-Based Feature	SVM	GPDS-300	12	EER: 3.46 (0.08)%
(Lu et al., 2021)	Spatial Transformer Network (STN)	Attentive Recurrent Comparator (ARC)	CEDAR	1	EER: 4.34 / 0.00%
			GPDS Synthetic	1	EER:7.87%
			BHSig-H	1	EER:5.97%
			BHSig-B	1	EER:3.96%
			ChnSig	1	EER:10.21%
(Maergner et al., 2018a)	Inkball models	MCS	GPDS-75	5,10	EER: 9.42, 6.84%
			MCYT-75	5,10	EER: 13.07, 8.71%
(Maergner et al., 2018b)	Graph Edit Distance	MCS	GPDS-75	5,10	EER: 9.24, 6.49%
			MCYT-75	5,10	EER: 15.56, 9.15%
(Stauffer et al., 2019)	Graph Embedding Approach	GDM	GPDS-75	5,10	EER: 8.76, 6.36%
			MCYT-75	5,10	EER: 15.11, 9.07%
(Hafemann et al., 2019)	SigNet	SVM	MCYT-75	5	EER :5.68%
			CEDAR	5	EER:4.52%
			GPDS	5	EER:4.14%
			Brazilian PUC-PR	5	EER:2.67%
(Souza et al., 2018)	deep CNN features	SVM	Brazilian PUC-PR	30	EER:1.48 (0.44)%
			GPDS-160	12	EER:2.86 (0.24)%
			GPDS-300	12	EER:2.86 (0.24)%

the textural characterization of orthogonal combination of LBP features (OC-LBP) and the topological information that is offered by the Longest Run features. Comparatively to the basic LBP, the proposed features improved the verification accuracy while reducing the size of data. Specifically, the AER improvement was about 0.5 % for both datasets with a substantial reduction in the data size.

Exploring the effectiveness of the textural descriptor, (Djeddi et al., 2015) proposed research on offline signature verification employing the run-length features and the two-dimensional autoregressive coefficients (2D AR coefficients). The study aims to detect skilled forgery, and the run-length distribution was applied only on the black pixels for the feature

extraction phase. In the classification phase, the multi-class Support Vector Machines (SVMs) were used with the one-against-all method, employing 521 writers of the GPDS960 database. The run-length feature outcomes were promising against the state-of-the-art methods.

(Ghanim and Nabil, 2018) introduced a paper presenting an automatic off-line system for signature verification and forgery detection. The features were extracted and their effect on system recognition ability was reported. The computed features included run length distributions, slant distribution, entropy, Histogram of Gradients features (HoG) and Geometric features. Finally, different machine learning techniques were applied on the computed features: bagging tree, random forest and Support Vector Machine (SVM). It was noticed that SVM outperforms the other classifiers when applied on HoG features. The system was applied on Persian Offline Signature Data-set (UTSig) database and achieved satisfactory results in differentiating between genuine and forged signature.

The table 5.5 summarizes the precedent works description, mentioning the used databases, number of signers and the signature verification results.

Table 2.7 Signature verification systems using run-length features

Study	**Features**	**Database**	**#Signers**	**Performance**
(Serdouk et al., 2014)	OC-LBP and Longest Run Features	GPDS 300	300 signers	AER=9.75 %
(Djeddi et al., 2015)	run-length features	GPDS 960	521 signers	EER = 11.11 %
(Ghanim and Nabil, 2018)	Slant distribution, entropy, run length features and HoG features	UTSig	115 signers	ACC=80.2 %

2.6 Challenges and opportunities to run-length features

Texture analysis using run-length features has been a common approach in several areas of image processing, including texture analysis (Chu et al., 1990; Dasarathy and Holder, 1991; Galloway, 1975), among other researches in medical imaging (Ergen and Baykara, 2014).

Furthermore, run-length features were used in the identification and verification of handwriting. They were used in many works, including (Bulacu and Schomaker, 2007; Hannad et al., 2019) in the writer identification and, most notably, in the offline signature verification (Ghanim and Nabil, 2018; Serdouk et al., 2016), where the image-signature scan was based on the four main standard directions.

The previously cited works showed that run-length distributions are valuable for pattern recognition, particularly in signature verification. Despite this, the results achieved by using these features are still not impeccable. The different researches used only the four primary

directions in the signature scanning process without adding extra direction information or reinforcing the standard directions with circumferential angles to enhance the information provided during the scan. Thus, the run-length features need support for better browsing the signature.

As a result, we upgraded the notions of run-length distributions and used them for offline signature verification to increase the verification process's efficiency.

Furthermore, the signature browsing is enhanced by adding new directions to the four traditional run-length directions and strengthening each direction by its neighboring angles to contribute more extra information to the signature.

The next chapter will discuss our update and application of the run-length features on the offline signature verification.

Chapter 3

Offline signature verification system based on run-length features

3.1 Introduction

Many forensic and security experts have long prioritized signature verification and identifying genuine from forged signatures, especially when it comes to skilled forgeries. One of the processes that leads to practical outcomes in this context is the implementation of a robust algorithm that can accurately and effectively extract the features of each signature. Among these logarithms, we highlight textural descriptors, which have proven to be very beneficial in pattern recognition, such as handwriting recognition and writer identification, as well as the topic under study, handwritten signature verification. The technique we offer in our work is one of the most sturdy procedures employed in this meaning.

In addition, the classification stage, which is as critical as the feature extraction step in the Signature verification process, is one of the most important stages. Many studies have used positive and negative samples side by side in this stage to acquire a good classification and acceptable results by comparing . Admittedly, negative samples were not always available, necessitating the use of a system based solely on genuine samples for the classification stage, both to avoid the lack of forged samples on the one hand, and to simulate the real-world scenario, in which only positive samples are usually available. This commands the use of a one-class classifier, ensuring that negative sampling is not needed as well as providing valuable results. This chapter introduces a novel offline signature verification technique for detection of skilled forgeries using a writer-dependent technique. The proposed technique employs run-length histograms of binary images of signatures as features as illustrated in Fig. 5.1. Classification is carried out using One-Class Support Vector Machine (OC-SVM)

by using only genuine specimens in the training set to match the real world scenarios. Evaluations on the offline GPDS-960 signature corpus (Vargas et al., 2007) using the familiar metrics of forensic handwriting experts (FHE) realize promising results. It is also worth mentioning that based on our initial study to characterize signature using run-length features, a system based on these features was also submitted to the ICFHR2018 (Suwanwiwat et al., 2018) and ICFHR2020 (Das et al., 2020) signature verification competitions.

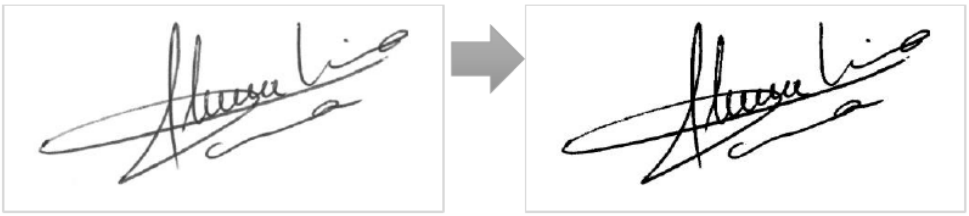


Fig. 3.1 Binarized signature (GPDS-960 Dataset) (Vargas et al., 2007).

An overview of the proposed method is presented in Fig. 3.2. It should be noted that the same series of steps (listed in Fig. 3.2) is carried out in the design and evaluation phases. First, in the design phase, the system searches for the optimal value of decision threshold using a small subset of the dataset. Then, the same steps of signature modeling and presenting the system with questioned signatures are carried out in the evaluation phase to quantify the system performance. The key contributions are summarized as follows:

- Investigation of run-length distributions to characterize signatures and detect skilled forgeries. The feature extractor considers that grayscale images may not be available in all cases and works on binary images.
- Use only of the positive specimens (genuine signatures of every individual) to train a One-Class Support Vector Machine (OC- SVM) without any forged samples in the training set matching the real-world verification scenarios.
- Study of the stability of the proposed verification system through a Single Reference Signature System (SRSS) that is evaluated using the metrics accepted by the forensic experts.

3.2 Run-length Distribution Features

This section presents the details on the run-length features employed in our study. Run length distributions of black and white pixels (Djeddi et al., 2015, 2013, 2012; Fan et al., 2012)

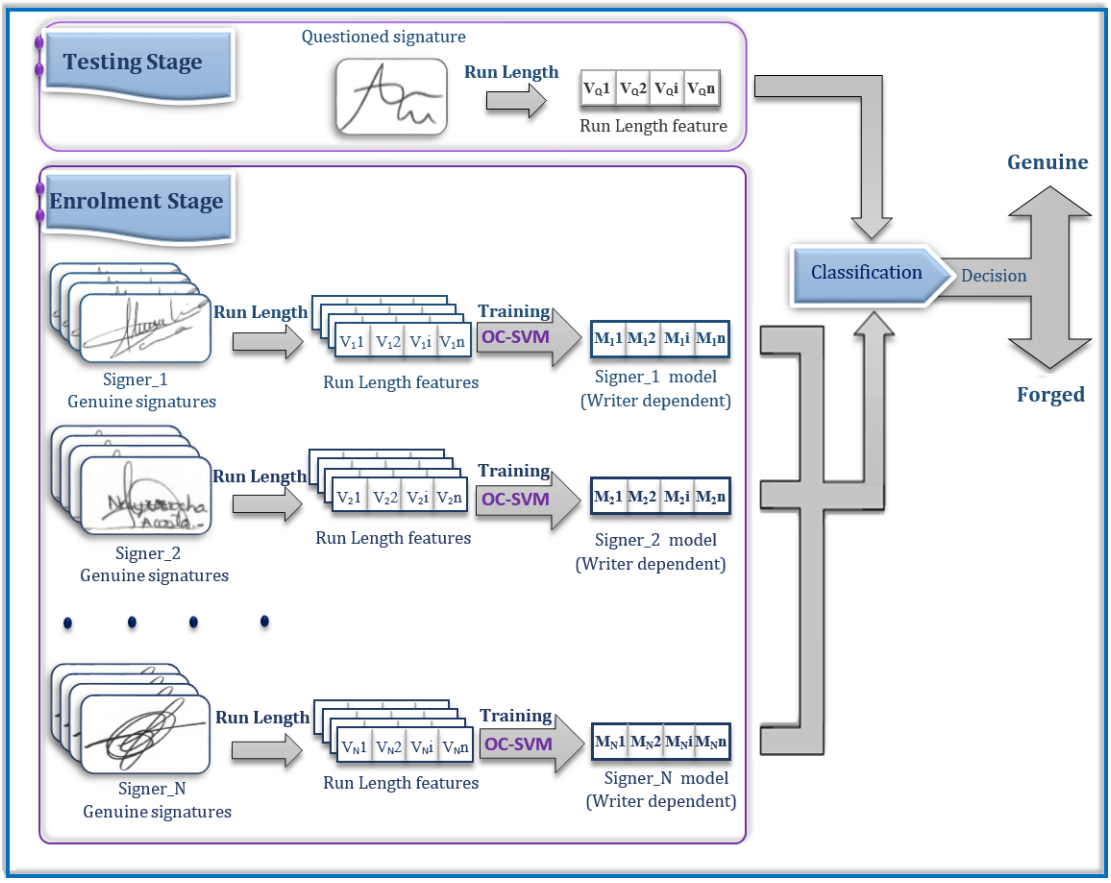


Fig. 3.2 An overview of the proposed signature verification system

have been investigated for writer identification task. They have also been employed in the writer identification competitions held in conjunction with ICDAR 2011 (Fornes et al., 2011; Hassaïne et al., 2011; Louloudis et al., 2011) and ICFHR 2012 (Hassaïne and Al Maadeed, 2012; Louloudis et al., 2012). These features realized interesting results in these competitions. The present study is intended to explore their effectiveness on the more challenging task of offline signature verification where a very limited amount of text is available as opposed to traditional writer recognition methods.

Run-length distributions are computed on binary images of signatures taking into account the black pixels which correspond to the ink trace of the signature and the white pixels which correspond to the background of the signature image (Fig. 4.3). Although grayscale images of signatures carry more information (for instance pressure), all signatures to verify my not have this information. Indeed, the more challenging scenarios in the latest competitions on signature verification employ binary images of signatures (e.g. offline Italian database (Malik et al., 2015) or offline Japanese corpus (Malik et al., 2013)).

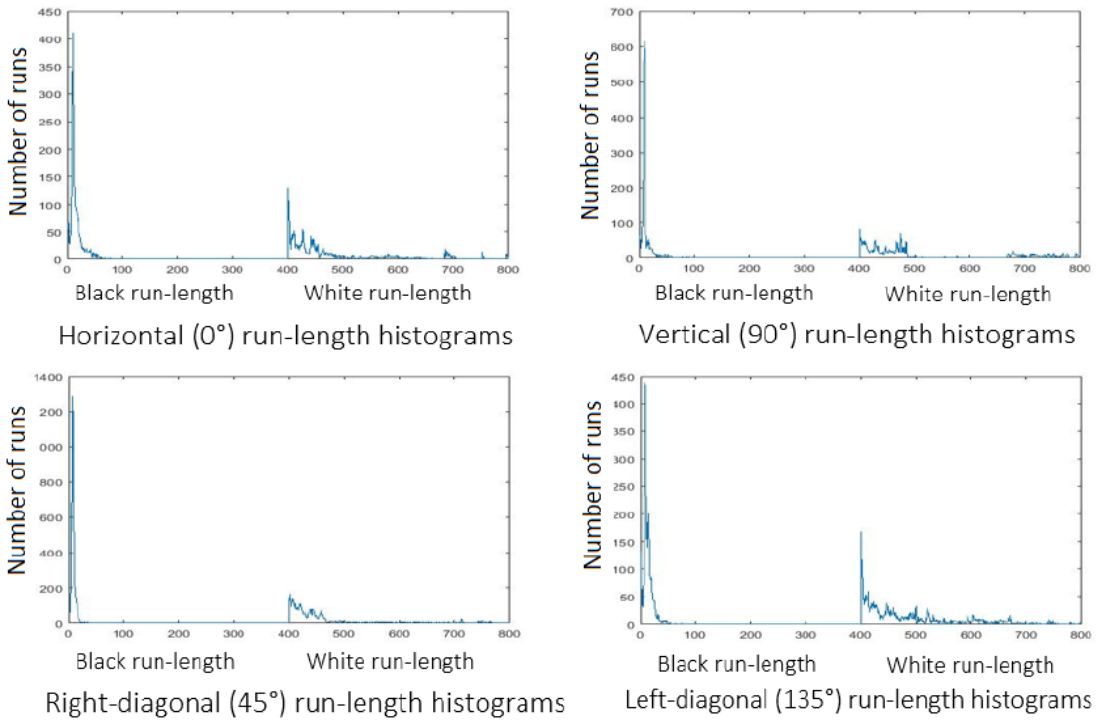


Fig. 3.3 Run-lengths histograms of an image-based signature.

A 'run' is defined as a sequence of connected pixels in a given direction all having the same intensity. In our case, a run is equivalent to a segment which is a sequence of white or black pixels. The run-length matrix is defined as a matrix $\mathbf{P}$ where the value at position (i, j) in the matrix represents the number of pixel runs of color i and length j in a given direction. The size of the matrix is $M \times K$ where M represents the number of unique colors (intensities) in the image while K is the maximum possible length of a run in a given direction. In our study, we consider the horizontal, vertical, left-diagonal and right-diagonal run-lengths on black and white pixels of the binarized images of signatures. The extraction of the run-length distribution features is illustrated for a 9×10 binary image in Figure 3.4.

We can design run-length histograms in the next equations;

$RL_b(i|\theta)$ is the (i)th element describing the number of runs with black values and length i, occurring in the image along an angle θ. Thus, $RL_w(j|\theta)$ is the (j)th element describes the number of runs with white value and length j occur in the image along angle θ.

Let's indicate the following notations:

- RL_b is the number of black run lengths in the image.
- RL_w is the number of white runs lengths in the image.
- N_b is the black run-length histograms for the four directions.

- N_w is the white run-length histograms for four directions.
- RL_{4D} is the Global black and white Run-Length histograms for 04 directions. The black and white run-length histograms are defined, respectively, as:

$$RL_b(\theta) = \sum_{i=1}^{RL_b} N_b(i|\theta) \tag{3.1}$$

$$RL_w(\theta) = \sum_{i=1}^{RL_w} N_w(j|\theta) \tag{3.2}$$

$\forall 1 \leq i \leq N_b$ and $1 \leq j \leq N_w$.

The black and white run-length histograms for a given direction are concatenated as:

$$RL(\theta) = [RL_b(\theta), RL_w(\theta)] \tag{3.3}$$

According to the pixel color, the black and white run-length histograms for the four directions are processed as:

$$RL_b = [RL_b(0), RL_b(45), RL_b(90), RL_b(135)] \tag{3.4}$$

$$RL_w = [RL_w(0), RL_w(45), RL_w(90), RL_w(135)] \tag{3.5}$$

Where the final feature vector based on run-length histograms are concatenated as:

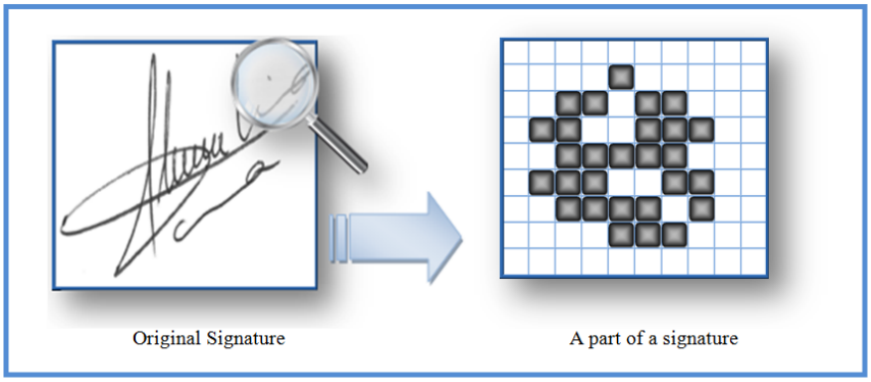
$$RL_{4D} = [RL_b, RL_w] = [RL_b(0), RL_b(45), RL_b(90), RL_b(135), RL_w(0), RL_w(45), RL_w(90), RL_w(135)] \tag{3.6}$$

A visual example of these vectors are represented in Fig. 4.3 for each direction.

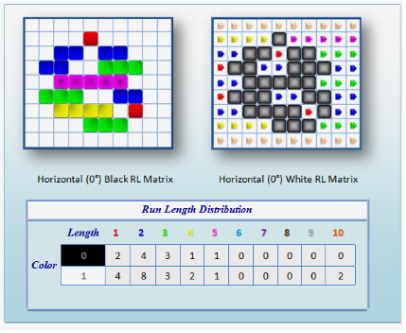
In our work, we vectorized the 2D image to get a single long line. At this level, the run-lengths are calculated for both black and white pixels. This procedure is applied to the other three directions, i.e. vertical, right-diagonal and left-diagonal. In another meaning, before calculating the lengths of runs, we juxtaposed the lines of the image in the desired direction, line by line in a way to form a single vector that denotes a new different presentation of the image. On this vector, we apply the same algorithm to calculate the Run-Length distributions for this given direction, and so for the other directions.

The elements of the matrices in Figure 3.4 represent the number of times, runs of different length occur in the four directions. For example, the elements in the first row of the horizontal

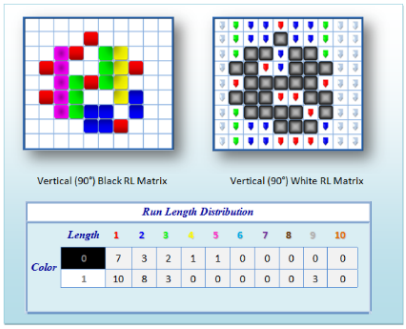
run-length matrix indicate that for pixel value 0 (‘black pixels’), there are 2 runs of length 1, 4 runs of length 2, 3 runs of length 3, 1 run of length 4, 1 run of length 5, and so on in the horizontal direction. The second row indicates the similar values for runs of 1 (‘white pixels’), where we have 4 runs of length 1, 8 runs of length 2, 3 runs of length 3, 2 run of



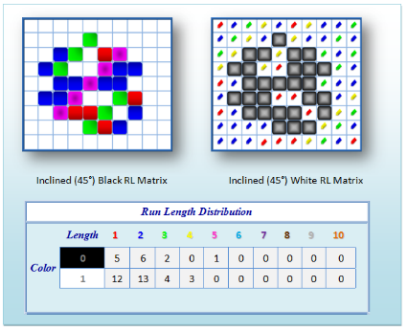
(a) Part of a black and white signature



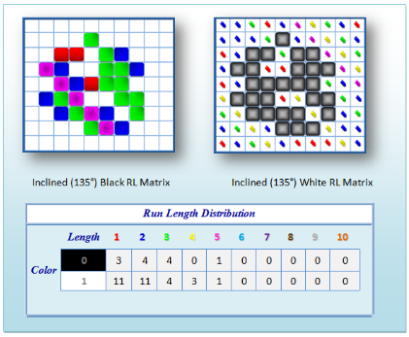
(b) Horizontal Run-lengths distributions (0°)



(c) Vertical Run-lengths distributions (90°)



(d) Right-diagonal Run-lengths distributions (45°)



(e) Left-diagonal Run-lengths distributions (135°)

Fig. 3.4 Run-lengths computation on a part of a image-based signature.

length 4, 1 run of length 5, and so on.Likewise, the run length matrices for other directions are illustrated in Figures 3.4-c 3.4-e.

The size of the run-length matrices is a function of the image size. However, it can be noticed that the informative non-zero values occur in the initial columns of the matrix only. Consequently, as described in algorithm 1, for each matrix, we only keep the first 400 columns. In other words, a run of a maximum of 400 pixels (determined empirically) is considered in our study. The four run length matrices are normalized and are converted to vectors (each of dimension 400) which are concatenated together to form a single feature vector.

Algorithm 1 Implementation of the run-length distributions features

Input: Read_image
Output: [Run_length_histograms]

```
[M,N] ← Image_size;
Black_run_length ← zeros(1,800);
White_run_length ← zeros(1,400);
Run_length_histograms ← zeros(1,400);
for row = 1 : M do
    Read(row)
    for current_pixel =1:length(row) do
        Score ← 0;
        if current_pixel==0 then
            repeat
                score++;
                Go to the next pixel;
            until (current_pixel_value different from next_pixel_value) or (score==400)
            Black_run_length(score)++;
        else
            repeat
                score++;
                Go to the next pixel;
            until (current_pixel_value different from next_pixel_value) or (score==400)
            White_run_length(score)++;
        end if
    end for
end for
Run_length_histograms = [Black_run_length, White_run_length]
```

For a given direction, The vector representing runs of white pixels is concatenated with the one representing runs of black pixels resulting in the vector of dimension 800 (400+400). The

total feature vector is the concatenation of the four directions vectors (horizonal 0º, vertical 90º, right-diagonal 45º and left-diagonal 135º), it get the size of 3200 (800+800+800+800).

3.3 One-Class Support Vector Machine Classifier

One of the requirements of traditional Support Vector Machine (SVM) classifier is the availability of positive and negative training examples, i.e. true samples and the counter examples. Finding effective negative examples, however, can be a challenging and costly task. Consequently, for many applications, the limitation of having negative examples results in replacing the option of using SVM classifier by distance-based systems. For signature verification systems, the most effective counter examples are the skilled forgeries that may allow selecting the appropriate thresholds. On the other hand, as highlighted in (Loka et al., 2017), the use of a validation set is one of the best strategies for threshold selection. Although systems trained using such negative examples are likely to report more competitive performances, skilled forgeries are not always available. Furthermore, as concluded by (Batista et al., 2012), designing a verification system that requires skilled forgeries in the training phase would lead to complications in using this technology for practical applications (i.e. a further investment would be required to collect the forged signatures each time an individual is enrolled in the system). A solution to alleviate this practice is using synthetic skilled forgeries in the training set (Ferrer et al., 2015). Using synthetic forgeries implies that a human expert in forging signatures is not required each time a new user enrolls in the system (Ballard et al., 2007).

Based on our research on this subject, we maintain the view that training set should not contain skilled forgeries. With the availability of state-of-the-art classifiers (like OC-SVM discussed shortly) which are able to report very low error rates without seeing the forged signatures, making the training process dependent on availability of forged signatures is indeed a big constraint. As discussed earlier, from the view point of practical verification systems, this would imply the requirement to collect forged signatures every time a new individual is enrolled in the system; a requirement that is not likely to be appreciated by the end users.

For scenarios where negative examples are hard to acquire, (Schölkopf et al., 1999) recommends training the algorithm for 'single-class classification' which is termed as *'novelty detection'* or detection of *'newness'*. This allows to classify the objects of a single class only differentiating them from all other possible objects. In general, the classifier gathers objects and considers others as outliers (Bergamini et al., 2009) while the decision model knows a set of examples and detects all that is new. Such classification techniques represent a good

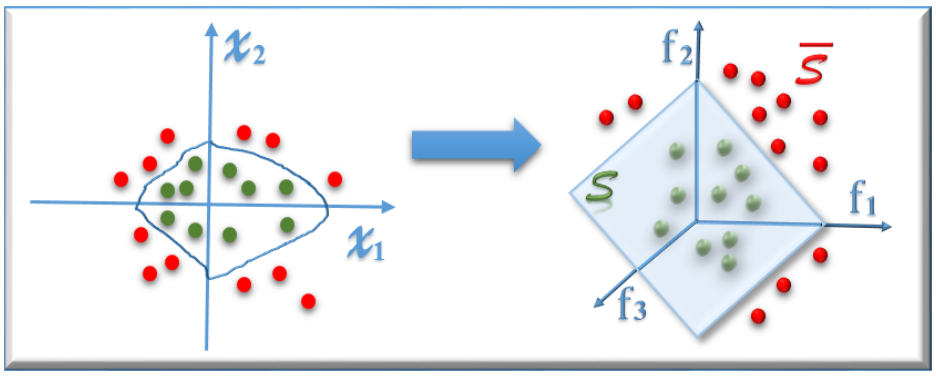


Fig. 3.5 Illustration of One-Class SVM (OC-SVM) Classification

choice for tasks like signature verification, a two-class pattern recognition problem where a questioned signature is to be classified as genuine or forged.

One-Class (OC) classification (Lumini and Nanni, 2009) distinguishes the target class from all other classes using only training data from the target class. The goal is to find a boundary between the examples of the target class and the rest of the space, i.e. a border around the target class that accepts as many examples as possible targets (Vapnik, 2013). This boundary is represented by a decision function which is positive within a class **S** and negative in the complement of $\bar{\mathbf{S}}$. Figure 3.5 shows a hypothetical example where data is transformed from a two dimensional feature space 'X' to a three dimensional space 'F' using an appropriate kernel function. The transformation ensures that the target class is linearly separable from all other classes (as SVM is a linear classifier). The algorithm returns a function $f_{OC}(x)$ that takes value $+1$ in a 'small' region capturing most of the data vectors of the positive class and -1 elsewhere (Schölkopf et al., 1999).

$$f_{OC}(x) = \begin{cases} +1 & \text{if } x \in \mathbf{S} \\ -1 & \text{if } x \in \bar{\mathbf{S}} \end{cases} \tag{3.7}$$

To set up the proposed signature verification system, we employ a design step and an evaluation step. Design steps aims to find the parameters of OC-SVM and decision threshold while evaluation step is employed to compute the verification errors and quantify system performance. The design step involves selecting a subset of signers, generation of signature models and finally the selection of optimal decision threshold. More specifically, a set of signers is selected from the database (281 signers in our case), each having N_g genuine signatures. To build the signer models, *GR* genuine signatures (4 for Scenario 1, 8 for Scenario 2 and 12 for Scenario 3) per signer are used, and the remaining genuine signatures for each signer are divided into two parts (Figure 3.6) namely Subset A and Subset B (Guerbai

et al., 2015). The first subset (Subset A) contains N_g^P genuine signatures which are used to find the parameters of the OC-SVM while the second subset (Subset B) contains N_g^T genuine signatures, which are used to determine the optimal decision threshold. The distribution of signatures in the two subsets corresponding to different experimental scenarios is summarized in Table 3.1. The parameters to be determined for the OC-SVM include the proportion of outliers ($\vartheta \in [0\,1]$) and the radial basis function kernel parameter ($\gamma \in [0\,1]$). The RBF kernel was itself chosen after experimenting with a number of kernel functions. To find optimal values of the parameters ($\vartheta^{OPT}, \gamma^{OPT}$), we train models using GR genuine signatures and vary these parameters in the range $[0\,1]$. The Half Total Error Rate, which corresponds to the average of the False Rejection Rate (FRR) and the False Acceptance Rate (FAR) is computed using Forged Questioned (FQ) and Genuine Questioned (N_g^P) signatures. The values of ϑ and γ which minimize the Half Total Error Rate are chosen. The number of signatures N_g^P and FQ for each of the experimental scenarios is listed in Table 3.1.

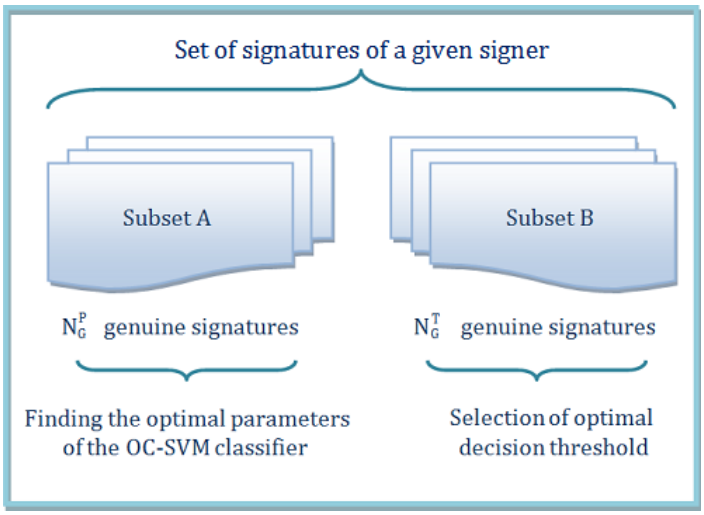


Fig. 3.6 Design step of the proposed system.

Table 3.1 Different Experimental Scenarios.

Scenario	Design step					Evaluation step			
	# Signers	GR	GQ		FQ	# Signers	GR	GQ	FQ
			N_g^P	N_g^T					
I	281	4	10	10	30	600	4	20	30
II	281	8	8	8	30	600	8	16	30
III	281	12	6	6	30	600	12	12	30

GR: Genuine Reference, GQ: Genuine Questioned, FQ: Forged Questioned.

For a given questioned signature(X_{S_i}), a positive value of the decision function of the OC-SVM does not ensure that the signature is genuine. In order to ensure a better verification performance, we propose to add a threshold t_f as a verifier using the following decision rule: X_{S_i} is accepted as a genuine signature if $f_{OC}(X_{S_i}) \geq t_f$, otherwise it is rejected. Conversely, when X_{S_i} is rejected (i.e. $f_{OC}(X_{S_i}) < t_f$), X_{S_i}, the signature is considered as a forged. The optimal threshold t_f is determined with the Subset B of N_g^T genuine signatures and FQ forged signatures (Table 3.1). Similar to optimization of SVM parameters, the Half Total Error Rate is used as metric to select the optimal threshold.

Once the optimal threshold t_f is determined, we proceed to the evaluation step where signature models are trained and subsequently evaluated for 600 signers according to different experimental scenarios discussed in the following section..

3.4 Experiments, results and discussion

This section describes the experiments that were performed, including the used databases, validates the proposed verification technique, and examines the outcomes. The evaluation measures are presented first, followed by the details of the different experiments.

3.4.1 Database

The experiments reported in our study are carried out on the GPDS-960 corpus (Vargas et al., 2007) of offline signatures, one of the largest publicly available signature databases. The database was collected in three different series. In the first series, 160 contributors were registered, the GPDS-100 release of the database comprises the signatures of the first 100 signers of this series. Later, 140 new contributors were added to produce the well-known GPDS-300 dataset. Finally, in the third series of data collection, 660 more signers were added to produce the GPDS-960 database[1]

The collection of all signatures of an individual was carried out in a single session. Each contributor provided 24 samples of their original signatures on a sheet of paper that was divided into 24 blocks. The volunteers used their own writing instrument to perform the signatures. Each of the sheets was then digitized at 600 dpi. Although 960 signers contributed to data collection, the current corpus comprises 881 signers as some of the contributors were missed later. Consequently, the corpus has a total of $881 \times 24 = 21144$ digital signature

[1]The actual GPDS-960 corpus is not available for privacy reasons. Instead, the GPDS Synthetic Offline Signature database is provided after signing a the license agreement (www.gpds.ulpgc.es). After extensive performance-based evaluations (Ferrer et al., 2017, 2013, 2015), the closenesses between GPDS Synthetic Offline Signature and GPDS-960 corpus has been proven.

images with a variety of writing instruments. To capture skilled forgeries, volunteers were asked to forge (after practice) the genuine signatures of each of the other signers three times. In total, 10 different volunteers falsified the signatures of a genuine user. As a result, 30 signatures repersenting skilled forgeries were acquired for each individual making a total of $881 \times 30 = 26430$ false specimens. Similar to the genuine signatures, the forgers used their own writing instruments to falsify the signatures.

All signatures (both genuine and forged) in the database were collected in Spain and hence are more suited for evaluation of signature verification systems targeting signatures in the Western style, one of the most complex forms for verification. As studied in (Diaz-Cabrera et al., 2015), the GPDS-960 corpus is distributed into signatures composed of only flourishes, only text (mainly the name and/or surname of the signer), and a combination of both text and flourishes. This wide variety of typologies allows evaluating signature verification systems in a more realistic scenario.

3.4.2 Performance Evaluation

The performance of the proposed verification system is quantified using a number of standard metrics. The most commonly used metrics for any verification system include Type I and Type II errors. Type I error or the False Rejection Rate (FRR) represents the ratio of the number of genuine test signature images rejected to the total number of genuine test signature images. Type II error or the False Acceptance Rate (FAR) represents the ratio of the number of accepted forgeries to the total number of forgeries. In addition to these standard measures, we also employed the metrics used in the recent signature verification competitions (Malik et al., 2015, 2013). These include accuracy (ACC) and the cost of the the log-likelihood ratios C_{llr}^{min} in its minimal possible value $\hat{C}_{llr}^{min}$. Accuracy (ACC) represents the percentage of correct decisions with respect to all disputed signatures. The motivation of using the minimal log-likelihood ratio is two-fold (Malik, 2015). Not only this metric is considered to be a significant measure for evaluating automatic verification systems from the perspective of pattern recognition community, it also supports the evidences assessed by Forensic Handwriting Examiners (FHE).

3.4.3 Experiments

We carried out a series of experiments to validate the ideas put forward in this study. First, we study the effectiveness of the run-length distribution features in detecting forged signatures and compare the performance of these features with known state-of-the-art features. Next, we analyze the performance evolution as a function of the number of enrolled signers in

the system. To match the real-world scenarios, we also study the verification performance in the context of a single reference signature system (SRSS). Finally, additional system implementations are evaluated and compared with the existing features.

3.4.4 Experiment I: Run-Length Features to Detect Forgeries

In the first experiment, we quantify the performance of run-length distribution features and compare the performance with a few of the latest features reported in the literature. These include Autoregressive Coefficients (Djeddi et al., 2015), Local Directional Pattern features (Ferrer et al., 2012), Local Binary Patterns (Ferrer et al., 2012) and Local Derivative Pattern (Ferrer et al., 2012). In addition, contour-direction distributions & contour-hinge distributions (Gilperez et al., 2008) and curvelet transform-based features (Guerbai et al., 2015) are also considered. Table 3.2 summarizes a brief description and the corresponding dimensionality for each of the employed features.

Experiments are carried out under three different scenarios. The first scenario includes, for each individual, four genuine signatures in the training set. The test set comprises, for each signer, the remaining twenty genuine signatures as questioned genuine specimens and all thirty skilled forgeries. In the second scenario, the reference set is composed of eight genuine signatures for each signer. Accordingly, the test set contains sixteen genuine signatures as well as thirty skilled forgeries. The third scenario employs twelve genuine signatures in the training set and, twelve genuine signatures and the thirty available skilled forgeries in the test set. The distribution of training and test sets in the three scenarios is summarized in Table 3.1. It should be noted that these experimental settings match closely to real world scenarios where only genuine signatures are available to be used for training and skilled forgeries are

Table 3.2 Summary of features employed in our study.

Feature	Description	Dimension
f1	Black Run-lengths Distributions, *this work*	400
f2	White Run-lengths Distributions, *this work*	400
f3	Black and White Run-lengths Distributions, *this work*	800
f4	Autoregressive Coefficients (Djeddi et al., 2015)	24
f5	Local Directional Pattern (LDP) (Ferrer et al., 2012)	672
f6	Local Binary Pattern (LBP)) (Ferrer et al., 2012)	3060
f7	Local derivative pattern (LDerive)) (Ferrer et al., 2012)	12240
f8	Contour-direction (Gilperez et al., 2008)	12
f9	Contour-hinge (Gilperez et al., 2008)	1042
f10	Curvelet transforms (Guerbai et al., 2015)	10

Table 3.3 Verification Performance of Different Features in Scenario I

Feat.	ACC (%)	FAR (%)	FRR (%)	$\hat{C}_{llr}^{min}$
f1	93.31	10.04	4.45	0.22
f2	90.51	15.08	5.76	0.35
f3	93.81	9.66	3.88	0.20
f4	72.44	26.22	28.45	0.78
f5	68.78	28.22	33.21	0.83
f6	77.82	21.36	22.73	0.68
f7	75.99	21.84	25.45	0.71
f8	67.55	28.59	35.03	0.85
f9	72.62	24.02	29.62	0.78
f10	71.34	29.51	28.10	0.82

Table 3.4 Verification Performance of Different Features in Scenario II

Feat.	ACC (%)	FAR (%)	FRR (%)	$\hat{C}_{llr}^{min}$
f1	94.45	7.82	4.34	0.19
f2	92.54	12.71	4.66	0.29
f3	94.92	7.77	3.65	0.18
f4	74.51	23.75	26.42	0.75
f5	69.48	25.67	33.10	0.80
f6	79.99	20.11	19.96	0.63
f7	78.37	18.59	23.24	0.64
f8	68.37	27.31	33.93	0.85
f9	73.62	22.72	28.40	0.77
f10	71.40	28.59	28.61	0.83

encountered only in the test phase. The database is divided into two parts. The first part is composed of the first 281 signers, which are considered in the design step, whereas the remaining 600 signers are considered in the evaluation step. This division allows to create signature models that can reject forgeries efficiently and also ensures that the process is tolerant to intra-writer variations.

The results are reported in terms of ACC, FAR, FRR and $\hat{C}_{llr}^{min}$, as discussed in Section 3.4.2. The performance of the proposed features as well as that of the state-of-the-art features when evaluated using the three scenarios is summarized in Tables 3.3, 3.4 and 3.5.

A number of interesting observations can be drawn from the realized verification errors. For Scenario I, it can be seen from Table 3.3 that the FAR and FRR of different features vary significantly with run-length distribution features (*f3*), outperforming all other features reporting a FAR of 9.66% and a FRR of 3.88%. Similar trends can be observed for Scenario

Table 3.5 Verification Performance of Different Features in Scenario III

Feat.	**ACC (%)**	**FAR (%)**	**FRR (%)**	$\hat{C}_{llr}^{min}$
f1	95.15	6.65	4.13	0.18
f2	93.42	11.44	4.64	0.27
f3	95.68	6.64	3.63	0.16
f4	73.83	22.79	27.52	0.75
f5	68.71	24.93	33.84	0.81
f6	80.86	18.89	19.23	0.61
f7	77.48	17.19	24.66	0.63
f8	66.28	26.47	36.63	0.86
f9	72.10	22.01	30.14	0.77
f10	70.34	28.32	30.20	0.83

II in Table 3.4, where the run-length distribution features again outperform all other features reporting a FAR of 7.77% and a FRR of 3.65%. Likewise, in the last experimental scenario (Table 3.5), the run-length distribution features report the minimum FAR of 6.64% as well as the minimum FRR of 3.63%. The relatively low performance of contour-direction features (*f8*), for example, can be attributed to the low dimensionality (12) of this feature as compared to other features.

From the perspective of other metrics employed in our study, a significant difference is observed in the $\hat{C}_{llr}^{min}$ values of run-length distributions and other features. For instance, the value for black and white run-length distribution features is 0.20 in the first scenario, 0.18 in the second and 0.16 in the third scenario. The best values of $\hat{C}_{llr}^{min}$ for other features read 0.61 and 0.63 when using LBP features (*f6*) for Scenario-III and Scenario-II respectively, 0.63 when using LDerive (*f7* - Scenario-III) and 0.68 for LBP again (Scenario-I). For all other features, the $\hat{C}_{llr}^{min}$ values exceed 0.7 in all three scenarios. Likewise, comparing the accuracy values, the highest realized accuracy reads 95.68% using black and white run-length distributions (*f3*) followed by a closer value of 95.15% when using black run-length distributions (*f1*) in Scenario-III of evaluations. The same features report accuracies of 94.92% and 94.45% respectively for the second scenario and 93.81% and 93.31% respectively for the first scenario. Similarly, the third best performance is reported by the white run-length distributions with accuracies of 93.42%, 92.54% and 90.51% for Scenarios III, II and I respectively. Among other features, the LBP descriptor realized accuracies of 80.86% 79.99% and 77.82% while the LDerive feature reported 77.48%, 78.37% and 75.99% for Scenarios III, II and I respectively.

It is interesting to note that for all the metrics employed, the run-length features outperform all other features in all three evaluation scenarios. Comparing the performance

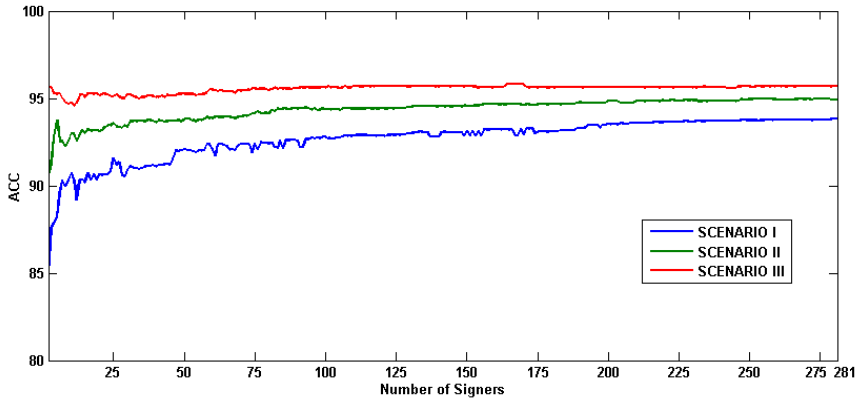
of different features across the three scenarios, it can be seen that the error rates reduce as the number of genuine signatures in the training set is increased. This observation is very much natural and consistent across all the features. The performance enhancement is more significant in the case of run-length distribution features as opposed to any of the other features. Summarizing, the error rates in Tables 3.3, 3.4 and 3.5 demonstrate the effectiveness of run-length distributions in detecting skilled forgeries and the ability to realize acceptable error rates with a small number of genuine samples in the training set. The results with other features, however, do not seem to be as effective and can be explored further by investigating different configurations (e.g. neighborhood sizes in computation of LBP or AR coefficients) for possible improvements.

3.4.5 Experiment II: Stability of System Performance as a Function of Number of Signers

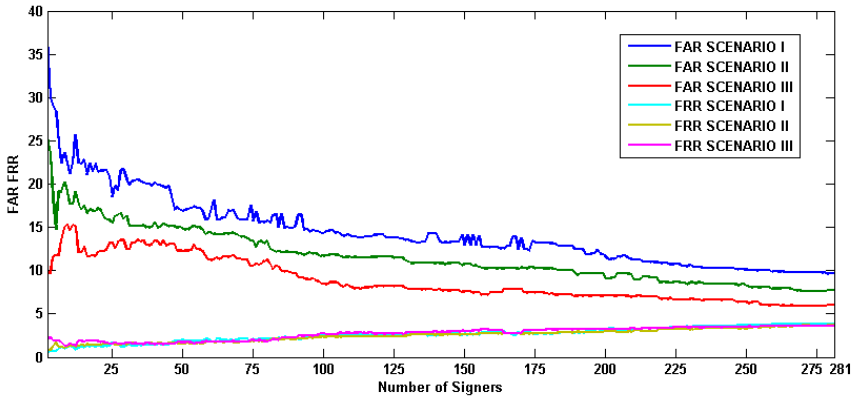
In the second series of experiments, we study the performance evolution concerning the number of signers both in the design and evaluation sets. Experiments are carried out by varying the number of signers for each of the three scenarios listed in Table 3.1. The performance of the OC-SVM models depends on two crucial parameters, the number of genuine samples used to construct the models and the number of signers involved in determining the optimal decision threshold (t_f). These two parameters will be considered in this experiment to study the stability of the proposed features. Like the previous experiments, the first 281 signers in the GPDS-960 constitute the design set while the remaining 600 signers constitute the evaluation set. The number of signers in each set is varied for experiments, as represented in the following description.

- The number of signers in gradually increased in the design set to compute the decision threshold (t_f) and the verification performance is measured on the evaluation set against each computed value of t_f (Figure 3.7).
- The decision threshold t_f computed using all 281 signers in the design set is fixed and the number of signers in the evaluation set is varied, under each of the three scenarios (Figure 3.8).

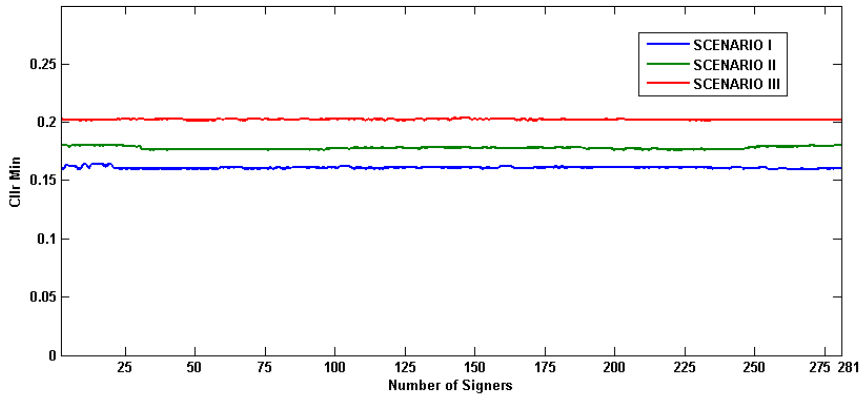
For the design set, it can be seen in Figure 3.7 that the ACC and $\hat{C}_{llr}^{min}$ values are more stable as compared to FAR and FRR values. Nevertheless, acceptable error rates are realized when using a small number of signers in the design set (to learn the decision threshold). On the evaluation set, it can be observed from Figure 3.8 that the evaluation metrics suffer from variations up to 50 signers. As the number of signers is increased further, the performance



(a) ACC values

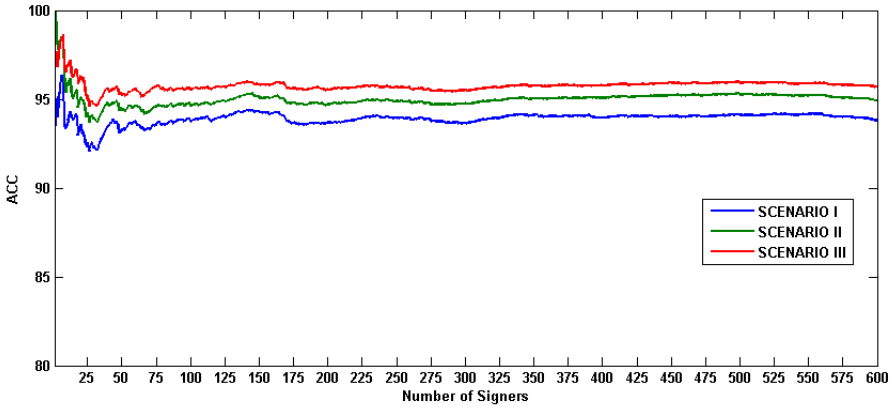


(b) FAR and FRR values

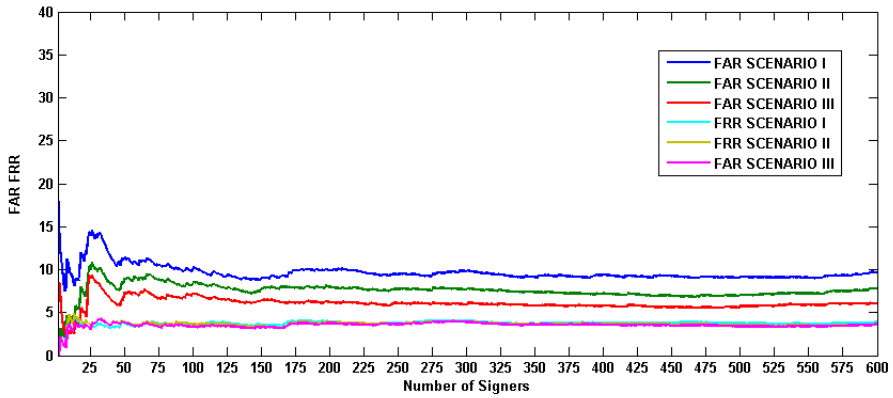


(c) $\hat{C}_{llr}^{min}$ values

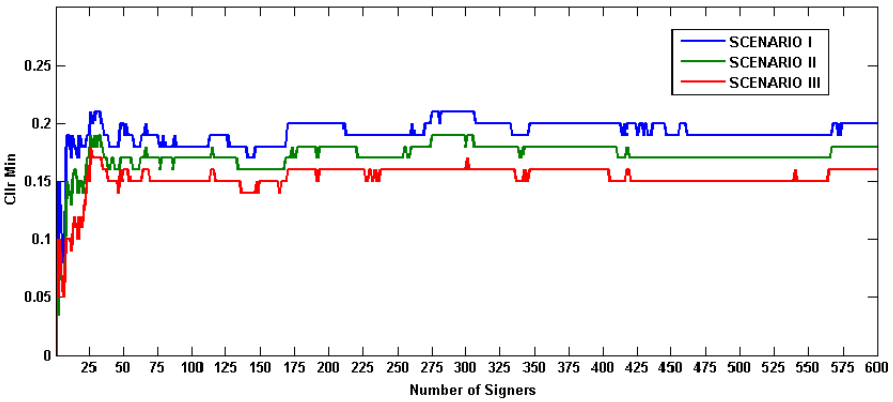
Fig. 3.7 Verification performance by changing the number of signers in the Design set



(a) ACC values



(b) FAR and FRR values



(c) $\hat{C}_{llr}^{min}$ values

Fig. 3.8 Verification performance by changing the number of signers in the Evaluation set

curves for different metrics start to stabilize and converge to almost straight lines. This stability is indicative of the effectiveness of the run-length distributions which appear to be almost insensitive to the number of signers.

3.4.6 Experiment III: Robustness in SRSS Scenario

One of the objectives of this study is to evaluate the effectiveness of the run-length distributions in the Single Reference Signature System (SRSS) scenario. A key issue in real world signature verification applications is the availability of only a limited number of samples per signer to train the system. These experiments are designed to target such real world scenarios where, in an extreme case, only a single genuine signature of each individual is available for training purposes. The experimental settings in these evaluations, therefore, involve only one genuine signature for each signer in the training set. The remaining twenty three genuine signatures as well as the thirty skilled forgeries per signer make part of the test set. Since each signer has a total of twenty four genuine signatures in the database, the experiments are repeated twenty four times using a different signature as reference in the training set. The results of these evaluations are reported in Table 3.6.

It can be observed from Table 3.6 that the run-length distributions realize low error rates even with a single reference signature in the training set. An average accuracy of 93.23% and $\hat{C}_{llr}^{min}$ value of 0.23 is reported by these experiments. This performance is better than any of the other features considered in our study (Table 3.2) even when using a larger number of training signatures with these features. Another interesting aspect of these evaluations is that the standard deviation of different performance measures (for 24 evaluations) is very low indicating that the features are equally effective when using any of the 24 signatures in the training set. These results are very encouraging indeed and validate the effectiveness of run-length distributions for practical verification systems (for instance banking applications) where only a limited number genuine signatures are available for training purposes.

3.5 Run-length features in competitions

For a more detailed comparison of the run-length features with the best performing signature verification features, we describe in this section the results of two competitions: the tough ICDAR 2015 signature verification competition SigWIcomp2015 (Malik et al., 2015) and the Thai Student Signatures and Name Components Recognition and Verification (TSNCRV2018), in conjunction with the 16th International Conference on Frontiers in Handwriting Recognition (ICFHR 2018) (Suwanwiwat et al., 2018).

Table 3.6 Off-line SRSS results when using different single sample as a reference.

Signature label	ACC (%)	FAR (%)	FRR (%)	$\hat{C}_{llr}^{min}$
1	91.92	8.75	7.57	0.27
2	91.64	10.36	6.84	0.27
3	92.66	9.64	5.57	0.24
4	92.97	9.33	5.27	0.23
5	93.06	7.41	6.57	0.23
6	93.40	7.86	5.63	0.22
7	93.79	7.64	5.10	0.21
8	93.69	7.14	5.68	0.21
9	93.69	7.00	5.78	0.21
10	93.80	7.92	4.88	0.21
11	93.70	7.23	5.58	0.21
12	93.75	7.06	5.63	0.22
13	93.53	7.49	5.68	0.22
14	93.51	8.11	5.24	0.21
15	93.34	8.39	5.33	0.22
16	93.59	6.92	6.02	0.22
17	93.62	8.12	5.04	0.22
18	93.33	8.57	5.21	0.22
19	93.29	7.92	5.78	0.22
20	93.51	7.08	6.03	0.22
21	93.32	8.25	5.48	0.22
22	92.93	9.01	5.58	0.23
23	93.20	7.87	5.98	0.23
24	92.28	9.95	6.01	0.26
Overall	93.23	8.13	5.73	0.23

3.5.1 ICDAR 2015 competition

The signature data was available in Bengali, Italian, and German, as well as handwriting data in several writing styles in English. The participants were asked to generate a comparison score (e.g., a degree of similarity or difference) as well as the evidential value of that score, which was expressed as the ratio of the probabilities of finding that score when the questioned signature is genuine versus when it is a forgery (i.e., the likelihood ratio). Signatures and handwritten text were designated as two modalities, with the following four challenges for this competition:

- Task 1: Italian off-line signature verification
- Task 2: Bengali off-line signature verification
- Task 3: German on-line signature verification

Table 3.7 Results of the SigWIcomp2015 Competition (Malik et al., 2015): Italian offline signature verification

ID	Systems	$\hat{C}_{llr}$	$\hat{C}_{llr}^{min}$
1	**HOG and LBP features + SVM (Yilmaz et al., 2011)**	**0.65**	**0.02**
2	Edge directional features + Manhattan distance	0.99	0.89
3	Edge-hinge features	1.07	0.95
4	Multi-scale run length features (Djeddi et al., 2013)	1.07	0.88
5	Edge-hinge and Multi-scale run length features (Djeddi et al., 2012)	1.07	0.90
6	Edge-hinge, Multi-scale run length and Edge directional features	1.04	0.90
7	Geometrical features and logistic regression classifiers (configuration 1) (Hassaïne et al., 2012)	8.90	0.97
8	Geometrical features and logistic regression classifiers (configuration 2) (Hassaïne et al., 2012)	13.11	0.96
9	Commercial System	1.00	0.99

- Task 4: Writer identification based on handwriting styles

The competition also included run-length distribution features, and the systems were ranked using the minimum log-likelihood ratio C^min (also used in our experiments). During the competition, the following sub-tasks were taken into consideration.

1. *Italian Offline Signature Verification.* A database of offline Italian signatures was required for this task. There were 50 signers in the database, including 229 authentic signatures and 249 skilled forgeries. Nine systems were submitted to this challenge. The signatures were all written in black and white (binary images). The fact that the signatures were collected over a five-year period makes this database particularly intriguing because it allows for the study of systems under various aging conditions.

   As summarized in Table 3.7, the run-length features, with $\hat{C}_{llr}^{min} = 0.88$, came in second among nine submitted systems for this task. The first rank was for the combination between two textural features (HOG and LBP) having $\hat{C}_{llr}^{min} = 0.02$, while the second system using Edge directional features had 0.88 for the same measure.

   The run-length feature had a $\hat{C}_{llr}$ of 1.07, the first system employing HOG and LBP features got a $\hat{C}_{llr}$ of 0.65, and the second system based on Edge directional features obtained a $\hat{C}_{llr}$ of 0.99. The commercial system was in third position with a score of 1.00, followed by the sixth system with a score of 1.04 for a combination of textural features (Edge hinge, run-length, and Edge directional features), the other systems gained lower performance.

2. *Bengali Offline Signature Verification* for this challenge; the same precedent systems were submitted, the Bengali offline signatures are offered, which varied significantly

Table 3.8 Results of the SigWIcomp2015 Competition (Malik et al., 2015): Bengali offline signature verification

ID	Systems	$\hat{C}_{llr}$	$\hat{C}_{llr}^{min}$
1	HOG and LBP features + SVM (Yilmaz et al., 2011)	0.69	0.05
2	Edge directional features + Manhattan distance	0.93	0.15
3	Edge-hinge features	0.94	0.12
4	**Multi-scale run length features (Djeddi et al., 2013)**	0.92	0.04
5	Edge-hinge and Multi-scale run length features (Djeddi et al., 2012)	0.93	0.06
6	Edge-hinge, Multi-scale run length and Edge directional features	0.93	0.06
7	Geometrical features and logistic regression classifiers (configuration 1) (Hassaïne et al., 2012)	1.16	0.97
8	Geometrical features and logistic regression classifiers (configuration 2) (Hassaïne et al., 2012)	2.84	0.30
9	Commercial System	1.00	0.89

from Western signatures. The used datasets included ten signers with a total of 120 genuine signatures and 330 skilled forgeries.

Table 3.8 indicates that run-length features outperformed all other systems with a minimum $\hat{C}_{llr}^{min}$ of 0.04 and gained the second rank with $\hat{C}_{llr}$ of 0.92 preceded by the first system with $\hat{C}_{llr}$ = 0.69. On the other hand, the first system (HOG and LBP features) got a $\hat{C}_{llr}$ of 0.05 and $\hat{C}_{llr}$ of 0.69 as the first best value among the other systems. The run-length distribution get also good scores when combining it with Edge-hinge features and with both of Edge-hinge features and Edge-directional features having the same scores for the two combinatiosn ($\hat{C}_{llr}$ =0.93 and $\hat{C}_{llr}^{min}$ =0.06).

3. *English Writer Identification based on Different Writing Styles* may provide writing samples in several writing styles for this challenge, which must be identified (for instance a threat letter written by a writer in an intentionally varied writing style). This assignment was created to match the needs of Forensic Handwriting Experts (FHE). Each of the database's 55 contributors submitted six writing examples in six different writing styles. The training set included writing examples in three distinct writing styles for each writer (a total of 165 images), whereas the test set included writing samples in the remaining three writing styles for each writer (a total of 165 images). The F1-measure (Fawcett, 2006), the harmonic mean of precision and recall, was used to compare the submitted systems.

   The run-length features get a score of 32.53%, as a third best score. On the other hand, the combination of these features with the Edge-hinge features could get the first rank with a score of 33.94%, and obtained the second rank combined with two features: Edge-hinge features and Edge directional features with the rate of 33.54 as shown in

Table 3.9 Results of the SigWIcomp2015 competition (Malik et al., 2015): Offline writer identification

ID	Systems	Avg. F1-measure (%)
22	Edge directional features + Manhattan distance	17.37
23	Edge-hinge features	33.54
24	Multi-scale run length features (Djeddi et al., 2013)	32.53
25	**Edge-hinge and Multi-scale run length features (Djeddi et al., 2012)**	**33.94**
26	Edge-hinge, Multi-scale run length and Edge directional features	33.54
27	Geometrical features and logistic regression classifiers (configuration 1) (Hassaïne and Al-Maadeed, 2012; Hassaïne et al., 2012)	30.71
28	Geometrical features and logistic regression classifiers (configuration 2) (Hassaïne and Al-Maadeed, 2012; Hassaïne et al., 2012)	21.01
29	Geometrical features and logistic regression classifiers (configuration 3) (Hassaïne and Al-Maadeed, 2012; Hassaïne et al., 2012)	21.01
30	Geometrical features and logistic regression classifiers (configuration 4) (Hassaïne and Al-Maadeed, 2012; Hassaïne et al., 2012)	20.80
31	Geometrical features and logistic regression classifiers (configuration 5) (Hassaïne and Al-Maadeed, 2012; Hassaïne et al., 2012)	19.60

Table 3.9. The Edge-hinge features got the same score of 33.54 followed by the system based on Geometrical features and logistic regression classifiers (configuration 1) with a measure of 30.71.

3.5.2 ICFHR2018 competition

For this challenge, nine submissions were received from prestigious academic and industrial laboratories. Five of them presented their algorithms. The Equal Error Rate (EER) was used as the performance measure.

The competition employed two different types of datasets: the first comprises Thai student signatures, and the second contains Thai student name components. The competition datasets collected signatures and name components from 100 volunteers. For each writer in the Thai signature dataset, there are 30 authentic signatures, 12 skilled forgeries, and 12 simple forgeries. Likewise, there are 30 genuine Thai name components and 12 skillfully forged name components for each writer. On each dataset, the tasks were divided into two parts: verification and recognition, for a total of four tasks.

The performed tasks were

- Signature verification task.
- Signature recognition task.

Table 3.10 EER results of the signature verification Task (Suwanwiwat et al., 2018)

Rank	Algorithm	Random Forgery	Skilled Forgery	Simple Forgery	Avg
1	CNN features (B)	0.0024	0.0830	0.0150	0.0327
2	Run-length features (C)	0.1304	0.2133	0.1600	0.1679
3	Edge-hinge features (D)	0.1551	0.2675	0.1992	0.2072
NA	CNN features ((E,F,G,H,I,J)	0.2301	ANS	ANS	NA
NA	Textural features combination (A)	CNC	0.0100	0.0100	NA
Benchmark Das et al. (2018)		0.0201	0.1108	0.0031	0.0447

- Name component recognition task.
- Name component verification task.

Our system was focused on the verification tasks.

1. *Signature verification task:* A total of 100 volunteers signed both genuine and forged Thai student signatures. In total, 3,000 genuine signatures were acquired (100 signer's 30 times). For each genuine signature, 12 skillfully forged signatures and 12 simple forged signatures were created, totaling 24 forged signatures per genuine signer.

   Skilled forged signers were asked to learn to forge genuine signatures of the other genuine signers. Simple forgeries are a set of 12 signatures per user with similar vocal outcomes as the original. It was found that 31 volunteers signed their signatures in English script, whereas the other 64 signed their signatures in Thai and 5 signers used both scripts to sign a single signature.

   As shown in Table 3.10, among the handcrafted features run-length features get the first position with EER = 0.1304% for random forgery, EER=0.2133% for skilled forgery and EER = 0.1600% for simple forgery with an average of 0.1679%. The system winner among all the systems is system B based on learned features, with EER =0.0024%, EER=0.0830% and EER=0.0150% for rand, skilled and simple forgery respectively with an average of 0.0327%.

2. *Name component verification task:* Genuine name components were obtained in the amount of 6,000 (100 students x 2 name components x 30 times). A total of 12 expertly forged name components were created for each genuine name component. There are a total of 2,400 skillfully forged names (100 students x 2 name components x 12 times). The components of Thai names, both genuine and forged, were acquired from 100 students aged 12 to 16 years old.

Table 3.11 EER results of the Name component verification Task (Suwanwiwat et al., 2018)

Rank	Algorithm	First name			Last name			First + Last Name Image Level			First + Last Name Score Level		
		RF	SF	Avg	RF	SF	Avg	RF	SF	Avg	RF	SF	Avg
1	CNN features (B)	0.0052	0.0574	0.0313	0.0023	0.0451	0.0237	0.0016	0.0292	0.0154	0.0004	0.0233	0.0118
2	Run-length features (C)	0.1656	0.1717	0.1687	0.1471	0.1492	0.1482	0.1140	0.1258	0.1199	0.1169	0.0958	0.1063
3	Edge-hinge features (D)	0.2368	0.2908	0.2638	0.2124	0.2467	0.1340	0.1055	0.1467	0.1261	0.1877	0.2283	0.2080
NA	L	0.3215	ANS	NA	0.3099	ANS	NA	0.2831	ANS	NA	ANS	ANS	NA
NA	Textural features combination (A)	CNC	0.0383	NA	CNC	0.0350	NA	CNC	0.0383	NA	CNC	0.0150	NA
(Suwanwiwat et al., 2017)		0.0004	0.1191	0.0598	0.0002	0.1111	0.0556	RP	RP	RP	RP	RP	RP

Using the motion time interval approach, each student was instructed to write their name (first and last name) 30 times on white paper in the provided space, then all of the samples have been binarized.

The results indicated in Table 3.11, show that the system B based on CNN features was the winner in all the verification tasks with an average of 0.0313% and 0.0237% for the first name and last name verification respectively. For the first and last name verification it got an average score of: 0.154% and 0.0118% for the image level and the score level respectively. Our system gained the second rank among all the systems and the first one among the handcrafted systems with an average of 0.1687% for the first name verification, avg=0.1482% for the las name verification, avg=0.1199% for the first and last name verification (image level) and lastly an average of 0.1063% for the first and last name verification (score level).

3.6 Conclusions

This chapter addressed the problem of offline signature verification using black and white run-length distributions as features and One-Class Support Vector Machine (OC-SVM) for classification. Only genuine signatures were employed to train the models to match the real-world verification scenarios, and different samples per signer were considered in the training set. The system reported low error rates even with a limited number of genuine signatures per signer in the training set. The system also realized very encouraging performance in the challenging Single Reference Signature System (SRSS) scenario with only one genuine signature per signer in the training database. Experiments also demonstrated that the system is stable concerning the varying number of signers in the design (to choose decision threshold) and evaluation steps. The run-length features are also shown to outperform the state-of-the-art signature verification features using the multiple evaluation metrics under the same experimental settings. Moreover, a comparison of run-length distributions with the systems

submitted in the ICDAR 2015 signature verification and ICFHR2018 competitions have also been presented.

Our further study on this subject is intended to include an investigation of other textural features for signature verification as well as feature selection methods to identify the most appropriate subset of features for this task. Among different potential directions, we plan to study the effectiveness of run-length distributions by adding new directions to the standard four directions. On the other hand, we aim to supplement every direction by its neighboring angles in order to get more performance in the signature verification.

Chapter 4

Offline signature verification system using multidirectional run-length features

4.1 Introduction

The usefulness of Run-length features in offline signature verification was demonstrated in the preceding chapter. Four direction-based elements are included in the configuration of these handcrafted features (Bouamra et al., 2020).

However, the number of directions used by this technique can limit the extracted information. This means of browsing may not give complete insight into the distribution of the runs around different directions. Furthermore, each direction can be reinforced by its neighborhood to perform a more detailed description of the entire area covered by the central direction, which would imply the reinforcement of the direction.

In this chapter, we propose to add further spatial information of the signature to the standard run-length features. Such information is worked out in two stages: firstly, beyond classical four directions, more directions are studied. Secondly, improving the knowledge of each direction by combining the information of the neighbor directions. This new configuration has been used in two classifiers, one based on Euclidean distance and another based on a one-class support vector machine.

On the one hand, the new features' contribution is trying to improve the performance of the run-length features and attempting to increase the yield of each direction by supporting it with the two adjacent directions. On the other hand, this strategy provides expanded information to the scanned direction.

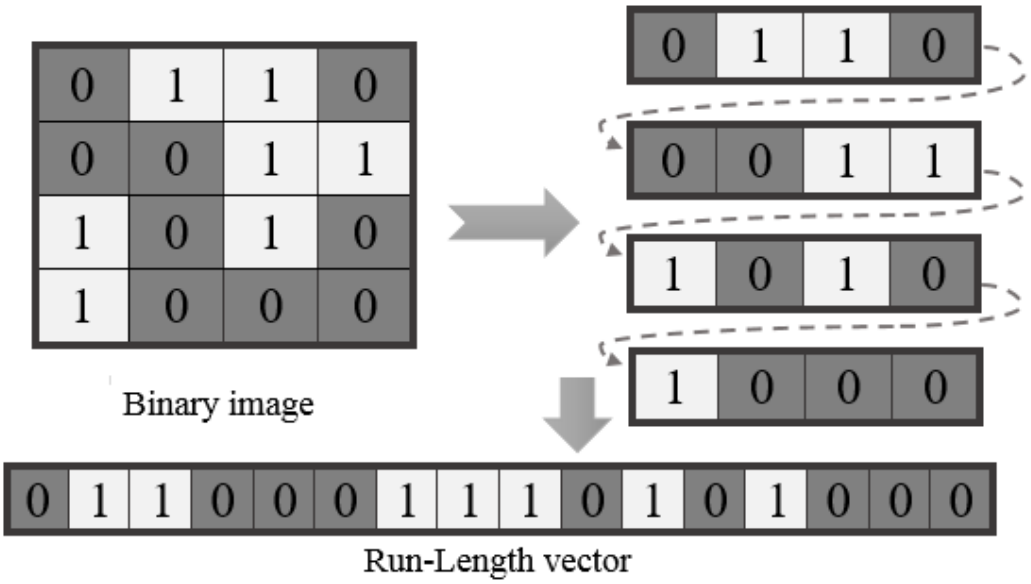
4.2 Multidirectional Run-Length Features

The standard run-length computations are outlined in concatenating the lines of the image-based signature, one after the other, respectively, in a feature with only one dimension. The task of runs counting will be executed on this feature. The concatenations are affected on the horizontal direction (0°) as much as for the three directions (45°), (90°) and (180°). Thus, the procedure consists first of transforming the image to a vector. Then, this vector will include the same image information but in a sequence of the lines of the image. After converting the image into the vector, we apply the algorithm of the run-lengths on the latter to have the run-length concluded features. We perform this procedure for the four principal directions.

The standard run-length features are illustrated in Figure 5.1. From the first row of the table, we notice that the image contains three runs composed of one pixel, no run of two pixels, and two runs of three pixels when we consider the black pixels runs. On the other side, for the white pixels, we observe two runs formed by one pixel, one run of two pixels, and another run of three pixels, as illustrated in the second row of the table. The total feature made by the black pixels is about 400 values, as well as the feature constituted of white pixels, which results in a full run-length feature of 800 values.

This chapter explains our contribution to multidirectional run-length features. Our proposal implies considering the four main directions already defined in the standard run-length features. Then, we added other directions, as a first step in increasing the directions of the scan to n directions. The new directions were interstitial within the four main directions to have a balanced scan of the image-based signature. To this aim, we had n directions to scan; for each direction of these n directions, it will be joined with its direct neighborhood to cover a full peripheral area, thus enriching the information given by this direction (see Figure 5.2).

For example: for the θ direction, we calculated three run-length features for the three directions: θ, $\theta_1 = \theta - \tau$ and $\theta_2 = \theta + \tau$, τ being 15° in this work. The constituted run-length feature of these three directions will present a run-length composite feature of the θ direction. For the 0° direction, we calculate three features for the three directions θ = 0°, θ_1 = -15° and θ_2 = 15°. Consequently, for the 90° direction, we calculate three features for the three directions θ = 90°, θ_1 = 75° and θ_2 = 105°. The composite feature is set for the 90° direction, and so on. The total number of directions to calculate their features is n $\times$ 3; every three features make a resulting composite feature, so the final feature is composed of n composite features. The value chosen for every composite feature is the minimum of the three features. The composite run-length feature at θ direction was the minimum of the three run-length features of neighboring directions. For θ = 0°, the composite feature is the minimal of the three features following the three directions θ = 0°, θ_1 = -15° and θ_2 = 15°,



	Run Length distribution				
Color/Length	**1**	**2**	**3**	**4**	**5**
Black (0)	3	0	2	0	0
White (1)	2	1	1	0	0

Fig. 4.1 Standard Run-lengths distributions

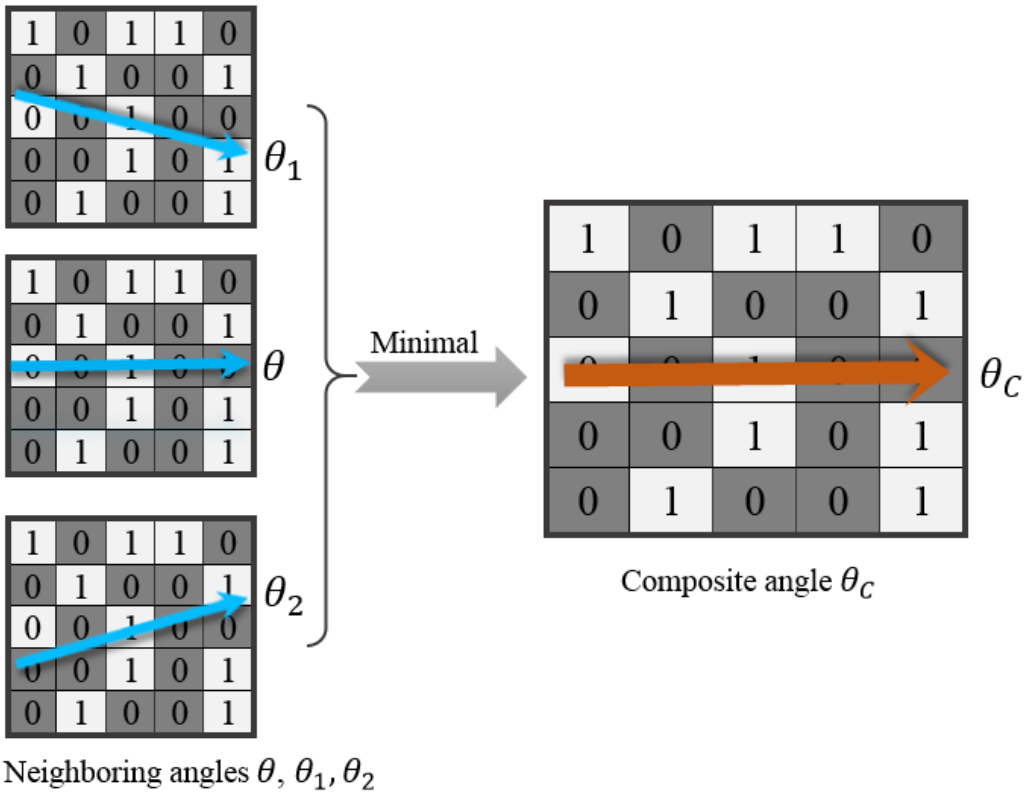


Fig. 4.2 Composite direction θ_c

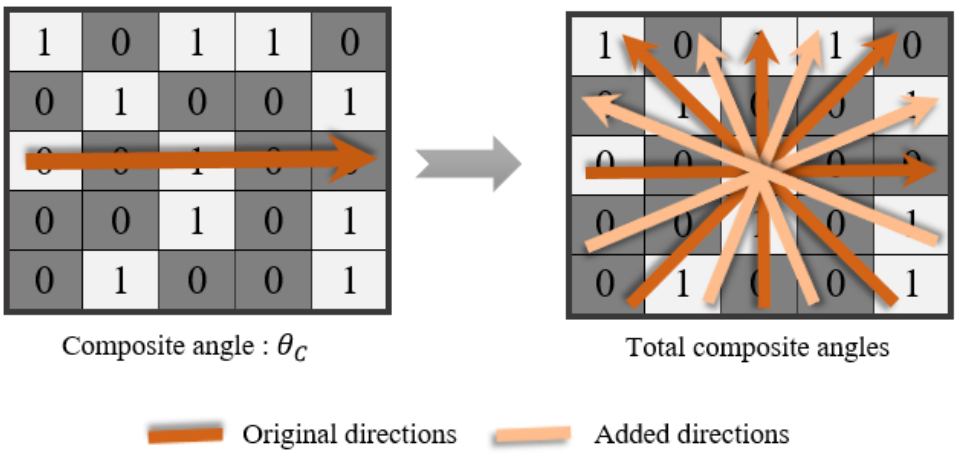


Fig. 4.3 Multidirectional run-length features conception

with the same method, we define the composite feature of the 90° direction as the minimum of the three directions$\theta = 90°$, $\theta_1 = 75°$ and $\theta_2 = 105°$. The final feature is composed of the total n composite directions. The Figure 4.3 depicts a visual example of the multidirectional run-lenght features.

Furthermore, every run-length feature of the eight ones is composed of 800 values 400 or black pixels + 400 for white pixels), so as a result, the final run-length feature will be composed of eight elementary features, with a size of 6400 equal 800×8.

We consider the following notations:

- θ is the given direction for the scan.
- (θ_1, θ_2) are the neighboring directions to θ.
- θ_c is the composite feature resulted of: θ_1, θ, θ_2.
- N_b is the black run-length histograms for the given direction.
- N_w is the white run-length histograms for the given direction.
- RL_b is the number of black run-lengths in the image.
- RL_w is the number of white runs lengths in the image.
- RL_{MD} is the multidirectional black and white run-length histograms for composite directions.

The black and white run-lengths defined, respectively, as:

$$RL_b(\theta) = \sum_{i=1}^{RL_b} N_b(i|\theta) \tag{4.1}$$

$$RL_w(\theta) = \sum_{i=1}^{RL_w} N_w(j|\theta) \tag{4.2}$$

$\forall 1 \leq i \leq N_b$ and $1 \leq j \leq N_w$

The black and white run-length histograms for a given direction are concatenated as:

$$RL(\theta) = [RL_b(\theta), RL_w(\theta)] \tag{4.3}$$

Thus, the black and white run-length histograms for a given composite direction are concatenated as:

$$RL_b(\theta_c) = [\min(RL_b(\theta_1), RL_b(\theta), RL_b(\theta_2))] \tag{4.4}$$

$$RL_w(\theta_c) = [\min(RL_w(\theta_1), RL_w(\theta), RL_w(\theta_2))] \tag{4.5}$$

$$RL(\theta_c) = [RL_b(\theta_c), RL_w(\theta_c)] \tag{4.6}$$

$$RL(\theta_c) = [\min(RL_b(\theta_1), RL_b(\theta), RL_b(\theta_2)), \min(RL_w(\theta_1), RL_w(\theta), RL_w(\theta_2))] \tag{4.7}$$

The run-length distributions for the eight composed directions by black and white pixels are computed as:

$$RL_b = [RL_b(0), RL_b(22.5), RL_b(45), RL_b(67.5), RL_b(90), RL_b(112.5), RL_b(135), RL_b(157.5)] \tag{4.8}$$

$$RL_w = [RL_w(0), RL_w(22.5), RL_w(45), RL_w(67.5), RL_w(90), RL_w(112.5), RL_w(135), RL_w(157.5)] \tag{4.9}$$

This leads to calculating the last feature as follows:

$$RL_{MD} = [RL_b, RL_w] \tag{4.10}$$

$$\begin{aligned} RL_{MD} = & [RL_b(0), RL_b(22.5), RL_b(45), RL_b(67.5), RL_b(90), RL_b(112.5), \\ & RL_b(135), RL_b(157.5), RL_w(0), RL_w(22.5), RL_w(45), \\ & RL_w(67.5), RL_w(90), RL_w(112.5), RL_w(135), RL_w(157.5)] \end{aligned} \tag{4.11}$$

We can assume that the new enrichment is an extended presentation of the standard run-length features; it provides a second number of directions, making it possible to calculate the runs in new directions that were not computed in our initial contribution (Bouamra et al., 2018). Besides, the calculation of each of these directions is affected peripherally. For instance, the information determined by this direction is enriched by its two neighboring directions. Therefore, the browsing is carried out in one composite direction developed by the three directions instead of only one direction.

On the other hand, the introduction of the minimal value from these three features enabled the improvement of the system's performance. Its results were better than introducing the averaged or the maximal value to present the composite direction.

Algorithm 2 presents the image transformation to a feature by one given direction. The second one shows the computation of run-length features for one composite direction composed of three adjacent directions.

4.3 Experiments

In this section, we introduce the databases, the experimental protocol carried out, and the experimental results obtained with the standard run-length features as well as the multidirectional run-length features.

4.3.1 Database

To compare our contribution with the standard run-lengths features, we employed two databases recently used in recent contributions with these handcrafted features. The first one is the GPDS960 database, which was used in Chapter 3 of this dissertation. The second one is the Thai student signatures database from the ICFHR18 competition (Suwanwiwat et al., 2018).

Algorithm 2 Multidirectional run-length feature extraction

Input: Read(θ), τ
Output: Multidirectional_Hist(θ)

```
Multidirectional_Hist(θ) ← zeros(1,800);
θ = θ − τ;
for direction_counter = 1 : 3 do
    Directional_Feature(θ)                                          ▷ Input
    Directional_Black_Hist ← zeros(1,400);                          ▷ Initialize
    Directional_White_Hist ← zeros(1,400);
    Directional_Hist ← zeros(1,800);
    Read(Directional_Feature);
    for current_pixel =1:lenght(Directional_Feature) do
        Score ← 0;
        if current_pixel==0 then
            repeat
                score++;
                Go to the next pixel;
            until (current_pixel_value different from next_pixel_value) or (score==400)
            Directional_Black_Hist(score)++;
        else
            repeat
                score++;
                Go to the next pixel;
            until (current_pixel_value different from next_pixel_value) or (score==400)
            Directional_White_Hist(score)++;
        end if
    end for
    Directional_Hist(θ) = [Directional_Black_Hist, Directional_White_Hist];
    Multidirectional_Hist(θ) = [Multidirectional_Hist(θ), Directional_Hist(θ)];
    θ = θ + τ;
end for
Multidirectional_Hist(θ) = min(Multidirectional_Hist(θ));
```

- **GPDS960 database:** This corpus comprises 881 users, with 24 genuine signatures and 30 skilled forgeries per user (Vargas et al., 2007). Although the signatures were collected in grayscale in a Spanish University with a different ballpoint pen, we binarize the whole database to work with the proposal features.

- **ICFHR18 Thai signature database (Das et al., 2018)**: This corpus comprises 100 users with 30 genuine signatures, 12 skilled and 12 simple forgeries for each writer. The images were shared in black and white format. Notice that this database contains signatures in Thai style and was used during the last competition in signature verification at the ICFHR18 conference.

4.3.2 Experimental Protocol

For the classification step, we performed two classifiers following the same protocol of the recent previous works that employed run-length features: The first one developed a one-class support vector machine (OC-SVM) classifier (Bouamra et al., 2020, 2018) whereas the second one worked out a Euclidean distance-based one (Suwanwiwat et al., 2018).

To set up the OC-SVM, the GPDS960 database was divided into a design dataset, composed of 281 writers, and in an evaluation dataset, with the remaining 600 writers. The design phase found the optimal value of the decision threshold, applying the first dataset. Specifically, we select a set of signers from the design dataset, each having several genuine signatures, one part of G_M is employed for creating signer models, depending on the scenario: four genuine signatures for the first one, eight and twelve signatures for the second and the third scenarios respectively (Bouamra et al., 2018). The remaining set is genuine questioned G_Q is by turn divided into two subsets, the first one G_P is used to settle the parameters of the OC-SVM and the second is G_T is used to fix the optimal decision threshold. The OC-SVM settings contain the portion of the outlier ϑ and the radial basis function kernel parameter γ (Guerbai et al., 2015). These parameters are trained and varied in the range [0, 1] for obtaining their optimal values $(\vartheta^{OPT}, \gamma^{OPT})$ that minimize the half total error rate HTER, which corresponds to the average of the false rejection rate (FRR) and the false acceptance rate (FAR) using forged questioned (F_Q) and genuine questioned signatures (G_Q) (Bouamra et al., 2018; Guerbai et al., 2015). It is worth pointing out that the use of OC-SVM efficiency is to resemble the real world verification scenarios, where we apply only positive samples to train the models.

Regarding the experimental protocol, we considered two previous proposals followed in (Bouamra et al., 2018) and (Suwanwiwat et al., 2018). Similar to (Bouamra et al., 2018), employing the evaluation dataset from GPDS960, three scenarios were simulated, which

comprise the use of the first four, eight, and twelve genuine signatures per user for training. In each case, the remaining samples and all the skilled forgeries were used for testing.

According to (Suwanwiwat et al., 2018), for the Thai signature database, the first five genuine samples of each signer were used for training. The Euclidean distance classifier computes the dissimilarity $d(Q,R)$ between the two signatures to match a questioned signature $Q = (Q_1, ..., Q_n)$ with each one of the first five genuine signatures $R = (R_1, ..., R_5)$.

$$d_i(Q,R_i) = (Q - R_i) \tag{4.12}$$

The average of the five distances is used as a score for each questioned signature.

$$D(Q,R) = \frac{1}{5}\sum_{i=1}^{5} d_i(Q,R_i) = \frac{1}{5}\sum_{i=1}^{5}(Q - R_i) \tag{4.13}$$

The remaining genuine specimens and all available skilled and simple forgeries per user were used for testing. For random forgery, we use the rest of the genuine samples of other users. For genuine scores: the number of scores for a signer is 25, and the total scores for all the signers are 25 $\times$ 100 equals 2500 scores. For the simple and skilled forgeries, respectively, the total number of scores is 12 $\times$ 100 comprises 1200. For the random forgeries: the number of scores for each user is the distances between the first five genuine signatures of the current signer and all the genuine signatures of all the other signers: 30 $\times$ 99 equal 2970 scores for each user, which produces the total random forgery for all users as 2970 $\times$ 100 equal 297000 scores. The performance measure is obtained by using the ROC curve between the genuine and forgery scores, depending on the type of forgery.

Different metrics evaluated our results. We used the accuracy (ACC), the false acceptance rate (FAR), the false rejection rate (FRR) and we enriched the performance measures by the cost of the log-likelihood ratios (C_{llr}) in its minimal possible value ($\hat{C}_{llr}^{min}$) for the evaluation of GPDS960, similar to (Bouamra et al., 2018). The $\hat{C}_{llr}^{min}$ has recently been employed to evaluate the accuracy of the output from automatic verification systems as an appropriate metric (Bouamra et al., 2018). As used in the ICFHR18 competition (Suwanwiwat et al., 2018), the equal error rate (EER) and the average EER evaluated the performance of the ICFHR18 Thai student database.

4.3.3 Optimizing the number of directions

We have chosen a defined number of directions to improve the run-length distributions. We did some experiments on one part of the GPDS960 database to determine the best combination of multidirectional run-length distribution. We changed the number of directions in order to

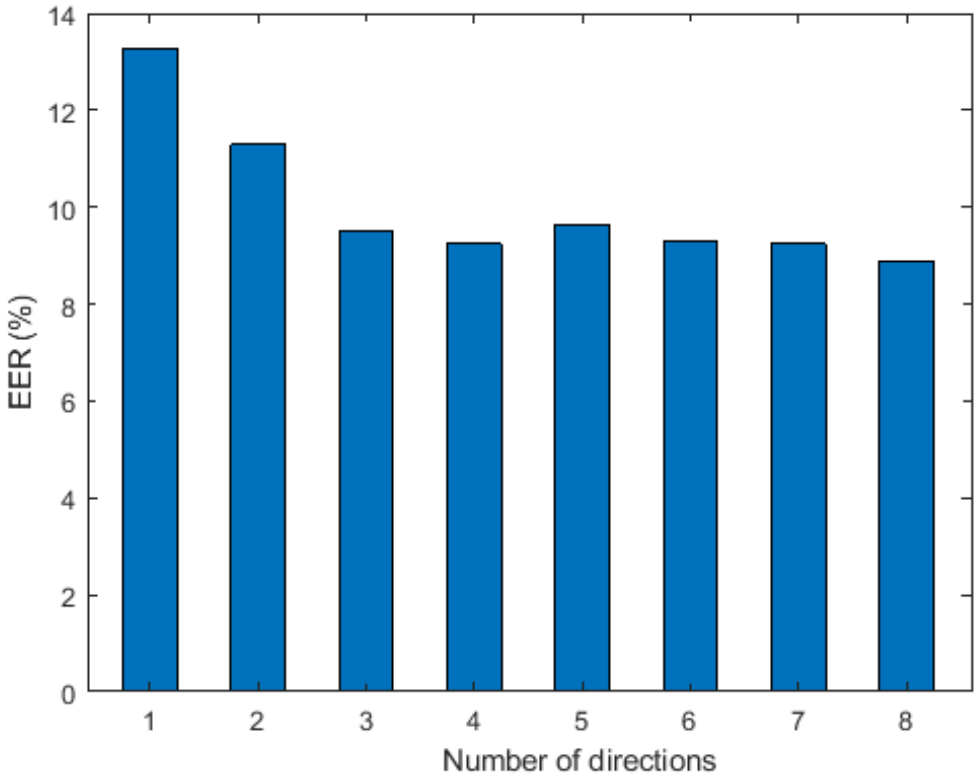


Fig. 4.4 Number of directions optimization.

select the optimal number that gives performant features. The experiments were based on selecting from one to eight directions and applying the multidirectional run-length features, then comparing the results one by one between these combinations. The metric used for the decision was EER. These experiments could give the results introduced in Figure 4.4.

The EER curve values show that using eight directions was the better choice compared to the other combinations. For instance, for one direction, the EER was about 13.27, the EER was reduced to EER = 9.24 for four directions, with a slight rise for the combination of five and six directions with EER=9.65 and EER=9.29 respectively. Finally, the curve receded again until getting minimal value EER=8.8. Thus, the best combination to choose is eight (n = 8) directions to have more performant features.

4.4 Results

The obtained performances of the two considered signature databases are shown in Tables 5.1 and 5.2. Generally, we notice a slight improvement with multidirectional run-length features in all cases.

Table 4.1 Results with Multidirectional Run-Length Features on GPDS960 database

System	Training Sign.	GPDS-960			
		ACC	FAR	FRR	C_{llr}^{min}
Standard RL (Bouamra et al., 2018)	4	93.81	**9.66**	**3.88**	0.20
This work		**93.91**	11.28	4.35	**0.23**
Standard RL (Bouamra et al., 2018)	8	94.92	7.77	3.65	**0.18**
This work		**95.02**	**6.53**	**4.45**	**0.18**
Standard RL (Bouamra et al., 2018)	12	95.68	6.64	3.63	0.16
This work		**95.99**	**5.65**	**3.45**	**0.15**

Table 4.2 Results with Multidirectional Run-Length Features on Thai Student database.

System	EER			
	RF	SF	RF	SF
System B (Suwanwiwat et al., 2018)	0.0024	0.0830	0.0150	0.0327
System K (Suwanwiwat et al., 2018)	0.2301	ANS	ANS	NA
System D (Suwanwiwat et al., 2018)	0.1551	0.2675	0.1992	0.2072
Standard RL (Suwanwiwat et al., 2018)	0.1304	0.2133	0.1600	0.1679
This work	0.0986	0.2000	0.1258	0.1415

In the case of GPDS-960 database, the performance with the proposed features was constant with the previous results for the first two scenarios, i.e., when 4 and 8 signatures were used for training. Instead, the performance was higher for the third scenario as shown in Figure 4.5: ACC = 95.99% in this work regarding ACC=95.66% in our past work. In addition, the FAR was slightly reduced from 6.64% to 5.65%. A similar observation is seen in the FRR metric, which was reduced from 3.63% to 3.45%, and $\hat{C}_{llr}^{min}$ in turn was reduced from 0.16 to 0.15.

In the case of ICFHR18 Thai student database, the results of the multidirectional run-length features were slightly better compared to previous results, as shown in Figure 4.6. According to Table 5.2, the EER for the random forgery was EER = 0.0986 for the multidirectional run-lengths against 0.1304 for the standard run-lengths. For the skilled forgery, we obtained: EER = 0.2000 for the new contribution against EER = 0.2133.

The current results also outperformed those obtained by the rest of the system that used handcrafted features. For the case of simple forgeries, the difference was more apparent where we got EER = 0.1258 against EER = 0.1600 for the standard run-length contribution.

The results have improved using multidirectional run-length features, more than using standard run-lengths. These techniques are still not very competitive with that basing on deep

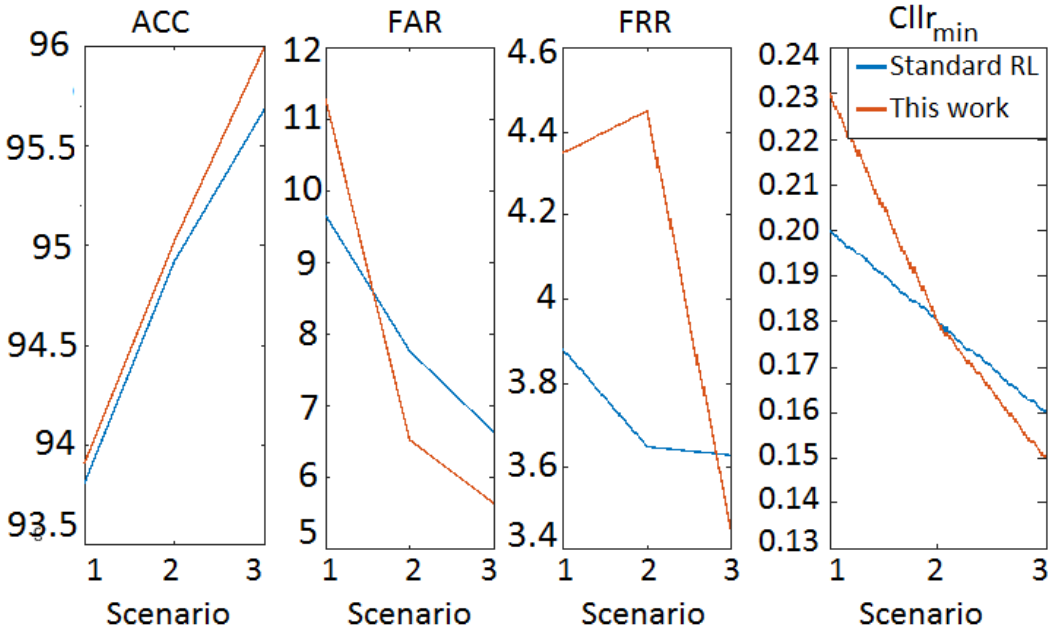


Fig. 4.5 Results on GPDS-960 database.

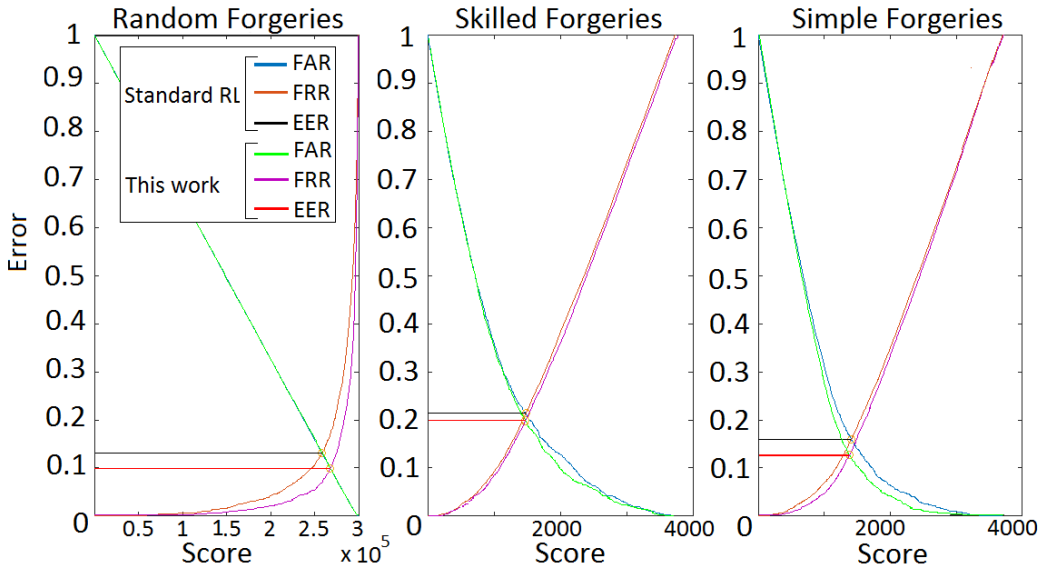


Fig. 4.6 Results on ICFHR Thai student database

learning approaches, as we can see in Table 5.2. It encourages us to combine the proposed features with learning-based ones in the future.

The experiments were effected in Matlab with an Intel Core i7-3770 CPU @ 3.40 GHz. The GPDS database contains 21144 genuine samples and 26314 skilled samples; the small signature is with a size of 153×258 pixels, and the large one is of size 819×1137 pixels. On the other hand, the Thai student signatures database includes 5400 signatures divided into 3000 genuine specimens, 1200 simple and 1200 skilled ones. The average sizes of these three types are varying from 49×11 pixels to 532×537. The computational costs in terms of averaged execution time for generating one signature using multidirectional run-length features are ranged between 6.58 and 41.27 seconds with an averaged time of 12.60 seconds for the GPDS database, and from 0.88 to 1.49 seconds for Thai student signatures database, taking 1.15 seconds as averaged time.

4.5 Multidirectional run-length features in competitions

In order to check the strength of our features in verifying signatures and ensuring their originality; the multidirectional run-length features were part of the submitted algorithms in the competition on Short answer ASsessment and Thai student SIGnature and Name COMponents Recognition and Verification (SASIGCOM 2020) in conjunction with the 17th International Conference on Frontiers in Handwriting Recognition (ICFHR 2020) (Das et al., 2020).

The proposed competition contains three elements which are short answer assessment (recognition and marking the answers to short-answer questions derived from examination papers), student name components (first and last names) and signature verification and recognition (Das et al., 2020). The performance measures used were the Equal Error Rate (EER) for the signature verification and the Accuracy (ACC) for the signature recognition task.

The same protocol of ICFHR18 was followed for the datasets (Suwanwiwat et al., 2018), where the signatures and name components data were collected from 100 volunteers. For the Thai signature dataset, there are 30 genuine signatures, 12 skilled and 12 simple forgeries for each writer. With Thai name components dataset, there are 30 genuine and 12 skillfully forged name components for each writer. There are 104 exam papers in the short answer assessment dataset, 52 of which were written with cursive handwriting; the rest of 52 papers were written with printed handwriting. The exam papers contain ten questions, and the answers to the questions were designed to be a few words per question as shown in Table 5.3.

Table 4.3 Detail of training and test partition on Thai Student dataset.

Nº. of Users	**Train**	**Test**		
	No. Of genuine samples/ user	**No. Of genuine samples/ user**	**No. Of skilled forgeries Sample/ user**	**No. Of Simple forgeries sample/ user**
100	5	25	12	12

Table 4.4 EER results of the signature verification task for ICFHR 2020.

Rank	**Algorithm**	**Random forgeries**	**Skilled forgeries**	**Simple forgeries**	**Avg**
1	SCUT-CNN	0.0019	0.0710	0.0090	0.0273
2	LTP+ oBIFs	0.0109	0.1091	0.0712	0.0637
3	ERL	0.0302	0.1780	0.0955	0.1012
4	oBIFs	0.0444	0.1876	0.1010	0.1110
5	LTP	0.0511	0.1901	0.1105	0.1172
6	MDRL	0.0986	0.2000	0.1258	0.1415
7	SPIRAL RL	0.1108	0.2045	0.1459	0.1537
8	RL400	0.1308	0.2145	0.1599	0.1686
9	RL	0.1308	0.2145	0.1599	0.1686
	Benchmark (Das et al., 2018)	0.0201	0.1108	0.0031	0.0447

Three teams from distinguished labs submitted their systems. For short answer assessment, word spotting task was also performed.

The submitted systems were divided in two types, learned features and handcrafted features. For the signature verification task and among the handcrafted ones (Table 5.4), our algorithm based on the multidirectional run-length features had an average performance EER=0.1415%, the best performance was for a learned based CNN system with EER=0.0273%. Our algorithm best performance was in random forgeries with EER=0.0986% then EER=0.1258% and EER=0.2000% for simple forgeries and skilled forgeries respectfully as shown in Table 5.4.

For the signature recognition task, our submitted system has a good ranking as the third of the global systems, and the second as a handcrafted system. It had a recognition accuracy ACC=0.9901%, where the first system had EER=0.9998% using CNN learned features followed by 0.9978% for the second one, as can be seen in Table 5.5.

Table 4.5 EER results of the signature recognition task for ICFHR 2020.

Rank	Algorithm	Recognition Accuracy
1	SCUT-CNN	0.9998
2	ERL	0.9978
3	MDRL	0.9901
4	SPIRAL RL	0.9874
5	RL400	0.9831

4.6 Conclusion

In this chapter, we propose the multidirectional run-length features for off-line ASV. These features can be seen as an update of the standard run-length features (Bouamra et al., 2018; Suwanwiwat et al., 2018). While previous features were based on the computation of the runs having pixels of the same value, and in a one given direction; our work is summarized by increasing the number of directions for the scan direction with its neighborhood. To this aim, we added the two neighboring directions to determine a combined value that defines a certain direction. Additionally, we increased the number of directions covered by the original technique to have a better performance compared to the initially browsing directions.

The obtained results slightly improved all previous and recent results obtained with standard run-length features. As proof of concepts, we adapt the multidirectional run-length features to two ASV systems, defined in (Bouamra et al., 2018; Suwanwiwat et al., 2018).

Current results motivated us to exploit other approaches that can increase the precision of our system, such as adding new directions or performing other combinations to raise the rate of falsifications detection.

Chapter 5

Offline signature verification system using spiral run-length features

5.1 Introduction

The run-length features proved their performance as a powerful spatial presentation of pixels under the concept of runs. Typically, such spatial distribution is achieved by counting the runs in four directions: horizontal, vertical, and two diagonal directions (Bouamra et al., 2022).

However, a significant problem is a user signature's well-known high inner variability. It could be mainly due to changes in shape, size, or other visual aspects, which causes a spatial distribution distortion within the image signature. All this limits the classic run lengths performance.

The main contribution of this chapter is the definition of a new direction in the framework of run-length features. This new direction is called the spiral direction, which adds a new image representation. Moreover, we combine this new direction with the classical four directions to improve the presentation of the run-length features. Finally, our work aims to study the efficiency of run-length features when adding the spiral direction for off-line ASV.

This new direction is expected to expand the run-length limitations due to its flexibility within the orientation of the scanned lines, which raises its robustness regarding the inner variability. In addition, it compensates for the static of each direction of the run-length features that traverses the image line by line in only one given direction.

5.2 Spiral Run-Length Features

In this section, based on the standard run-length distributions, we describe first the proposed spiral run-length feature. Next, we suggest two combinations to fuse the new feature with the previous four directions.

We vectorized the 2D image to get a single long line in our work. The run-length distributions are calculated for both black and white pixels at this level. This procedure is applied to the other three directions, i.e. vertical, right-diagonal and left-diagonal. In another meaning, before calculating the lengths of runs, we juxtaposed the lines of the image in the desired direction, line by line, to form a single vector that denotes a new different presentation of the image. On this vector, we apply the same algorithm to calculate the Run-Length distributions for this given direction, and so for the other directions.

Figure 5.1 illustrates a toy example of this procedure for the horizontal direction. For the black pixels, we get no run of length one, two runs of length two, one run of length three, and one run of length four, as indicated in the first row. A similar observation can be made for white pixels. The final horizontal vector is about 800 values (400 + 400 for black and white pixels, respectively). The procedure is repeated for the remaining directions. Because of the final concatenation of the four directions, the output run-length feature vector dimensions contain 3200 values.

Let's indicate the following notations:

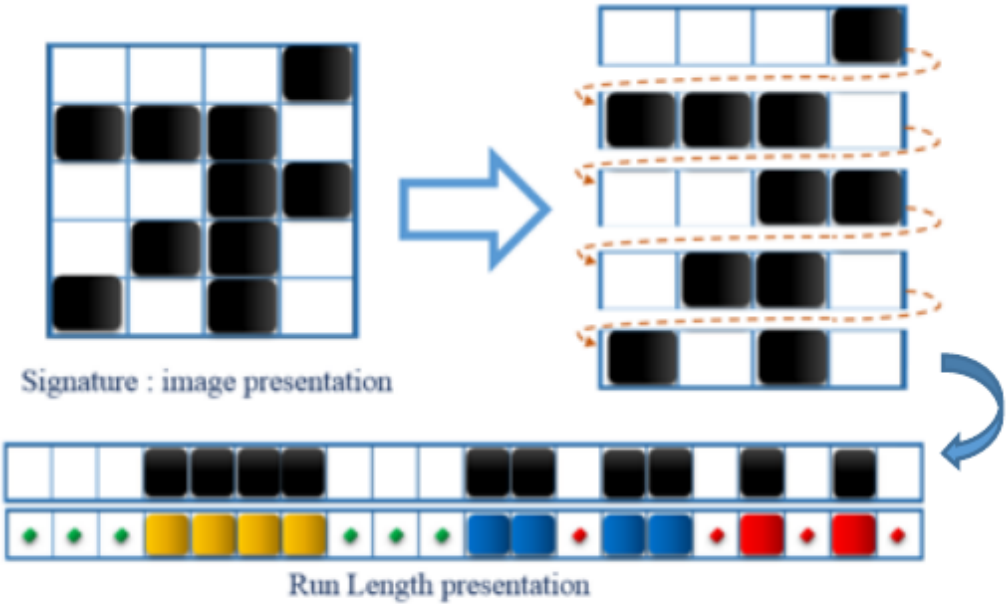
- RL_b is the number of black run lengths in the image.
- RL_w is the number of white runs lengths in the image.
- N_b is the black run-length histograms for the four directions.
- N_w is the white run-length histograms for four directions.
- RL_{4D} is the Global black and white Run-Length histograms for 04 directions.

The black and white run-length histograms are defined, respectively, as:

$$RL_b(\theta) = \sum_{i=1}^{RL_b} N_b(i|0) \tag{5.1}$$

$$RL_w(\theta) = \sum_{j=1}^{RL_w} N_w(j|0) \tag{5.2}$$

$\forall 1 \leq i \leq N_b$ and $1 \leq j \leq N_w$.



	Run Length distribution							
Color/Length	1	2	3	4	5	6	7	8
Black	2	2	0	1	0	0	0	0
White	4	0	2	0	0	0	0	0

Run Length Distribution

Fig. 5.1 Run-length distributions

The black and white run-length histograms for a given direction are concatenated as:

$$RL(\theta) = [RL_b(\theta), RL_w(\theta)] \tag{5.3}$$

According to the pixel color, the black and white run-length histograms for the four directions are processed as:

$$RL_b = [RL_b(0), RL_b(45), RL_b(90), RL_b(135)] \tag{5.4}$$

$$RL_w = [RL_w(0), RL_w(45), RL_w(90), RL_w(135)] \tag{5.5}$$

Where the final feature vector based on run-length histograms are concatenated as:

$$RL_{4D} = [RL_b, RL_w] = [RL_b(0), RL_b(45), RL_b(90), RL_b(135), RL_w(0), RL_w(45), RL_w(90), RL_w(135)] \tag{5.6}$$

5.2.1 Spiral feature vector

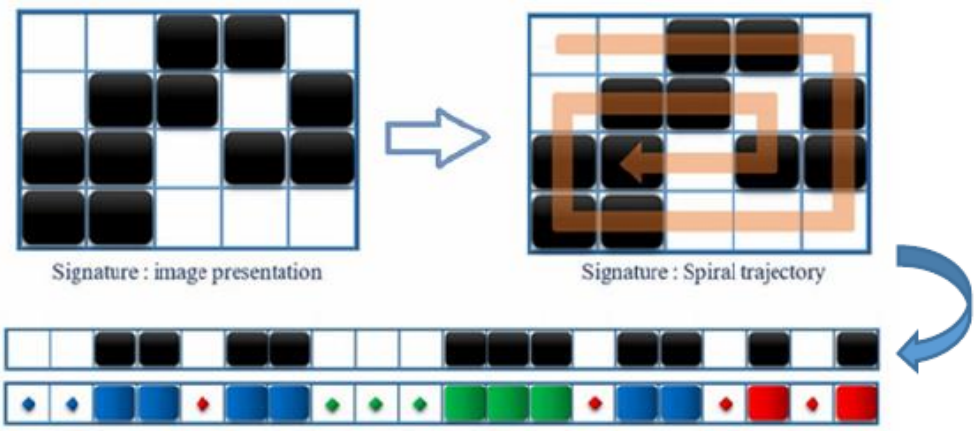
A uniform displacement describes it on a rotating line until reaching a final center point. This way, the spiral run-length feature traverses the entire image in a spiral counterclockwise curve starting from the first pixel at the upper left corner of the image. Then it moves away more and more towards a last central point. This spiral movement rotates between the horizontal and the vertical directions. The procedure is shown in Figure 5.2.

The spiral features treat four orthogonal directions differently, as shown in Figure 5.2. The movement hither is done permanently, starting with a horizontal direction with an angle $\theta_1 = 0$, followed by a descending vertical scan with an angle $\theta_2 = -90$. On reaching the end of the vertical column, the direction changes again, moving towards the horizontal direction but on the contrary direction to the first angle with an angle $\theta_3 = 180$. The last direction to progress is the vertically upward direction by exploring the entire column from bottom to top on an angle $\theta_4 = 90$. This round of four directions is iterated until browsing the entire signature image.

For counting the length of runs, the same procedure described in the previous chapter is applied to the resulting vector of the spiral function. Accordingly, the final spiral vector size contains 800 values (400 for black pixels + 400 for white ones).

We consider the next notations:

- SP_b is the number of black run lengths in the image.



Signature : Spiral Run Length presentation

	Spiral Run Length distribution							
Color\Length	1	2	3	4	5	6	7	8
Black	2	3	1	0	0	0	0	0
White	4	1	1	0	0	0	0	0

Spiral Run Length distribution

Fig. 5.2 Toy example describing the run-length distributions in spiral

- SP_w is the number of white runs lengths in the image.
- N_b is the black run-length histograms in spiral direction.
- N_w is the white run-length histograms in spiral direction.
- SP is the global black and white run-length histograms in spiral direction.
- θ_k is the browsing spiral angle: $\theta_1 = 0, \theta_2 = -90, \theta_3 = 180, \theta_4 = 90$.

The black and white run-length histograms are defined, respectively, as follows:

$$SP_b = \sum_{i=1}^{SP_b} \sum_{k=1}^{4} N_b(i|\theta_k) \tag{5.7}$$

$$SP_w = \sum_{i=1}^{SP_w} \sum_{k=1}^{4} N_w(i|\theta_k) \tag{5.8}$$

$\forall 1 \leq i \leq SP_b$ and $1 \leq k \leq SP_w$.
The global Spiral Run-Length histograms are then concatenated as:

$$SP = [SP_b, SP_w] \tag{5.9}$$

Therefore, the spiral transformation of the image is dynamic in direction (two changes: vertical/horizontal) and in orientation (two changes for every direction : $(\rightarrow, \leftarrow)$ and $(\uparrow, \downarrow)$. It is also dynamic in size; with every change of direction, we subtract a pixel. This transformation is based on four changes of the directions, and every current movement is starting from the second pixel (the first of this current movement is the last of the precedent one, so it is already calculated)

The spiral feature regroups both of two horizontal and vertical directions at the same time. It helps to add complementary information to the four previous run-length directions. Thus, the spiral run-length feature can be considered as the fifth direction.

The steps of the proposed feature are highlighted in the pseudo-code algorithms 3 and 4. They describe the spiral vector extraction and the spiral run-length features, respectively.

5.2.2 Combining spiral with the previous directions.

Two combinations are proposed to use the new spiral feature along with the previous run-length features. Specifically, they consist of combining the run-lengths features at the feature and score level.

Algorithm 3 Spiral vector extraction

Input: Read_image

$i \leftarrow 1; j \leftarrow 1;$
$[M,N] \leftarrow$ Image_size;
SP $\leftarrow$ [];
while $(i \leq M)$ and $(j \leq N)$ **do**
    image = image$(i : M, j : N)$
    $[M,N] \leftarrow$ Image_size
    if $i == M$ **then**
        $SP \leftarrow [SP,$ first_row$(j : N|0)]$;
    else if $j == N$ **then**
        $SP \leftarrow [SP,$ first_row$(j : N|0)$,last_column$(i+1 : M| - 90)]$;
    else if $i == M-1$ **then**
        $SP \leftarrow [SP,$ first_row$(j : N|0)$,last_column$(i+1 : M| - 90)$,last_row$(N-1 : j : -1|180)]$;
    else
        $SP \leftarrow [SP,$ first_row$(j : N|0)$,last_column$(i+1 : M| - 90)$,last_row$(N-1 : j : -1|180)$,first_column$(M-1 : i+1 : -1|90)]$;
    end if
    $i++; j++;$
    $M--; N--;$
end while

Algorithm 4 Spiral Run Length feature extraction

Input: Read_image

$i \leftarrow 1; j \leftarrow 1;$
$[M,N] \leftarrow$ Image_size;
Spiral_Black_Hist $\leftarrow zeros(1,400)$;
Spiral_White_Hist $\leftarrow zeros(1,400)$;
Spiral_Hist $\leftarrow zeros(1,800)$;
Read(SP); ▷ Spiral vector outcoming from image by spiral transformation.
for current_pixel = 1: Length(Spiral_Vector) **do**
    Score $\leftarrow$ 0;
    if current_pixel==0 **then**
        repeat
            score++;
            Go to the next pixel;
        until (current_pixel_value **different from** next_pixel_value) *or* (score==400)
        Spiral_Black_Hist(score)++;
    else
        repeat
            score++;
            Go to the next pixel;
        until (current_pixel_value **different from** next_pixel_value) *or* (score==400)
        Spiral_White_Hist(score)++;
    end if
end for
Spiral_Hist = [Spiral_Black_Hist, Spiral_White_Hist];

On the feature level, the combination consists of concatenating the four run-length features and the spiral feature. On the one hand, we concatenate all the five black run-length histograms and, on the other hand, the five white run-length histograms. This way, the combined histograms contain the five directions. Let RL_{5D} be the combined run-length histograms, it is defined as follows:

$$\begin{aligned} RL_{5D} =& [RL_b, SP_b, RL_w, SP_w] \\ =& [RL_b(0), RL_b(45), RL_b(90), RL_b(135), SP_b, \\ & RL_w(0), RL_w(45), RL_w(90), RL_w(135), SP_w] \end{aligned} \tag{5.10}$$

On the score level combination, this fusion is concerned by the scores generated by classifiers. The global score is a combination of the two scores of the previous four run-length features and the spiral one. An average of the two scores performs the combination:

$$Sc = \frac{Sc_1 + Sc_2}{2} \tag{5.11}$$

Sc being the final score, Sc_1 being the score of four directions run-length features and Sc_2 being the score of the spiral one. In both cases of features, we process the black and white pixel distribution.

The experiments are implemented on each of the two levels of combination, where more aspects will be specified in the next section.

5.3 Experiments

In this section, we present the used databases, the experimental protocol and the experiments with the two types of combinations: at both feature and score level when run-length features are used in ASV.

5.3.1 Database

We used the following two databases to evaluate our system.

- **GPDS75 database:** this database was introduced by (Vargas et al., 2007). It contains the first 75 writers; each one has 24 genuine signatures and 30 skilled forgeries.
- **CEDAR database:** it is one of the most frequently used database for off-line ASV (Kalera et al., 2004). This database comprises a total of 55 signatures of different signers. Each individual signed 24 genuine signatures and has a total of 24 forged specimens.

5.3.2 Experimental Protocol

We used the One-Class Support Vector Machine (OC-SVM), which has proved itself as an adequate and accurate signature classifier (Diaz et al., 2019; Guerbai et al., 2015). The experiments were implemented on the GPDS75 and CEDAR databases.

In the training stage, the first five (R5) and ten (R10) genuine signatures are kept as reference signatures. The testing stage was conducted by employing ten genuine samples and ten skilled forgeries for the experiments in both databases.

We have chosen the equal error rate EER to evaluate our systems. To this aim, we have studied the improvement of the system when combining it with the new spiral run-length features.

5.4 Results

We discuss here the combination at two levels: feature level and score level. The results of such fusions on GPDS75 and CEDAR databases are shown in Table 5.1 and Table 5.2, respectively. The same protocol was used for the CEDAR database; the results are mentioned in Table 5.2.

The experimental results obtained on Tables 5.1 and 5.2 indicate that the information combination increases the rate and improves the system performance; for the combination at both feature and score level.

Furthermore, we compare our results with previous works. Table 5.3 shows different works that have used the GPDS75 database. We can observe that our performances are in line with state of the art. For example, (Maergner et al., 2018b) obtained the best EER = 6.49%, while in another work, they got an EER = 6.84%. When we combine the five run-length features at the score level, our best performance was 6.86% on GPDS75.

According to Table 5.4, our results were competitive compared with previous works in CEDAR database. We observe a gap getting two minimal rates: EER = 0.18% and EER = 0.36%, followed by (Hamadene and Chibani, 2016) with AER = 2.10%, then hafemann et al. with EER= 4.63% accompanied by (Sharif et al., 2020) with EER = 4.67%. We conclude that our system was more performant with CEDAR database than GPDS75 database.

Table 5.1 Results in EER(%) on GPDS75 by combining at feature and score level.

System	GPDS-75	
	R5	**R10**
Basic RL (RL)	10.78	9.38
Spirial RL (SP)	12.88	11.62
[4RL, SP]*: Feature level	9.24	8.26
]4RL, SP]*: Score level	7.98	6.86

*]4RL,SP]: Combination of four run-length features with the spiral one.

Table 5.2 Results in EER(%) on CEDAR by combining at feature and score level.

System	CEDAR	
	R5	**R10**
Basic RL (RL)	0.73	0.55
Spirial RL (SP)	0.91	0.55
[4RL, SP]*: Feature level	0.55	0.36
[4RL, SP]*:Score level	0.73	0.18

*[4RL, SP]: Combination of four run-length features with the spiral one.

Table 5.3 Results on GPDS75 - comparison between the state of the art and our system.

Reference	Samples/user	EER%
(Maergner et al., 2018a)	10	6.84
(Maergner et al., 2017)	10	9.42
(Maergner et al., 2018b)	10	6.49
(Ferrer et al., 2012)	10	16.01
This work (score level)	5	7.98
This work (score level)	10	6.86

Table 5.4 Results on CEDAR - comparison between the state of the art and our system.

Reference	Samples/user	AER/EER%
(Guerbai et al., 2015)	12	5.6
(Sharif et al., 2020)	12	4.67
(Hafemann et al., 2017b)	12	4.63 ($\pm$0.42)
(Hamadene and Chibani, 2016)	5	2.10
This work (feature level)	10	0.36
This work (score level)	10	0.18

Table 5.5 Detail of training and test partition on Thai Student dataset.

Nº. of Users	**Train**	**Test**		
	No. Of genuine samples/ user	**No. Of genuine samples/ user**	**No. Of skilled forgeries Sample/ user**	**No. Of Simple forgeries sample/ user**
100	5	25	12	12

5.5 Offline automatic signature verification results in competitions

As well as the multidirectional features; the spiral run-length features were part of the submitted algorithms in the competition on Short answer ASsessment and Thai student SIGnature and Name COMponents Recognition and Verification (SASIGCOM 2020) in conjunction with the 17th International Conference on Frontiers in Handwriting Recognition (ICFHR 2020) (Das et al., 2020).

The proposed competition contains three elements which are short answer assessment (recognition and marking the answers to short-answer questions derived from examination papers), student name components (first and last names) and signature verification and recognition (Das et al., 2020). The performance measures used were the Equal Error Rate (EER) for the signature verification and the Accuracy (ACC) for the signature recognition task.

Following the same protocol of ICFHR18 for the datasets (Suwanwiwat et al., 2018), three teams from distinguished labs submitted their systems. The signatures and name components data were collected from 100 volunteers. For the Thai signature dataset, there are 30 genuine signatures, 12 skilled and 12 simple forgeries for each writer. Likewise, there are 30 genuine and 12 skillfully forged name components with the Thai name components dataset for each writer. Eventually, 104 exam papers in the short answer assessment dataset, 52 of which were written with cursive handwriting; the rest of the 52 papers were written with printed handwriting. The exam papers contain ten questions, and the answers to the questions were designed to be a few words per question, as shown in table 5.5.

Three teams from distinguished labs submitted their systems. For short answer assessment, word spotting task was also performed.

The submitted systems were divided in two types, learned features and handcrafted features. For the signature verification task and among the handcrafted ones (Table 5.6), our algorithm based on the multidirectional run-length features had an average performance EER=0.1415%, the best performance was for a learned based CNN system with

Table 5.6 EER results of the signature verification task for ICFHR 2020.

Rank	Algorithm	Random forgeries	Skilled forgeries	Simple forgeries	Avg
1	SCUT-CNN	0.0019	0.0710	0.0090	0.0273
2	LTP+ oBIFs	0.0109	0.1091	0.0712	0.0637
3	ERL	0.0302	0.1780	0.0955	0.1012
4	oBIFs	0.0444	0.1876	0.1010	0.1110
5	LTP	0.0511	0.1901	0.1105	0.1172
6	MDRL	0.0986	0.2000	0.1258	0.1415
7	SPIRAL RL	0.1108	0.2045	0.1459	0.1537
8	RL400	0.1308	0.2145	0.1599	0.1686
9	RL	0.1308	0.2145	0.1599	0.1686
	Benchmark (Das et al., 2018)	0.0201	0.1108	0.0031	0.0447

Table 5.7 EER results of the signature recognition task for ICFHR 2020.

Rank	Algorithm	Recognition Accuracy
1	SCUT-CNN	0.9998
2	ERL	0.9978
3	MDRL	0.9901
4	SPIRAL RL	0.9874
5	RL400	0.9831

EER=0.0273%. Our algorithm best performance was in random forgeries with EER=0.1108% then EER=0.1459% and EER=0.2045 for simple forgeries and skilled forgeries respectfully as shown in Table 5.6.

For the signature recognition task (see Table 5.7), our submitted system has a good ranking as the third of the global systems, and the second as a handcrafted system. It had a recognition accuracy ACC=0.9874%, where the first system had EER=0.9998% using CNN learned features followed by 0.9978% for the second one.

5.6 Conclusion

This chapter presents a new direction for run-length features based on the signature's spiral path. We observe performance improvements by combining the previous well-known four directions in run-length features with the proposal spiral direction. Thus, the spiral run-length features can be understood as the fifth direction, which is more robust to inner variability and get better results than using only the four run-length features.

The acquired outcomes confirm the usefulness of these features compared to other systems from the state-of-the-art.

In our future works, we seek to improve the performance of automatic signature verification by applying other techniques of fusion and combination. Also, we study other methods to process the run-length features and to extend its use in on-line signatures.

Chapter 6

Conclusions and Future Works

This thesis investigated the challenge of offline handwritten signature verification by using novel handcrafted algorithms during the feature extraction phase. We started with a quick overview of biometrics' key concepts, emphasizing the technology's extensive use in human identification and the difficulties in verifying its validity. We then focused on handwritten signatures as an essential biometrics modality, highlighting the need for a means to validate these signatures for legal and commercial reasons.

We have also listed several current state-of-the-art methods that have been employed for the same objective. Then we discussed our proposed systems in detail, which are based on scanning the image signature and computing pixels with similar values in a well-defined way while browsing in several directions and pursuing various angles. The proposed techniques have been put through experiments and submitted into international competitions. The results were promising when compared to other similar systems.

6.1 Conclusions

In this dissertation, we used run-length distributions to create an effective system for handwritten signature verification. The objective of our technique was instead based on how it examines and analyzes the signature image. The scanning process is carried out in a thoroughgoing path of the signature. It pursues multiple orientations as it follows distinct angles that allow for effective and efficient signature browsing and analysis. This hypothesis was introduced in the first chapter through introductory sections.

The application of run-length features in numerous fields is significant, as evidenced by the second chapter's in-depth exploration of state-of-the-art algorithms based on texture analysis and description. They were employed in image classification and texture analysis. They were also used in various medical sectors, including the analysis of medical imaging

such as ultrasound and MRI, as well as images of different diseases such as breast cancer. In addition, many studies used run-length features in the feature extraction process, which provided appropriate precision ratios and supported researchers in extrapolating their research results using this technique. Eventually, we single out two areas specific to handwriting: writer identification and handwritten signature verification.

As explained in the third chapter, the proposed technique was performed in verifying the handwritten signatures, where we employed the classical run-length features. Therefore, it is doable to witness its strength and resilience, as used on the well-known signature database. Furthermore, it was compared with concurrent algorithms for signature verification. Its precedence and height were recorded compared to other related algorithms.

Two further aspects have been found to support the theory of run-length distributions, which is based on paths and directions. The fourth and fifth chapters go through their explanation in more detail and show how they work. One of them works by adding a fifth direction to the standard features and spiraling through the signature. Furthermore, the second relies mainly on the nearby angles reinforcing each direction in order to create a composite angle and increase the number of angles from four simple angles to eight compound angles. This enrichment surpassed signature browsing in terms of the quality of signature analysis and classification.

We may ensure the credibility of our hypothesis and justify employing our chosen technique by participating in external competitions, in addition to the database experiments we conducted and comparing them to the state-of-the-art. That included additional databases with a previously defined protocol, which makes judging the quality of the proposed features more impartial and evident. Furthermore, when compared to handcrafted techniques, whether in verifying signatures or even in accompanying tasks related to handwriting, such as name recognition and verification, our methods occupied acceptable and even pioneering positions among the algorithms participating in these competitions.

It is important to point out some limitations to our features, such as the fact that we only used binary images, which limited image browsing to black and white pixels. Browsing grayscale as well as color photographs will surely extract more information from the signature and make distinguishing between different signatures more typical.

On the other hand, we had implemented our features one by one, especially when they were submitted to the external competitions. Therefore, it is convinced that their combination will undoubtedly allow a more efficient verification process with higher and more efficient results and accuracy.

Moreover, we presented run-length characteristics as a textural strategy for verifying offline handwritten signatures. Our method is based on a thorough examination of the entire

image. It counts how many similar runs there are and groups them into a run-length matrix. Furthermore, we have contributed three contributions to improve feature extraction and, as a result, reliable signature verification. The good benefit of run-length employment in the many domains of texture analysis, as well as the competing results produced in these fields, explains our choice. We employed well-known databases including GPDS-960, MCYT-75, and CEDAR in our research.

To imitate genuine scenarios with limited negative samples, we employed a one-class SVM classifier. Furthermore, throughout the training phase, we employed a small number of authentic samples to build the model for confirming the potency of our distributions.

The results of the many studies performed on each of the contributions support the validity of our choice, with the run-lengths outperforming other state-of-the-art systems.

Submissions to international competitions aimed at handwritten signatures and verification systems, such as ICFHR18 and ICFHR20, on the other hand, demonstrate the strength and efficiency of the chosen distributions in terms of feature extraction and, as a result, the acquired verification rate.

These findings motivate us to suggest additional contributions from run-length features and to join other contiguous domains.

6.2 Future Works

Our technique's performance and accuracy motivate us to improve our research and enlarge our application of run-length features to other fields.

The run-length features were applied on binary images, which boosts us to use them on grayscale signatures to have more extended information and bypass more images and databases that use signatures at the gray level. Likewise, the black/white run will become a gray level run. The computations will be based on the value of the pixel, the consecutive pixels having the same gray-level value will be computed in the same run. The run-length matrix will not contain only black and white runs but all the gray-level runs, that will enlarge the size of the matrix from two lines to the number of gray-level existing in the signature-image.

Such, a gray level run is a set of consecutive pixels having the same gray level value. The length of the run is the number of pixels in this run. We can compute a gray level run length matrix for runs as well as in binary images; with four principal directions: horizontal, vertical, left-diagonal, and right diagonal for a given image signature. The matrix element $M(i, j)$ specifies the number of times the signature contains a run of length j, in the given direction, consisting of points with gray level i.

In addition, we wish to use run-length features on hyperspectral images classification textural feature extraction to obtain spatial-spectral feature information. Hyperspectral images provide detailed spectral information through hundreds of spectral channels or bands with continuous spectral information that can classify diverse regions of interest (Rasti et al., 2020). The devoted information is a three-dimensional (3-D) array that denotes a pixel location in the image and denotes a wavelength corresponding to a spectral band. As well as the lack of training samples, the problem of image classification and characterization coming from the input space (remote sensing ...) of hyperspectral data is the enormous number of spectral bands that can exceed hundreds of bands. That involves using a feature extraction as textural features besides the spectral features to form a hyperspectral classification and obtain a mapping of the information from the set of bands that characterize the spectral signatures of the classes being discriminated.

For this reason, we propose using the run-length features in such an issue by considering the wavelengths as runs. Thus, we collect the wavelengths having the same length. Thereby, the run-length matrix M contains the number of spectral bands with the same color and same length. The matrix element $M(i, j)$ specifies the number of times the image contains a band of length j, with the color (spectra) i.

We also desire to apply run-length histograms in the online handwritten signatures verification. In literature, the use of histograms in such field was introduced by many researches, basing firstly only on angles derived from vectors connecting two consecutive points of an online signature. Then, the use of histograms was generated to more information extracted from an online signature such as: Trajectory, velocity, pressure, acceleration and their derivatives (Sae-Bae and Memon, 2014).

Our aim is to introduce an online signature verification system based on run-length histograms. On one hand, we consider the velocity $v(t) = \sqrt{v_x^2(t) + v_y^2(t)}$ where $v_x(t)$ is the velocity in x and $v_y(t)$ is the velocity in y. On the other hand, we consider each velocity's lope (the angle) θ. To apply the run-length histograms, the velocity value replaces the length of the run and the velocity lope replaces the color of the run. So, the matrix M includes how many times a velocity v_i appear in the signature with the value $\sqrt{v_{i_x}^2(t) + v_{i_y}^2(t)}$ and with the angle $\theta(v_i)$. For instance the element $M(1, 15)$ contains the number of velocities having the value 1 with the angle of 15°. As such, other features like position and acceleration can also be used for online signature verification by implementing run-length historgrams.

References

Agustsson, E., Mentzer, F., Tschannen, M., Cavigelli, L., Timofte, R., Benini, L., and Van Gool, L. (2017). Soft-to-hard vector quantization for end-to-end learning compressible representations. *arXiv preprint arXiv:1704.00648*. (Cited on page 8.)

Albregtsen, F. and Nielsen, B. (2000). Texture classification based on cooccurrence of gray level run length matrices. *Australian Journal of Intelligent Information Processing Systems*, 6(1):38–45. (Cited on pages 18 and 21.)

Aubin, V. and Mora, M. (2017). A new descriptor for person identity verification based on handwritten strokes off-line analysis. *Expert Systems with Applications*, 89(Supplement C):241 – 253. (Cited on page 27.)

Avola, D., Bigdello, M. J., Cinque, L., Fagioli, A., and Marini, M. R. (2021). R-signet: Reduced space writer-independent feature learning for offline writer-dependent signature verification. *Pattern Recognition Letters*, 150:189–196. (Cited on pages 28 and 35.)

Bal, R., Bakshi, A., and Gupta, S. (2019). Performance evaluation of optimization techniques with vector quantization used for image compression. In *Harmony search and nature inspired optimization algorithms*, pages 879–888. Springer. (Cited on page 8.)

Ballard, L., Lopresti, D., and Monrose, F. (2007). Forgery quality and its implications for behavioral biometric security. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1107–1118. (Cited on page 46.)

Batista, L., Granger, E., and Sabourin, R. (2012). Dynamic selection of generative–discriminative ensembles for off-line signature verification. *Pattern Recognition*, 45(4):1326–1340. (Cited on page 46.)

Bergamini, C., Oliveira, L. S., Koerich, A. L., and Sabourin, R. (2009). Combining different biometric traits with one-class classification. *Signal Processing*, 89(11):2117–2127. (Cited on page 46.)

Bertolini, D., Oliveira, L. S., Justino, E., and Sabourin, R. (2010). Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers. *Pattern Recognition*, 43(1):387–396. (Cited on page 6.)

Bidwell, J. (1988). *American History in Image and Text*. American Antiquarian Society, James Russell Wiggins lecture in the history of the book in American culture. (Cited on page 5.)

Blankers, V. L., van den Heuvel, C. E., Franke, K. Y., and Vuurpijl, L. G. (2009). Icdar 2009 signature verification competition. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 1403–1407. IEEE. (Cited on page 32.)

Blumenstein, M., Ferrer, M., and Vargas, J. (2010). The 4nsigcomp2010 off-line signature verification competition: Scenario 2. In *International Conference on Frontiers in Handwriting Recognition, (ICFHR)*, pages 721–726. (Cited on page 32.)

Bouamra, W., Diaz, M., Ferrer, M. A., and Nini, B. (2020). Off-line signature verification using multidirectional run-length features. In *Proceedings of the 10th International Conference on Information Systems and Technologies*, pages 1–8. (Cited on pages 10, 65, and 73.)

Bouamra, W., Diaz, M., Ferrer, M. A., and Nini, B. (2022). Spiral based run-length features for offline signature verification. In *Proceedings of the 20th International Conference on Graphonomics*, pages 1–15. (Cited on page 81.)

Bouamra, W., Djeddi, C., Nini, B., Diaz, M., and Siddiqi, I. (2018). Towards the design of an offline signature verifier based on a small number of genuine samples for training. *Expert Systems with Applications*, 107:182–195. (Cited on pages 7, 8, 9, 10, 12, 71, 73, 74, 76, and 80.)

Bulacu, M. and Schomaker, L. (2007). Text-independent writer identification and verification using textural and allographic features. *IEEE transactions on pattern analysis and machine intelligence*, 29(4):701–717. (Cited on pages 23 and 36.)

Bulacu, M., Schomaker, L., and Brink, A. (2007). Text-independent writer identification and verification on offline arabic handwriting. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 769–773. IEEE. (Cited on pages 23 and 26.)

Bulacu, M., Schomaker, L., and Vuurpijl, L. (2003). Writer identification using edge-based directional features. *writer*, 1:1. (Cited on page 26.)

Chawki, D. and Labiba, S.-M. (2010). A texture based approach for arabic writer identification and verification. In *2010 International Conference on Machine and Web Intelligence*, pages 115–120. IEEE. (Cited on pages 22 and 23.)

Chu, A., Sehgal, C. M., and Greenleaf, J. F. (1990). Use of gray value distribution of run lengths for texture analysis. *Pattern Recognition Letters*, 11(6):415–419. (Cited on pages 17, 20, and 36.)

Cpalka, K. and Zalasinski, M. (2014). On-line signature verification using vertical signature partitioning. *Expert Systems with Applications*, 41(9):4170 – 4180. (Cited on page 27.)

Dara, S. and Tumma, P. (2018). Feature extraction by using deep learning: a survey. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1795–1801. IEEE. (Cited on page 8.)

Das, A., Ferrer, M. A., Pal, U., Pal, S., Diaz, M., and Blumenstein, M. (2016). Multi-script versus single-script scenarios in automatic off-line signature verification. *IET biometrics*, 5(4):305–313. (Cited on pages 31 and 33.)

Das, A., Suwanwiwat, H., Ferrer, M. A., Pal, U., and Blumenstein, M. (2018). Thai automatic signature verification system employing textural features. *IET Biometrics*, 7(6):615–627. (Cited on pages 10, 62, 73, 79, and 93.)

Das, A., Suwanwiwat, H., Pal, U., and Blumenstein, M. (2020). Icfhr 2020 competition on short answer assessment and thai student signature and name components recognition and verification (sasigcom 2020). In *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 222–227. IEEE. (Cited on pages 8, 9, 10, 40, 78, and 92.)

Dasarathy, B. V. and Holder, E. B. (1991). Image characterizations based on joint gray level—run length distributions. *Pattern Recognition Letters*, 12(8):497–502. (Cited on pages 17, 21, and 36.)

Derea, A. S., Abbas, H. K., Mohamad, H. J., and Al-Zuky, A. A. (2019). Adopting run length features to detect and recognize brain tumor in magnetic resonance images. In *2019 First International Conference of Computer and Applied Sciences (CAS)*, pages 186–192. IEEE. (Cited on page 21.)

Diaz, M., Ferrer, M. A., Eskander, G. S., and Sabourin, R. (2016a). Generation of duplicated off-line signature images for verification systems. *IEEE transactions on pattern analysis and machine intelligence*, 39(5):951–964. (Cited on page 31.)

Diaz, M., Ferrer, M. A., Eskander, G. S., and Sabourin, R. (2017). Generation of duplicated off-line signature images for verification systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(5):951–964. (Cited on page 34.)

Diaz, M., Ferrer, M. A., Impedovo, D., Malik, M. I., Pirlo, G., and Plamondon, R. (2019). A perspective analysis of handwritten signature technology. *Acm Computing Surveys (Csur)*, 51(6):1–39. (Cited on pages 7, 27, and 90.)

Diaz, M., Ferrer, M. A., and Sabourin, R. (2016b). Approaching the intra-class variability in multi-script static signature evaluation. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1147–1152. (Cited on page 33.)

Diaz, M., Fischer, A., Ferrer, M. A., and Plamondon, R. (2016c). Dynamic signature verification system based on one real signature. *IEEE Transactions on Cybernetics*, PP(99):1–12. (Cited on page 32.)

Diaz, M., Fischer, A., Plamondon, R., and Ferrer, M. A. (2015). Towards an automatic on-line signature verifier using only one reference per signer. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 631–635. (Cited on page 32.)

Diaz-Cabrera, M., Ferrer, M. A., and Morales, A. (2015). Modeling the lexical morphology of western handwritten signatures. *PLOS ONE*, 10(4):1–22. (Cited on page 50.)

Djeddi, C., Meslati, L.-S., Siddiqi, I., Ennaji, A., El Abed, H., and Gattal, A. (2014). Evaluation of texture features for offline arabic writer identification. In *2014 11th IAPR international workshop on document analysis systems*, pages 106–110. IEEE. (Cited on page 26.)

Djeddi, C., Siddiqi, I., Al-Maadeed, S., Souici-Meslati, L., Gattal, A., and Ennaji, A. (2015). Signature verification for offline skilled forgeries using textural features. In *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 76–80. IEEE. (Cited on pages 9, 35, 36, 40, and 51.)

Djeddi, C., Siddiqi, I., Souici-Meslati, L., and Ennaji, A. (2013). Text-independent writer recognition using multi-script handwritten texts. *Pattern Recognition Letters*, 34(10):1196–1202. (Cited on pages 9, 24, 40, 59, 60, and 61.)

Djeddi, C. and Souici-Meslati, L. (2011). Artificial immune recognition system for arabic writer identification. In *International Symposium on Innovations in Information and Communications Technology*, pages 159–165. IEEE. (Cited on page 26.)

Djeddi, C., Souici-Meslati, L., and Ennaji, A. (2012). Writer recognition on arabic handwritten documents. In *International Conference on Image and Signal Processing*, pages 493–501. Springer. (Cited on pages 24, 26, 40, 59, 60, and 61.)

Doermann, D. S. and Rosenfeld, A. (1995). Recovery of temporal information from static images of handwriting. *International Journal of Computer Vision*, 15(1-2):143–164. (Cited on page 5.)

Dutta, A., Pal, U., and Lladós, J. (2016). Compact correlated features for writer independent signature verification. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 3422–3427. IEEE. (Cited on page 31.)

Ergen, B. and Baykara, M. (2014). Texture based feature extraction methods for content based medical image retrieval systems. *Bio-medical materials and engineering*, 24(6):3055–3062. (Cited on page 36.)

Erp, M. v., Vuurpijl, L. G., Franke, K., and Schomaker, L. R. (2003). The wanda measurement tool for forensic document examination. (Cited on page 22.)

Fairhurst, M. (1997). Signature verification revisited: promoting practical exploitation of biometric technology. *Electronics & communication engineering journal*, 9(6):273–280. (Cited on page 5.)

Fairhurst, M. and Kaplani, E. (2003). Perceptual analysis of handwritten signatures for biometric authentication. *IEE Proceedings-Vision, Image and Signal Processing*, 150(6):389–394. (Cited on page 5.)

Fan, D., Yu, P., Du, P., Li, W., and Cao, X. (2012). A novel probabilistic model based fingerprint recognition algorithm. *Procedia Engineering*, 29:201–206. (Cited on page 40.)

Faundez-Zanuy, M. (2006). Biometric security technology. *IEEE Aerospace and Electronic Systems Magazine*, 21(6):15–26. (Cited on page 2.)

Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874. (Cited on page 60.)

Ferrer, M. A., Diaz, M., Carmona-Duarte, C., and Morales, A. (2017). A behavioral handwriting model for static and dynamic signature synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1041–1053. (Cited on page 49.)

Ferrer, M. A., Diaz-Cabrera, M., and Morales, A. (2013). Synthetic off-line signature image generation. In *2013 International Conference on Biometrics (ICB)*, pages 1–7. (Cited on page 49.)

Ferrer, M. A., Diaz-Cabrera, M., and Morales, A. (2015). Static signature synthesis: A neuromotor inspired approach for biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):667–680. (Cited on pages 46 and 49.)

Ferrer, M. A., Vargas, J. F., Morales, A., and Ordonez, A. (2012). Robustness of offline signature verification based on gray level features. *IEEE Transactions on Information Forensics and Security*, 7(3):966–977. (Cited on pages 27, 51, and 91.)

Fischer, A., Diaz, M., Plamondon, R., and Ferrer, M. A. (2015). Robust score normalization for DTW-based on-line signature verification. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 241–245. IEEE. (Cited on page 27.)

Fornes, A., Dutta, A., Gordo, A., and Llados, J. (2011). The ICDAR 2011 music scores competition: Staff removal and writer identification. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1511–1515. IEEE. (Cited on page 41.)

Galbally, J., Fierrez, J., Martinez-Diaz, M., and Ortega-Garcia, J. (2009). Improving the enrollment in dynamic signature verfication with synthetic samples. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 1295–1299. IEEE. (Cited on page 32.)

Galloway, M. M. (1975). Texture analysis using gray level run lengths. *Computer graphics and image processing*, 4(2):172–179. (Cited on pages 19, 13, 14, 18, 20, and 36.)

Ghanim, T. M. and Nabil, A. M. (2018). Offline signature verification and forgery detection approach. In *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, pages 293–298. IEEE. (Cited on page 36.)

Gilperez, A., Alonso-Fernandez, F., Pecharroman, S., Fierrez, J., and Ortega-Garcia, J. (2008). Off-line signature verification using contour features. In *11th International Conference on Frontiers in Handwriting Recognition, Montreal, Quebec-Canada, August 19-21, 2008*. CENPARMI, Concordia University. (Cited on page 51.)

Guerbai, Y., Chibani, Y., and Hadjadji, B. (2015). The effective use of the one-class svm classifier for handwritten signature verification based on writer-independent parameters. *Pattern Recognition*, 48(1):103–113. (Cited on pages 30, 33, 47, 51, 73, 90, and 91.)

Guru, D., Manjunatha, K., Manjunath, S., and Somashekara, M. (2017). Interval valued symbolic representation of writer dependent features for online signature verification. *Expert Systems with Applications*, 80(Supplement C):232 – 243. (Cited on page 27.)

Hafemann, L. G., Sabourin, R., and Oliveira, L. S. (2016). Writer-independent feature learning for offline signature verification using deep convolutional neural networks. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 2576–2583. IEEE. (Cited on pages 31 and 33.)

Hafemann, L. G., Sabourin, R., and Oliveira, L. S. (2017a). Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition*, 70:163–176. (Cited on page 32.)

Hafemann, L. G., Sabourin, R., and Oliveira, L. S. (2017b). Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition*, 70:163 – 176. (Cited on pages 34 and 91.)

Hafemann, L. G., Sabourin, R., and Oliveira, L. S. (2017c). Offline handwritten signature verification - literature review. In *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–8. IEEE. (Cited on page 27.)

Hafemann, L. G., Sabourin, R., and Oliveira, L. S. (2019). Characterizing and evaluating adversarial examples for offline handwritten signature verification. *IEEE Transactions on Information Forensics and Security*, 14(8):2153–2166. (Cited on pages 30 and 35.)

Hamadene, A. and Chibani, Y. (2016). One-class writer-independent offline signature verification using feature dissimilarity thresholding. *IEEE Transactions on Information Forensics and Security*, 11(6):1226–1238. (Cited on pages 90 and 91.)

Hannad, Y., Siddiqi, I., Djeddi, C., and El-Kettani, M. E.-Y. (2019). Improving arabic writer identification using score-level fusion of textural descriptors. *IET Biometrics*, 8(3):221–229. (Cited on pages 26 and 36.)

Haralick, R. M., Shanmugam, K., and Dinstein, I. H. (1973). Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621. (Cited on page 14.)

Hassaïne, A. and Al Maadeed, S. (2012). Icfhr 2012 competition on writer identification challenge 2: Arabic scripts. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 835–840. IEEE. (Cited on page 41.)

Hassaïne, A. and Al-Maadeed, S. (2012). An online signature verification system for forgery and disguise detection. In *International Conference on Neural Information Processing*, pages 552–559. Springer. (Cited on page 61.)

Hassaïne, A., Al-Maadeed, S., Alja'am, J. M., Jaoua, A., and Bouridane, A. (2011). The ICDAR 2011 arabic writer identification contest. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1470–1474. IEEE. (Cited on page 41.)

Hassaïne, A., Al-Maadeed, S., and Bouridane, A. (2012). A set of geometrical features for writer identification. In *Neural Information Processing*, pages 584–591. Springer. (Cited on pages 59, 60, and 61.)

He, S. and Schomaker, L. (2016). General pattern run-length transform for writer identification. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 60–65. IEEE. (Cited on page 24.)

He, S. and Schomaker, L. (2017). Writer identification using curvature-free features. *Pattern Recognition*, 63:451–464. (Cited on page 25.)

Impedovo, D. and Pirlo, G. (2008). Automatic signature verification: The state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(5):609–635. (Cited on pages 5 and 32.)

Impedovo, D., Pirlo, G., and Plamondon, R. (2012). Handwritten signature verification: New advancements and open issues. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 367–372. IEEE. (Cited on page 32.)

Jain, A., Hong, L., and Pankanti, S. (2000). Biometric identification. *Communications of the ACM*, 43(2):90–98. (Cited on page 5.)

Jain, A. K., Ross, A. A., and Nandakumar, K. (2011). *Introduction to biometrics*. Springer Science & Business Media. (Cited on pages 19, 2, and 3.)

Justino, E. J., Bortolozzi, F., and Sabourin, R. (2001). Off-line signature verification using hmm for random, simple and skilled forgeries. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pages 1031–1034. IEEE. (Cited on page 27.)

Justino, E. J., El Yacoubi, A., Bortolozzi, F., and Sabourin, R. (2000). An off-line signature verification system using hmm and graphometric features. In *Proc. of the 4th International Workshop on Document Analysis Systems*, pages 211–222. (Cited on page 27.)

Kalera, M. K., Srihari, S., and Xu, A. (2004). Offline signature verification and identification using distance statistics. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(07):1339–1360. (Cited on pages 10 and 89.)

Kessentini, Y., BenAbderrahim, S., and Djeddi, C. (2018). Evidential combination of svm classifiers for writer recognition. *Neurocomputing*, 313:1–13. (Cited on page 26.)

Leclerc, F. and Plamondon, R. (1994). Automatic signature verification: The state of the art—1989–1993. *Progress in Automatic Signature Verification*, pages 3–20. (Cited on pages 5, 27, and 32.)

Liu, P., Guo, J.-M., Chamnongthai, K., and Prasetyo, H. (2017). Fusion of color histogram and lbp-based features for texture image retrieval and classification. *Information Sciences*, 390:95–111. (Cited on page 8.)

Liwicki, M., Malik, M. I., Alewijnse, L., v. d. Heuvel, E., and Found, B. (2012). Icfhr 2012 competition on automatic forensic signature verification (4NsigComp 2012). In *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 823–828. (Cited on page 32.)

Liwicki, M., Malik, M. I., Van Den Heuvel, C. E., Chen, X., Berger, C., Stoel, R., Blumenstein, M., and Found, B. (2011). Signature verification competition for online and offline skilled forgeries (sigcomp2011). In *2011 International conference on document analysis and recognition*, pages 1480–1484. IEEE. (Cited on pages 6 and 7.)

Liwicki, M., Schlapbach, A., Bunke, H., Bengio, S., Mariéthoz, J., and Richiardi, J. (2006). Writer identification for smart meeting room systems. In *International Workshop on Document Analysis Systems*, pages 186–195. Springer. (Cited on page 22.)

Loka, H., Zois, E., and Economou, G. (2017). Long range correlation of preceded pixels relations and application to off-line signature verification. *IET Biometrics*, 6(2):70–78. (Cited on page 46.)

Louloudis, G., Gatos, B., and Stamatopoulos, N. (2012). Icfhr 2012 competition on writer identification challenge 1: Latin/greek documents. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 829–834. IEEE. (Cited on page 41.)

Louloudis, G., Stamatopoulos, N., and Gatos, B. (2011). ICDAR 2011 writer identification contest. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1475–1479. IEEE. (Cited on page 41.)

Lu, X., Huang, L., and Yin, F. (2021). Cut and compare: End-to-end offline signature verification network. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3589–3596. IEEE. (Cited on pages 29 and 35.)

Lumini, A. and Nanni, L. (2009). Ensemble of on-line signature matchers based on over-complete feature generation. *Expert Systems with Applications*, 36(3, Part 1):5291 – 5296. (Cited on page 47.)

Maergner, P., Howe, N., Riesen, K., Ingold, R., and Fischer, A. (2018a). Offline signature verification via structural methods: Graph edit distance and inkball models. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 163–168. IEEE. (Cited on pages 29, 35, and 91.)

Maergner, P., Pondenkandath, V., Alberti, M., Liwicki, M., Riesen, K., Ingold, R., and Fischer, A. (2018b). Offline signature verification by combining graph edit distance and triplet networks. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 470–480. Springer. (Cited on pages 29, 35, 90, and 91.)

Maergner, P., Riesen, K., Ingold, R., and Fischer, A. (2017). A structural approach to offline signature verification using graph edit distance. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1216–1222. IEEE. (Cited on page 91.)

Maiorana, E. (2010). Biometric cryptosystem using function based on-line signature recognition. *Expert Systems with Applications*, 37(4):3454 – 3461. (Cited on page 27.)

Malik, M. I. (2015). *Automatic Signature Verification: Bridging the Gap between Existing Pattern Recognition Methods and Forensic Science*. PhD thesis. (Cited on pages 28 and 50.)

Malik, M. I., Ahmed, S., Marcelli, A., Pal, U., Blumenstein, M., Alewijns, L., and Liwicki, M. (2015). ICDAR2015 competition on signature verification and writer identification for on- and off-line skilled forgeries (sigwicomp2015). In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1186–1190. (Cited on pages 21, 32, 41, 50, 57, 59, 60, and 61.)

Malik, M. I., Liwicki, M., Alewijnse, L., Ohyama, W., Blumenstein, M., and Found, B. (2013). Icdar 2013 competitions on signature verification and writer identification for on- and offline skilled forgeries (sigwicomp 2013). In *12th International Conference on Document Analysis and Recognition*, pages 1477–1483. (Cited on pages 32, 41, and 50.)

Mudda, M., Manjunath, R., and Krishnamurthy, N. (2020). Brain tumor classification using enhanced statistical texture features. *IETE Journal of Research*, pages 1–12. (Cited on page 22.)

Nanni, L., Ghidoni, S., and Brahnam, S. (2017). Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, 71:158–172. (Cited on page 8.)

Nanni, L., Maiorana, E., Lumini, A., and Campisi, P. (2010). Combining local, regional and global matchers for a template protected on-line signature verification system. *Expert Systems with Applications*, 37(5):3676 – 3684. (Cited on page 27.)

Nosary, A., Heutte, L., and Paquet, T. (2004). Unsupervised writer adaptation applied to handwritten text recognition. *Pattern Recognition*, 37(2):385–388. (Cited on page 22.)

Novitasari, D. C. R., Lubab, A., Sawiji, A., and Asyhar, A. H. (2019). Application of feature extraction for breast cancer using one order statistic, glcm, glrlm, and gldm. *Advances in Science, Technology and Engineering Systems Journal*, 4(4):115–120. (Cited on page 22.)

Okawa, M. (2016a). Offline signature verification based on bag-of-visual words model using kaze features and weighting schemes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 184–190. (Cited on page 31.)

Okawa, M. (2016b). Vector of locally aggregated descriptors with kaze features for offline signature verification. In *2016 IEEE 5th Global Conference on Consumer Electronics*, pages 1–5. IEEE. (Cited on page 30.)

Piekarczyk, M. (2010). Hierarchical random graph model for off-line handwritten signatures recognition. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on*, pages 860–865. IEEE. (Cited on page 27.)

Pirlo, G. (1994). Algorithms for signature verification. In *Fundamentals in Handwriting Recognition*, pages 435–454. Springer. (Cited on page 5.)

Plamondon, R. (1994). *Progress in automatic signature verification*, volume 13. World Scientific. (Cited on page 5.)

Plamondon, R. (1995). A kinematic theory of rapid human movements: Part i. movement representation and generation. *Biological cybernetics*, 72(4):295–307. (Cited on page 5.)

Plamondon, R. (1998). A kinematic theory of rapid human movements: Part iii. kinetic outcomes. *Biological Cybernetics*, 78(2):133–145. (Cited on page 5.)

Plamondon, R. and Lorette, G. (1989). Automatic signature verification and writer identification—the state of the art. *Pattern recognition*, 22(2):107–131. (Cited on pages 5, 27, and 32.)

Plamondon, R. and Srihari, S. N. (2000). Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1):63–84. (Cited on pages 5, 27, and 32.)

Radhika, K. and Gopika, S. (2015). Online and offline signature verification: a combined approach. *Procedia Computer Science*, 46:1593–1600. (Cited on page 5.)

Rantzsch, H., Yang, H., and Meinel, C. (2016). Signature embedding: Writer independent offline signature verification with deep metric learning. In *International symposium on visual computing*, pages 616–625. Springer. (Cited on pages 31 and 33.)

Rasti, B., Hong, D., Hang, R., Ghamisi, P., Kang, X., Chanussot, J., and Benediktsson, J. A. (2020). Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox. *IEEE Geoscience and Remote Sensing Magazine*, 8(4):60–88. (Cited on page 98.)

Sae-Bae, N. and Memon, N. (2014). Online signature verification on mobile devices. *IEEE transactions on information forensics and security*, 9(6):933–947. (Cited on page 98.)

Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., Buhmann, J., Maass, W., Ritter, H., and Tishby, N. (1999). Single-class support vector machines. (Cited on pages 46 and 47.)

Serdouk, Y., Nemmour, H., and Chibani, Y. (2014). Combination of oc-lbp and longest run features for off-line signature verification. In *2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems*, pages 84–88. IEEE. (Cited on pages 32 and 36.)

Serdouk, Y., Nemmour, H., and Chibani, Y. (2016). New off-line handwritten signature verification method based on artificial immune recognition system. *Expert Systems with Applications*, 51(Supplement C):186 – 194. (Cited on pages 27, 31, 34, and 36.)

Sharif, M., Khan, M. A., Faisal, M., Yasmin, M., and Fernandes, S. L. (2020). A framework for offline signature verification system: Best features selection approach. *Pattern Recognition Letters*, 139:50–59. (Cited on pages 90 and 91.)

Siddiqi, I., Cloppet, F., and Vincent, N. (2009). Contour based features for the classification of ancient manuscripts. In *Conference of the International Graphonomics Society*, pages 226–229. (Cited on page 22.)

Soleimani, A., Araabi, B. N., and Fouladi, K. (2016). Deep multitask metric learning for offline signature verification. *Pattern Recognition Letters*, 80:84–90. (Cited on pages 31 and 33.)

Souza, V. L., Oliveira, A. L., Cruz, R. M., and Sabourin, R. (2020). Improving bpso-based feature selection applied to offline wi handwritten signature verification through overfitting control. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pages 69–70. (Cited on pages 29 and 35.)

Souza, V. L., Oliveira, A. L., and Sabourin, R. (2018). A writer-independent approach for offline signature verification using deep convolutional neural networks features. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 212–217. IEEE. (Cited on pages 8, 30, and 35.)

Srihari, S. N., Xu, A., and Kalera, M. K. (2004). Learning strategies and classification methods for off-line signature verification. In *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*, pages 161–166. IEEE. (Cited on page 27.)

Stauffer, M., Maergner, P., Fischer, A., and Riesen, K. (2019). Graph embedding for offline handwritten signature verification. In *Proceedings of the 2019 3rd International Conference on Biometric Engineering and Applications*, pages 69–76. (Cited on pages 29 and 35.)

Suwanwiwat, H., Das, A., Ferrer, M. A., Pal, U., and Blumenstein, M. (2017). An automatic student verification system utilising off-line thai name components. In *2017 International conference on digital image computing: techniques and applications (DICTA)*, pages 1–6. IEEE. (Cited on page 63.)

Suwanwiwat, H., Das, A., Pal, U., and Blumenstein, M. (2018). Icfhr 2018 competition on thai student signatures and name components recognition and verification (tsncrv2018). In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 500–505. IEEE. (Cited on pages 21, 8, 9, 10, 40, 57, 62, 63, 71, 73, 74, 76, 78, 80, and 92.)

Tang, X. (1998). Texture information in run-length matrices. *IEEE transactions on image processing*, 7(11):1602–1609. (Cited on page 18.)

Tang, Y. and Wu, X. (2016). Text-independent writer identification via cnn features and joint bayesian. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 566–571. IEEE. (Cited on page 8.)

Van, B. L., Garcia-Salicetti, S., and Dorizzi, B. (2007). On using the viterbi path along with hmm likelihood information for online signature verification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1237–1247. (Cited on page 27.)

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media. (Cited on page 47.)

Vargas, F., Ferrer, M., Travieso, C., and Alonso, J. (2007). Off-line handwritten signature gpds-960 corpus. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 764–768. IEEE. (Cited on pages 19, 9, 40, 49, 73, and 89.)

Veeramachaneni, K., Osadciw, L. A., and Varshney, P. K. (2005). An adaptive multimodal biometric management algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(3):344–356. (Cited on page 5.)

Vielhauer, C. and Dittmann, J. (2006). Biometrics for user authentication. encyclopedia of multimedia, ed. b. furth. (Cited on page 5.)

Wang, K., Wang, Y., and Zhang, Z. (2011). On-line signature verification using graph representation. In *Image and Graphics (ICIG), 2011 Sixth International Conference on*, pages 943–948. IEEE. (Cited on page 27.)

Xu, S. S.-D., Chang, C.-C., Su, C.-T., and Phu, P. Q. (2019). Classification of liver diseases based on ultrasound image texture features. *Applied Sciences*, 9(2):342. (Cited on page 21.)

Yilmaz, M. B., Yanikoglu, B., Tirkaz, C., and Kholmatov, A. (2011). Offline signature verification using classifier combination of hog and lbp features. In *Biometrics (IJCB), 2011 International Joint Conference on*, pages 1–7. IEEE. (Cited on pages 59 and 60.)

Zheng, Y., Iwana, B. K., Malik, M. I., Ahmed, S., Ohyama, W., and Uchida, S. (2021). Learning the micro deformations by max-pooling for offline signature verification. *Pattern Recognition*, 118:108008. (Cited on pages 28 and 34.)

Zois, E. N., Alewijnse, L., and Economou, G. (2016). Offline signature verification and quality characterization using poset-oriented grid features. *Pattern Recognition*, 54:162–177. (Cited on pages 30 and 33.)

Zois, E. N., Theodorakopoulos, I., Tsourounis, D., and Economou, G. (2017). Parsimonious coding and verification of offline handwritten signatures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 134–143. (Cited on page 31.)

Zois, E. N., Zervas, E., Tsourounis, D., and Economou, G. (2020). Sequential motif profiles and topological plots for offline signature verification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13248–13258. (Cited on pages 28 and 35.)