People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Echahid Cheikh Larbi Tebessi University – Tebessa

Faculty of Exact Sciences and Natural and Life Sciences

Department of Computer Science

Thesis

Presented in fulfillment of the requirements for the MASTER's degree

Field: Computer Science

Option: Information Systems

Title

Offline Writer Identification System Using Textural Descriptors

By

Farouk DJABRI

Jury members

Dr. Issam BENDIB	Echahid Cheikh Larbi Tebessi University	**President**
Dr. Assia BREK	Echahid Cheikh Larbi Tebessi University	**Examiner**
Dr. Walid BOUAMRA	Echahid Cheikh Larbi Tebessi University	**Supervisor**
Pr. Abdeljalil GATTAL	Echahid Cheikh Larbi Tebessi University	**Co-Supervisor**

Academic year: 2024/2025

Acknowledgments

I would first of all like to express my gratitude to my supervisors, Dr. Walid BOUAMRA and Pr. Abdeljalil GATTAL, for their constant encouragement, advice, and support throughout the course of this research. Their precious comments and experience have been appreciated immensely by the realization and completion of this work.

My highest appreciation goes to my family, whose unwavering support, resilience, and belief in me have always been a motivating force. Their concern and support helped me overcome all the challenges along the way.

I also extend my appreciation to my friends and colleagues, especially those who provided technical assistance, shared knowledge, and encouraged an effective research atmosphere.

Lastly, I would like to thank everybody who, directly or indirectly, contributed to this work's success.

Dedication

..To my dear family

Abstract

This thesis presents a novel approach for offline writer identification by applying textural descriptors to extract the unique writing style of individuals from static samples of handwriting. Offline writer identification is a challenging biometric problem due to the variability in handwriting caused by emotional, physical, and environmental factors, as well as missing dynamic writing details such as stroke order or pressure, to address these challenges, this manuscript investigates Run-Length-based feature extraction techniques, with a focus on Run-Length statistics.

Our study begins by investigating traditional Run-Length (RL) descriptors, which are used to describe the distribution and repetition of binary patterns or runs in handwriting images. We enhance their discriminative ability with a new descriptor Z-Run-Length (Z-RL) that traverses images along a zigzag course to detect more subtle structural differences in handwriting.

Furthermore, we evaluate a hybrid approach that combines baseline Run-Length features with the proposed Z-RL descriptor. The resulting feature vectors are classified using machine learning algorithms such as Support Vector Machines (SVM) and K-Nearest Neighbors (K-NN), and are validated using robust evaluation protocols, including k-fold cross-validation and Leave-One-Out Cross-Validation (LOOCV).

Experimental results on the BFL dataset demonstrate that the proposed system achieves high accuracy and exhibits strong consistency across diverse handwriting samples.

Keywords: *Offline Writer Identification, Textural Descriptors, Feature Extraction, Run-length-features, Z-Run-Length features, Classification, Evaluation.*

ملخص

تعرض هذه الأطروحة مقاربة جديدة للتعرف على هوية الكاتب في الوضع غير المتصل، من خلال تطبيق واصفات نسيجية لاستخراج أسلوب الكتابة الخاص بكل فرد انطلاقًا من عينات ثابتة من الكتابة اليدوية. يُعد التعرف على الكاتب في الوضع غير المتصل مشكلة بيومترية صعبة بسبب التغيرات الكبيرة في الكتابة اليدوية الناتجة عن عوامل عاطفية وجسدية وبيئية، بالإضافة إلى غياب التفاصيل الديناميكية للكتابة مثل ترتيب الحركات أو الضغط المطبق أثناء الكتابة. لمواجهة هذه التحديات، تبحث هذه الأطروحة تقنيات استخراج الخصائص المعتمدة على طول القطعة (*Run-Length*)، مع التركيز على الإحصائيات الخاصة بطول القطع.

تبدأ دراستنا بتحليل الواصفات التقليدية لطول القطع (Run-Length (RL، والتي تُستخدم لوصف توزيع وتكرار الأنماط الثنائية أو القطع في صور الكتابة اليدوية. ونعزز قدرتها التمييزية من خلال واصف جديد يُدعى Z-Run-Length (Z-RL)، والذي يقوم بمسح الصور وفق مسار متعرج لاكتشاف اختلافات بنيوية أكثر دقة في الكتابة.

علاوة على ذلك، ننشئ مقاربة هجينةً تجمع بين الخصائص الأساسية لطول القطع RL والواصف المقترحZ-RL . يتم تصنيف متجهات الخصائص الناتجة باستخدام خوارزميات التعلم الآلي مثل أدوات التصنيف (SVM) و(K-NN) ، ويتم التحقق من صحتها باستخدام بروتوكولات تقييم قوية، بما في ذلك التحقق المتقاطع k-fold والتحقق المتقاطع (LOOCV).

تُظهر النتائج التجريبية على قاعدة البيانات BFL أن النظام المقترح يحقق دقة عالية ويُظهر درجة كبيرة من التناسق عبر عينات متنوعة من الكتابة اليدوية.

الكلمات المفتاحية : *التعرف على الكاتب في الوضع غير المتصل، الواصفات النسيجية، استخراج الخصائص، خصائص طول القطع Run-Length، خصائص Z-Run-Length ، التصنيف، التقييم.*

Résumé

Cette thèse présente une nouvelle approche pour l'identification du scripteur hors ligne, en appliquant des descripteurs texturaux afin d'extraire le style d'écriture unique de chaque individu à partir d'échantillons statiques d'écriture manuscrite. L'identification du scripteur hors ligne constitue un problème biométrique complexe en raison de la variabilité de l'écriture manuscrite causée par des facteurs émotionnels, physiques et environnementaux, ainsi que par l'absence de détails dynamiques tels que l'ordre des traits ou la pression exercée. Pour relever ces défis, ce manuscrit étudie des techniques d'extraction de caractéristiques basées sur la longueur du segment, en se concentrant sur les statistiques de longueur des segments.

Notre étude commence par l'analyse des descripteurs traditionnels de longueur des segments Run-Length (RL), utilisés pour décrire la distribution et la répétition de motifs binaires ou de segments dans les images d'écriture manuscrite. Nous améliorons leur capacité discriminante grâce à un nouveau descripteur, Z-Run-Length (Z-RL), qui parcourt les images selon un chemin en zigzag afin de détecter des différences structurelles plus subtiles dans l'écriture.

En outre, nous évaluons une approche hybride combinant les caractéristiques de base de la longueur du segment avec le descripteur Z-RL proposé. Les vecteurs de caractéristiques obtenus sont classés à l'aide d'algorithmes d'apprentissage automatique tels que les machines à vecteurs de support (SVM) et les k-plus proches voisins (K-NN), et sont validés à l'aide de protocoles d'évaluation robustes, notamment la validation croisée k-fold et la validation croisée Leave-One-Out (LOOCV).

Les résultats expérimentaux sur la base de données BFL démontrent que le système proposé atteint une grande précision et présente une forte cohérence entre divers échantillons d'écriture manuscrite.

Mots-clés *: Identification du scripteur hors ligne, Descripteurs texturaux, Extraction de caractéristiques, Caractéristiques de longueur de segment Run-Length, Caractéristiques Z-Run-Length, Classification, Évaluation.*

Table of content

Acknowledgments.....II

Dedication.....III

Abstract.....IV

ملخص.....V

Résumé.....VI

List of figures.....XI

List of tables.....XII

Introduction.....1

Chapter **01**: General Context and Theoretical Foundations.....5

1.1 Introduction.....6

1.2 Writer Identification.....6

1.3 Types of Handwriting Recognition.....7

1.4 Writer identification and verification.....8

1.4.1 Writer identification.....8

1.4.2 Writer verification.....9

1.5 Application Domains.....10

1.6 General Architecture of a Writer Identification System.....11

1.6.1 Preprocessing.....12

1.6.2 Feature Extraction.....13

1.6.3 Classification.....14

1.6.4 Evaluation and Validation.....16

1.7 Handwriting Image Datasets.....17

1.7.1 Description of Dataset.....17

1.7.2 BFL.....17

1.7.3 CVL 19
1.7.4 IAM 19
1.7.5 KHAT 20
1.8 Image Characteristics and Formats 20
1.9 Challenges in Realistic Handwritten Data 21
1.10 Conclusion 21
Chapter 02: State of The Art 23
2.1 Introduction 24
2.2 Related works 24
2.3 State of the art Summary 36
2.4 Conclusion 41
Chapter 03: Proposed Approach 42
3.1 Introduction 43
3.2 Run_length Method 44
3.2.1 Description 44
3.2.2 History of Run_length 44
3.2.3 Run-length example and application 45
3.3 Our proposed approach (Z-Run-Length) 48
3.3.1 Motivation and Intuition Behind the Proposal 49
3.3.2 Exploring Z-shaped scanning 50
3.3.3 Benefits over classical row/column-based traversals 50
3.4 Z and Reverse-Z Traversal Definitions 50
3.4.1 Z Traversal 51
3.4.2 Reverse-Z Traversal 51
3.5 Algorithmic details of Z and RevZ scanning 51

3.5.1 Z-Traversal 51
3.5.2 Equations for Z traversal 52
3.5.3 Reverse-Z Traversal 52
3.5.4 Pseudocode: Z-Traversal 53
3.5.5 Pseudocode: Reverse-Z Traversal 54
3.6 Z-Run-length steps 56
3.6.1 Image Binarization 56
3.6.2 Generating Traversal Sequences 56
3.6.3 Run-Length Encoding Algorithm 57
3.6.4 Construction of the ZRL Feature Vector 58
3.6.5 Concatenation of Directional Vectors 59
3.7 Sample output visualization 60
3.8 Normalization techniques tested 61
3.9 Conclusion 62
Chapter **0**4: Experimental Results 63
4.1 Introduction 64
4.2 Classification 64
4.3 Dataset used 65
4.4 Experimental Protocol 65
4.5 Experimental results 66
4.5.1 With Classical RL 66
4.5.2 Experimental results of Z-Run-length and ZRL+RL 68
4.6 Discussion and Performance Analysis 70
Conclusion 72
Summary of contributions 73

Final results and achievements 74
Limitations encountered..... 75
Research perspectives 76
References..... 77

List of figures

Figure 1 Writer identification 8
Figure 2 Writer verification 9
Figure 3 General architecture of writer identification system 11
Figure 4 image binarization and reduce noise 12
Figure 5 feature extraction 14
Figure 6 Classification diagram 15
Figure 7 Example from BFL Dataset 18
Figure 8 Example from KHATT Dataset 20
Figure 9 Proposed method for the cutting of writing. 25
Figure 10 Block diagram of the writer identification system. 26
Figure 11 Offline writer identification method based on SIFT 27
Figure 12 Calculating the run-length matrix 28
Figure 13 Natural corpus-driven codebook generation 30
Figure 14 Steps of the proposed method. 30
Figure 15 Overview of the writer identification using implicit codebook of shapes 34
Figure 16 Update of run-length during history 45
Figure 17 Example of a binary image 46
Figure 18 Apply run length on binary image 46
Figure 19 Z traversal (left) and Reverse-Z traversal (right) scanning patterns 57
Figure 20 Binary image 4×4 60
Figure 21 Example of using Z traversal and Reverse-Z traversal on a given image 60

List of tables

Table 1 Description of BFL Dataset 18

Table 2 Summary of the State of the art..... 36

Table 3 SVM and K-NN works 40

Table 4 Related results of other works with RL 67

Table 5 Results of the classical run-length features..... 68

Table 6 Comparison of ZRL and Combined Descriptors..... 69

Table 7 Best results using ZRL and [ZRL,RL] 70

Introduction

For many years, all categories of basic and applied research have examined the identification of writers through handwriting. Professionals in the field including forensic analysts [1], questioned document examiners, and paleographers, have long been relying on opinions based on experience to assess handwriting exemplars for authenticity in writing documents, identify writing styles, even assess the potential age of historic manuscripts. Now that image processing and machine learning are available, these types of manual tasks can be adapted to be more scalable and automated.

Handwriting is a physical biometric trait that exhibits unique behavior and physiological characteristics of a person. The need to identify writers from handwritten documents has emerged in fields such as forensic science, historical manuscript analysis, document authentication, and biometric security systems over time. With the improved availability of documents scanned and technological advancements in image processing and machine learning resources, offline writer identification has emerged as a means to automate and develop traditional handwriting analysis techniques. In this context, texture-based analysis methods have proven potential because they identify intricate visual patterns and styles associated with an individual's handwriting [3] [5].

This study concerns such a computerized approach to offline writer identification, a challenging and significant problem in behavioral biometrics, writer identification attempts to find out who wrote a handwriting sample from a set of known writers. It is a challenging pattern recognition problem, exacerbated by challenges such as an enormous number of potential writers, limited training samples, and intra-writer variability. Handwriting is a unique motor act shaped by idiosyncratic tendencies and deviation from normalized writing models over the years. Such small writer-specific things such as individual allographs and connection types form the basis of distinguishing individuals [5].

Based on the acquisition modality, writer identification systems are generally considered to be online and offline techniques. Dynamic features such as stroke speed and pen pressure are captured by online systems through digital tablets or sensors, while offline systems analyze static, scanned handwritten information. The offline technique is most effective in real-world applications with pre-existing documents or scanned documents, where dynamic details cannot be retrieved. Further, writer identification can either be text-dependent, requiring writers to produce similar text,

or text-independent, where the system must identify the writer from random text that is nearer to actual application cases [4].

Despite advances in feature extraction and classification techniques, offline writer identification remains a challenging pattern recognition problem. Variations in handwriting due to emotional state, writing instrument, paper type, and rate of writing lead to large intra-writer variation, and variation among different writers leads to inter-writer confusion. Moreover, in real-world applications, systems must operate in text-independent mode, where the input text content is unknown and unrestricted. The existing methods, including hand-crafted descriptors like Run-Length features and learned features using CNNs, have limited generalizability, noise robustness, and performance across datasets.

The primary objective of this work is to develop an effective offline writer identification system using effective textural descriptors in order to detect the variability of writing styles of various individuals. Specifically, the study aims to:

- Investigate and assess the effectiveness of classical and run-length texture features for offline handwriting.
- Propose an improved variant of run-length descriptor (Z_RL) for enhanced discrimination.
- Combine different types of handcrafted features to improve the overall system performance (RL+Z_RL).
- Evaluate the proposed system across standard public handwriting datasets in text-independent scenarios, using different classifiers and validation protocols.

This paper presents the design and development of a Machine Learning-based Offline Writer Identification System. The system uses a blend of textural and structural feature extraction techniques, preprocessing algorithms, and supervised classifiers for effective writer identification from scanned handwriting. With machine learning, the system can learn intra- and inter-writer variations of any order without human intervention, yielding high accuracy and robustness. The use of modern classifiers with standard texture features enhances performance and interpretability.

This thesis is structured into five chapters, each addressing a central part of the offline writer identification research. Chapter 1 establishes the general context and theory of the study, provides

definitions, defines salient issues in writer identification, and discusses typical system architecture. It also offers descriptions of typical handwriting datasets and their characteristics respectively. Chapter 2 gives a detailed state of the art, reviewing significant earlier work in the field and evaluating a number of alternative approaches, paying particular attention to textural descriptors in handwriting analysis. Chapter 3 introduces the new approach, presenting the conventional Run-Length (RL) method together with the novel Zigzag Run-Length (ZRL) descriptor that is able to identify more subtle changes in structure. This chapter also includes a description of the BFL dataset used in the experiments. Chapter 4 provides an overview of the experimental procedure, including feature extraction details, classification on Support Vector Machines (SVM) and K-Nearest Neighbors (KNN), and evaluation setup based on fixed training/testing splits and Leave-One-Out Cross Validation (LOOCV). It illustrates the results obtained with RL, ZRL, and their combination, and ends by making comparisons with state-of-the-art approaches. Finally, the thesis concludes with a last conclusion summarizing the key contributions and findings, and pointing out possible future work directions within the offline writer identification field.

Chapter 01

General Context and Theoretical Foundations

1.1 Introduction

Author identification using handwriting has been an area of interest in most fields for a very long period, from forensic science to document analysis and behavioral biometrics. With more amounts of digitized documents and advances in machine learning and pattern recognition, author identification has emerged as a significant area of research under document image analysis. This chapter aims to set the background and theory of offline writer identification, with the most appropriate concepts, methods, and resources in view [4] [5].

The first subsection introduces the fundamental definitions and problems pertaining to writer identification, i.e., online and offline handwriting recognition, identification and verification tasks, etc. It also refers to the diversified application fields in which writer identification systems are of crucial significance, from forensic and authentication applications to digital archiving.

The second part explains the high-level architecture of a writing identification system and talks about the main processing steps involved in preprocessing, feature extraction, classification, and evaluation. All of them are addressing specific issues in transforming raw handwriting images into discriminative representations for accurate classification.

The last section of the book describes the handwriting image datasets that have been commonly employed in the research community. It discusses the characteristics and structures of common benchmark datasets, e.g., IAM, CVL, and BFL, and highlights the importance of diversity and realism of the datasets for the development of robust writer identification systems. Particular emphasis is given to realistic handwritten data challenges, e.g., handwriting style variability, acquisition noise, and document layout inconsistencies.

1.2 Writer Identification

Writer identification means the determination of an individual's identity based on his/her handwriting. Writer identification is a subset of the broader field of behavioral biometrics, which captures distinctive, repeatable patterns of human behavior. Unlike physiological biometrics such as fingerprints or iris scans, handwriting is a learned motor skill that develops in a distinctive manner in each person over time. Such writing habits are conveyed by stable, writer-dependent features such as stroke direction, pressure, character inter-space, and allographic preferences. From

these features, a writer identification system can match a handwriting sample to its author within a predefined set of candidates [1] [4].

Writer identification is confronted with several challenges, particularly for offline handwritten data. One of the significant challenges is intra-writer variability since an individual's handwriting also changes depending on emotional state, writing speed, writing instrument, or writing surface. Intra-writer variability presents difficulties to obtain consistent, discriminative features from the handwriting. Another of the primary difficulties is inter-writer similarity, where different individuals may share visually similar handwriting, especially for shared cultural or educational backgrounds. These similarities may lead to misclassification. Image quality issues such as noise, poor scans, background patterns (e.g., ruled paper), and ink smudging can also negatively impact preprocessing and feature extraction stages. Multilingual and multi-script documents present an additional complexity, where language-independent or script-adaptive feature extraction methods are required. Finally, limited labeled data per author under real-world conditions poses difficulty in training robust machine learning or deep learning models, particularly if supervised approaches are employed [2] [3].

1.3 Types of Handwriting Recognition

Handwriting recognition systems are generally divided into two general categories based on the strategy of data acquisition: online and offline recognition. This distinction has a vital influence on handwriting data acquisition, processing, and analysis for applications such as writer identification [4] [5].

Online handwriting recognition involves the capture of dynamic writing information in real time using digitizing devices such as tablets, smart pens, or styli. The systems record temporal and spatial information such as pen pressure, stroke speed, pen orientation, and coordinate order as the author forms characters. The temporal feature richness makes online systems extremely robust and accurate, especially in uses like handwriting recognition and dynamic signature verification. But there is a special hardware requirement that limits the application of web-based methods for most real-world situations [5] [6] [7].

In contrast, offline handwriting recognition deals with static handwriting data obtained by scanning handwritten documents. Only the final image of the written text exists, but not dynamic stroke information. Offline recognition systems thus must rely solely on visual features that can

be obtained from the handwriting image, such as texture, shape, and structure. While offline systems are harder due to the lack of temporal data, they are more practical and more widely applicable, with particular significance in forensic analysis, historical document analysis, and processed document digitization. Offline writer identification, which operates on scanned handwriting images and tries to identify the writer using static visual information only, is the focus of the present thesis [5] [8] [9].

1.4 Writer identification and verification

In handwriting-based biometric systems, two closely related but fundamentally different tasks are typically addressed: writer identification and writer verification. Though both aim to analyze handwriting to draw conclusions about the writer, they serve distinctly different purposes, take different approaches, and have different system designs [5].

1.4.1 Writer identification

Writer identification is a multi-class classification issue in which the objective is to establish the identity of the writer among a group of known writers. For a handwriting sample, the system matches it with a database of registered writers and identifies it with the most probable candidate, Figure 1 illustrates the general workflow of a writer identification system, from input acquisition to final classification.

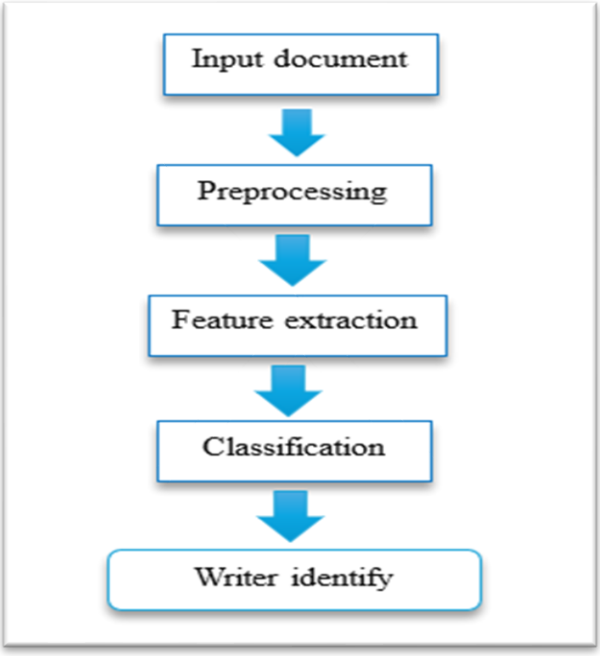


Figure 1 Writer identification

This task is common in such applications as forensic examination, document authorship analysis, and biometric security systems where the writer's identity is unknown and needs to be identified [2] [4].

1.4.2 Writer verification

On the other hand, writer verification is a two-class classification problem. The purpose is to confirm or reject an claimed identity through comparison of a test sample and a reference sample. The system will determine whether both samples originate from the same writer. Writer verification will typically be used in security-sensitive applications, e.g., legal or bank signature verification, where identity is claimed and must be confirmed [10] [1], Figure 2 illustrates the writer verification process, highlighting the comparison between input and reference samples.

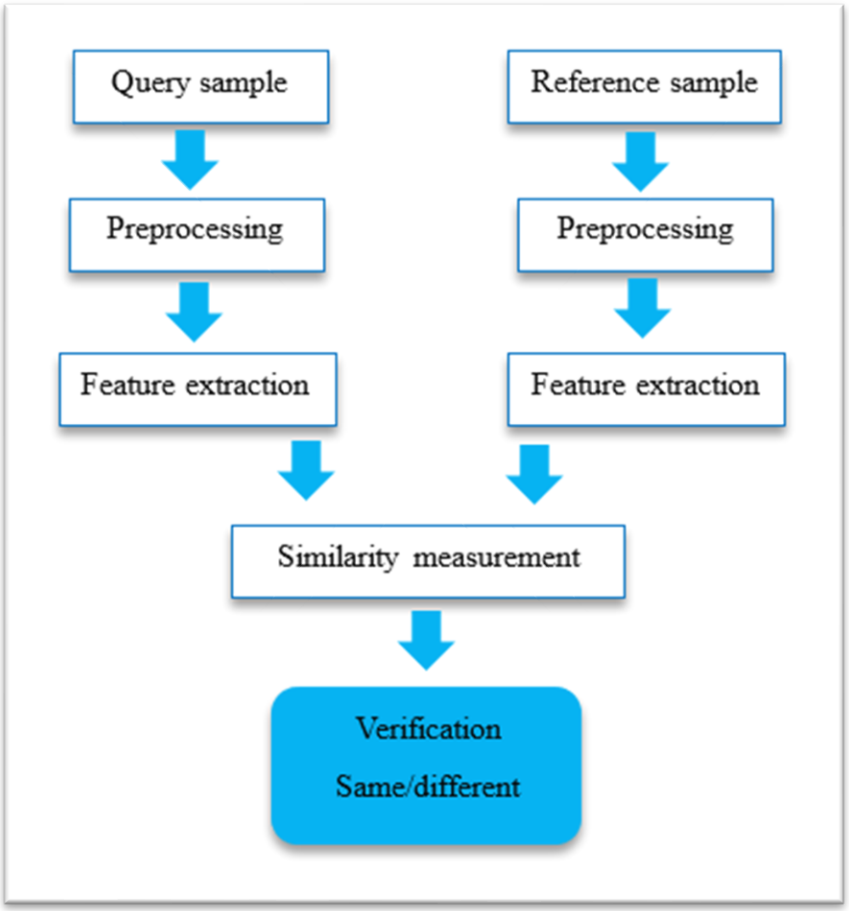


Figure 2 Writer verification

Although distinct, identification and verification share numerous aspects, including feature extraction and pattern matching algorithms. In the majority of cases, a writer identification system can be adapted to perform verification by incorporating a similarity threshold into the matching scores. However, verification systems typically demand higher accuracy and lower false acceptance rates due to security implications of incorrect classifications. Understanding the distinction between these two tasks is important in the design of systems with specific application requirements [11] [5].

1.5 Application Domains

Writer identification and verification systems have significant applications in several fields such as :

- forensics
- biometrics
- document analysis.

Handwriting is used by these systems as a unique behavioral trait to support applications ranging from individual authentication to authorship attribution and historical research [12].

In the field of forensics, writer identification is used to examine questioned documents in criminal investigations, court cases, and detection of fraud. Handwriting constitutes the most important evidence to link an offender to an offense or to ascertain the authenticity of a document. Automated writer identification systems aid forensics experts by providing objective, fact-based tools for processing large volumes of handwriting with improved consistency and speed [13] [14].

In biometric applications, handwriting, especially signatures, is utilized as a type of identity verification. Writer verification systems are employed in various security-limited environments such as banking, e-commerce, and secure document pickup. As compared to physiological biometrics, handwriting-based systems are less intrusive and may be used using simple input devices or documents scanned, thus convenient both for online and offline use [15] [10] [11].

In the application area of document analysis, author identification is very important in authorship attribution of historical documents, document clustering of authors in archive management systems, and facilitating research in digital humanities. Detection of authors for

anonymous and sensitive documents can help with historical interpretation, conservation, and cataloging [16] [17] [18].

These application domains illustrate the broad utility of writer identification systems and emphasize the need for accurate, scalable, and adaptable solutions, especially as digital transformation continues across various sectors.

1.6 General Architecture of a Writer Identification System

A writer identification system comprises several key stages that cumulatively transform raw handwriting samples into representative formats for classification. The system's success relies heavily on the proper design and concatenation of the stages, viz., preprocessing, feature extraction, classification, and evaluation and validation. Each stage addresses some problems in processing offline handwriting data and aids in building an effective and efficient identification pipeline [3] [16], Figure 3 provides a visual overview of the typical workflow involved in a writer identification system.

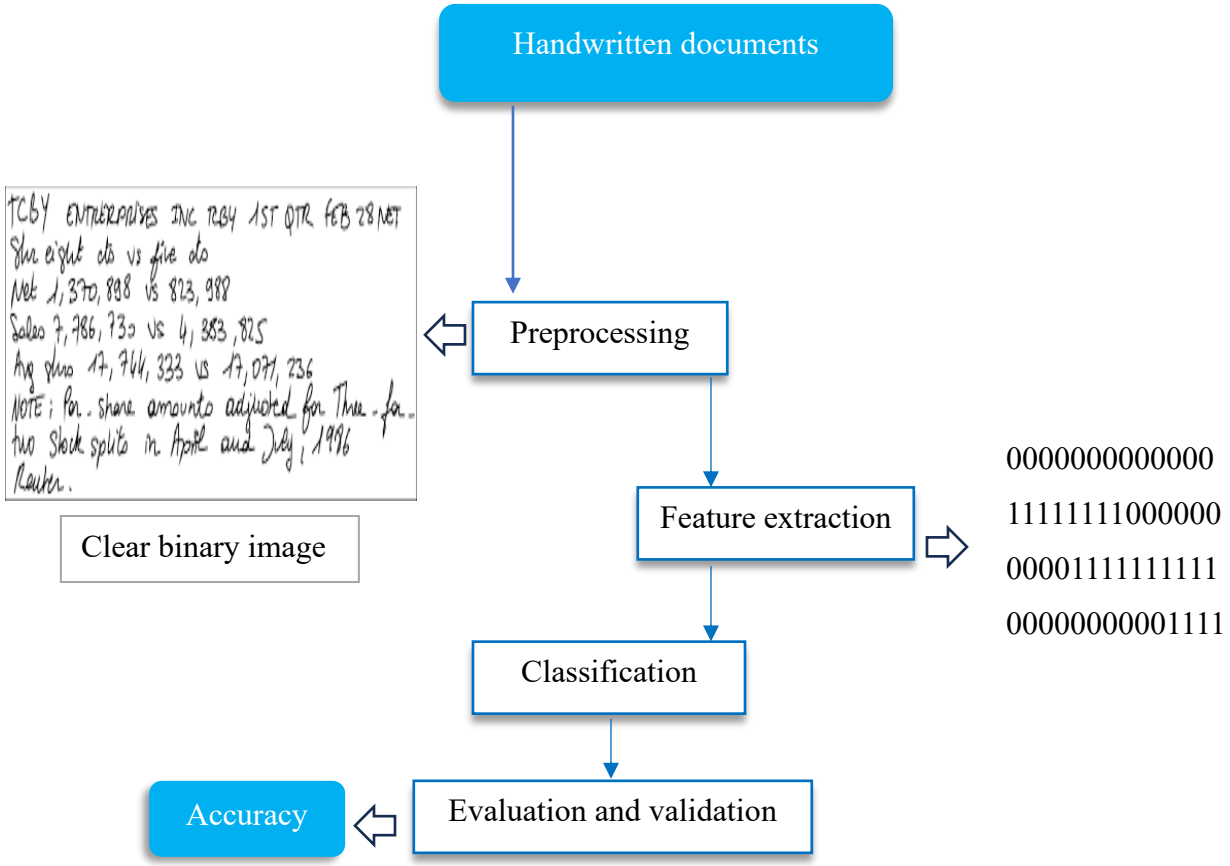


Figure 3 General architecture of writer identification system

1.6.1 Preprocessing

Preprocessing is a necessary first step in any writing identification system, particularly offline, where input data are scanned images of handwritings. Its aim is to enhance the quality of handwriting images, eliminate noise, and standardize data formats to obtain uniformity among all samples before feature extraction. A robust preprocessing pipeline to a great extent decides the reliability and accuracy of the system as a whole. Preprocessing begins typically with grayscale conversion that reduces the image to intensity levels by eliminating color information, thus simplifying downstream operations. This is followed by binarization, a process that converts the grayscale image to a binary image (black-and-white) to isolate handwriting foreground from the background. Algorithms such as Otsu's thresholding are commonly used to achieve this step effectively, even under varying illumination [11] [3] [19].

Secondly, noise reduction techniques are applied to eliminate minute spurious artifacts that may be caused by scanning errors, ink smudges, or paper texture. Morphological transformations such as erosion and dilation, or filter methods such as median or Gaussian filters, are used to eliminate the noise without compromising the integrity of the handwriting strokes [17] [20]. The preprocessing workflow, including key steps like grayscale conversion, binarization, noise reduction, normalization, and segmentation illustrated in Figure 4.

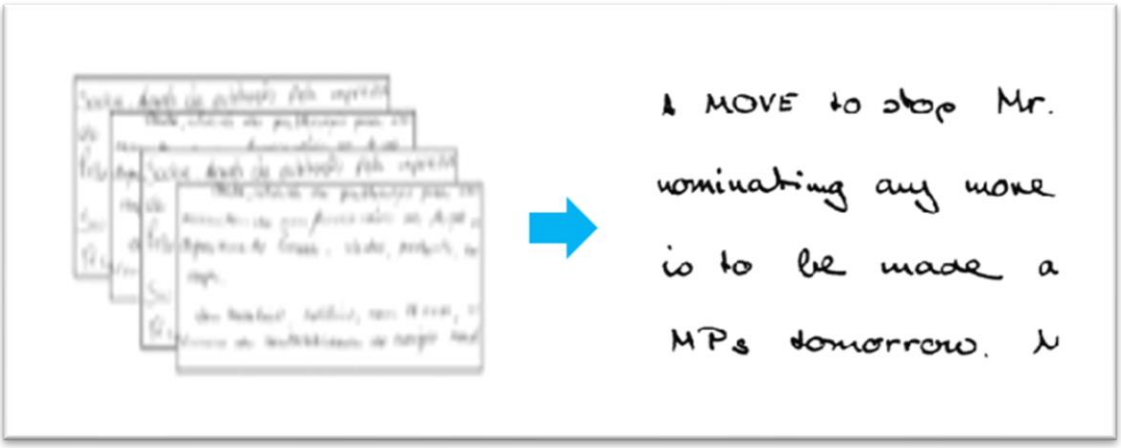


Figure 4 Image binarization and reduce noise

Normalization is another essential component, involving the resizing and alignment of handwriting samples to a uniform scale and orientation. This helps in mitigating differences caused by writing position, slant, or paper layout. Some systems may also include slant correction to adjust for the natural inclination of handwriting and baseline alignment to ensure that characters are aligned uniformly across different samples, in some cases, segmentation is performed to isolate specific elements such as lines, words, or characters, depending on whether the system uses global or local features. Although segmentation can be challenging due to the cursive nature of handwriting, it can enhance the discriminative power of the extracted features [21] [22].

Overall, preprocessing serves to reduce intra-class variability (variations within a single writer's samples) and highlight writer-specific characteristics. It creates a clean and standardized input that enhances the effectiveness of feature extraction and classification in the subsequent stages of the writer identification pipeline.

1.6.2 Feature Extraction

Feature extraction is an elementary process in a writer identification system as it transforms raw handwriting images into structured forms that are ready to be analyzed by classification algorithms. The aim of this process is to extract features that capture the uniqueness of a writer's handwriting style while mitigating the effects of text content or noise. Efficient feature extraction is most critical in identifying unique writers accurately [3] [23] [18].

The features used in writer identification can be broadly categorized into learned features and handcrafted features. Handcrafted features are designed based on domain knowledge and are often categorized into statistical, structural, and textural features. Some of the most widely used handcrafted features are textural descriptors, such as Run-Length Encoding (RL), Local Binary Patterns (LBP), and Gray-Level Co-occurrence Matrices (GLCM), which capture stroke density, directionality, and spatial relationships in handwriting. Structural features, on the other hand, capture the geometric shape of characters, curvature, and connectivity, while statistical features define global properties like slant angle, stroke width, and contour distribution [24] [12] [25]. Figure 5 illustrates the feature extraction process, highlighting how handwriting images are transformed into feature vectors suitable for analysis.

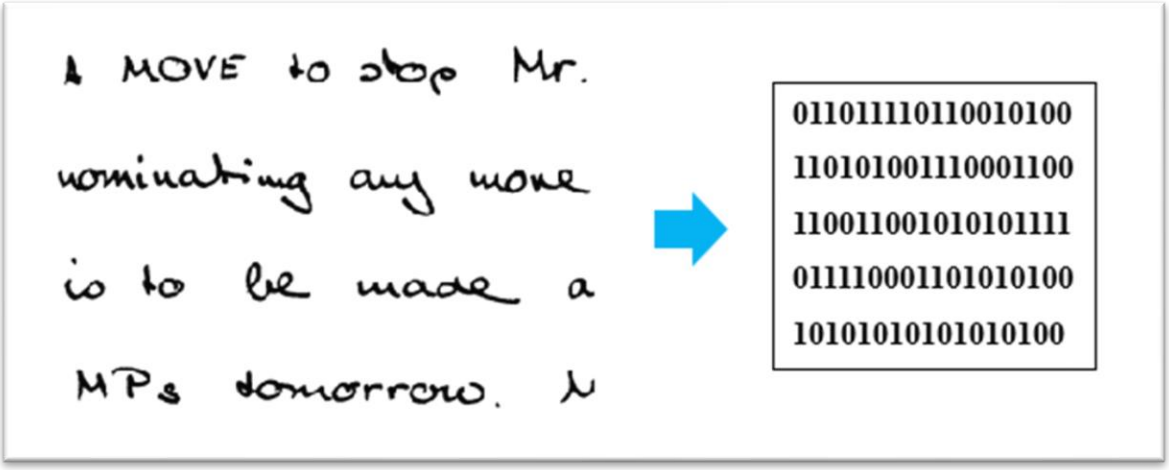


Figure 5 Feature Extraction

(Transform the image into vectors)

Conversely, learned features are automatically learned by deep learning networks, particularly Convolutional Neural Networks (CNNs). Hierarchical representations of handwriting are learned by deep learning networks in numerous layers, and they learn low-level features (edges, curves) and high-level patterns (writing style, spatial layout). Learned features are especially advantageous for large datasets where feature generalization is an important factor. Feature selection depends on application type, size and heterogeneity of the dataset, and computing resources. In offline writer identification, text variation robust features are needed in text-independent systems, and thus textural and holistic features are extremely useful [26] [27].

1.6.3 Classification

Classification is a critical stage in the writer identification process, with the responsibility of mapping a handwriting sample to the corresponding writer based on the features extracted in the previous step. It is cast as a supervised learning problem, where a labeled dataset is employed to train a classifier for learning the writer identity mapping between feature vectors. The generalization capability, strength, and accuracy of the system are all dependent upon the performance of the classifier, and diverse classification algorithms have thus been explored in the literature from traditional machine learning models to more recent deep learning models. Of these traditional classifiers, Support Vector Machines (SVM) are widely applied due to their advantages in high-dimensional feature spaces as well as their ability to handle nonlinear decision boundaries.

k-Nearest Neighbors (k-NN) is yet another well-known method, which classifies a sample by the majority class among its nearest neighbors in feature space. Though extremely simple, k-NN is capable of yielding strong performance when feature representation is discriminative. Decision Trees, Random Forests, and ensemble learning algorithms are some of the other conventional classifiers [28] [24] [29] [30]. Figure 6 presents a visual diagram of the classification process, illustrating how extracted feature vectors are passed through a trained model to produce writer identity predictions.

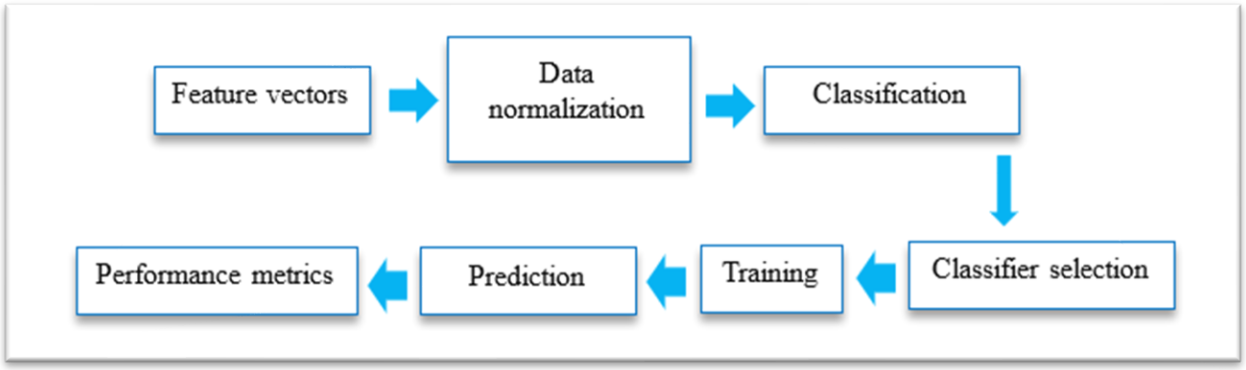


Figure 6 Classification diagram

Deep learning classifiers, particularly those with fully connected layers as the output layers of Convolutional Neural Networks (CNNs), have worked extremely well for writer identification during the past few years. The classifiers leverage end-to-end training and can learn high-level, nonlinear representations of script style. However, they generally require large annotated datasets and large amounts of computational power [27].

The choice of classifier must consider dataset size, type of features, computational cost, and the nature of the writing samples. Hybrid approaches, combining multiple classifiers or combining hand-engineered and learned features, are also employed by some systems for better performance and robustness. Classification is the decision-making component of the writer identification system and maps feature representations to writer labels. Its effectiveness is not only dependent on the algorithm used but also on the quality of the features and model training and optimization techniques adhered to [31] [32].

1.6.4 Evaluation and Validation

Validation and evaluation are important steps in creating a writer identification system since they measure the system's accuracy, robustness, and generalizability. Without evaluating the system, one cannot determine its true effectiveness or even compare it to existing approaches. The step involves the selection of appropriate performance metrics and validation strategies for measuring the performance of the system in writer identification, particularly for unseen data. The most common metric used in writer identification is classification accuracy, which is the ratio of handwriting samples identified correctly. For multi-class problems or imbalanced datasets, metrics based on accuracy can be deceiving, however. Precision, recall, F1-score, and confusion matrix give an additional view of how well the classifier performs on each writer class. These measurements assist in determining if the system is inclined toward particular writers or plagued by repeated misclassifications [33] [34] [24].

To ensure generalization ability of the model, validation techniques are employed. Two of the most prevalent ones are k-fold cross-validation and Leave-One-Out Cross-Validation (LOOCV). In k-fold cross-validation, data is divided into k equal subsets; the model is trained on $k-1$ subsets and tested against the remaining one. This process is repeated k times, and then an average is taken. LOOCV is a special case when k is made equal to the number of samples to have a better evaluation, best for small datasets. In writer identification tasks, it is also critical that training and test samples correspond to varying handwriting examples of the same author to prevent leakage of data. Further, text-independent systems should also be tested on different textual content samples to correctly reflect real usage contexts [35] [36] [17].

The test phase can also involve comparative analysis against baseline process or existing systems to verify improvement in performance. Runtime behavior, utilization of resources, and scalability problems can also be part of the verification process, particularly in installing the system for real or forensic deployment. Therefore, rigorous testing and verification are crucial in building a reliable writer identification system. They are evidence of the model's success, inform improvement, and prove that the system is adequate in demands of accuracy and reliability for practical use.

1.7 Handwriting Image Datasets

The success of a writer identification system highly depends on the quality, diversity, and realism of handwriting datasets for training and test evaluation. These datasets provide the empirical foundation for algorithm designing, testing, and validation across varying conditions, e.g., text variability, writer variability, and acquisition noise. Several publicly available datasets have been created to support research in offline handwriting analysis, each having its own characteristics and challenges [3] [37].

1.7.1 Description of Dataset

A dataset is a structured collection of data that is utilized for testing, training, and analysis across numerous application and research fields. In writer identification, a dataset is typically a collection of handwriting samples collected from various individuals under real or simulated environments. A dataset sample is usually labeled with metadata such as writer identity, writing instrument, language, or style. Datasets serve as the foundation upon which to construct, test, and validate algorithms, and aid researchers in the quantification of system performance and comparison of methods fairly. They are required to offer reproducible experiments and advance the field through standardized benchmarks [38].

1.7.2 BFL

We tested our system in BFL dataset, which contains 315 authors, three samples from each author, totaling 945 images. The reason to use such database is the growing interest of the Brazilian forensic experts and the Brazilian Federal Police in writer verification/identification,the samples were taken from three sessions of undergraduate students in one month,the authors could use his/her own pen, i.e., a few different pens were used. The text is full (131 characters) and includes all the characters (letters and numbers) as well as specific combinations of interest. It's therefore also suitable for text-dependent writer identification[39].The Table 01 presents the characteristics of BFL Dataset and it’s components.

Table 1 Description of BFL Dataset

Property	Description
Name	BFL (Brazilian Forensic Letter)
Purpose	Writer identification
Number of Writers	315 writers
Samples per Writer	3 handwritten images per writer
Total Number of Images	945 images (315 writers × 3 images)
Language / Script	Latin (French handwritten text)

That's a sample from BFL (Figure 7)

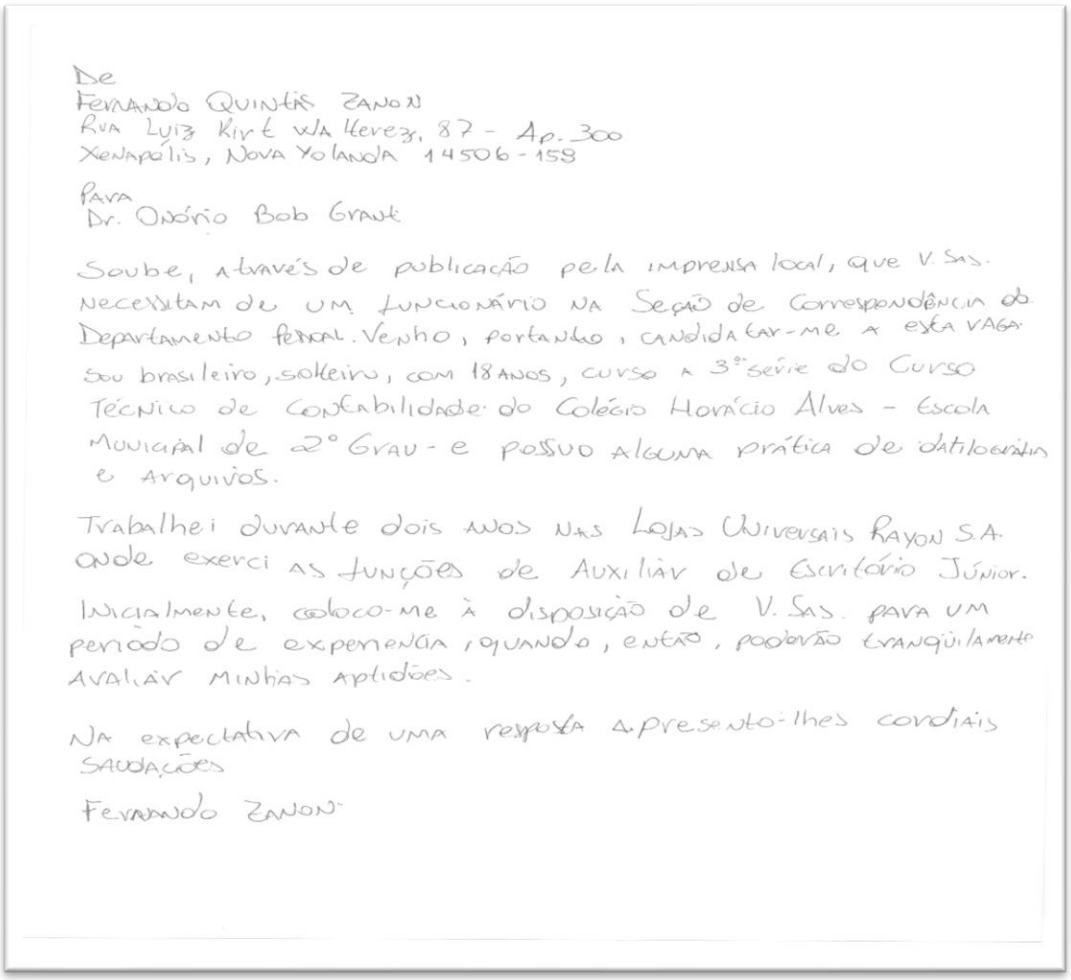


Figure 7 Example from BFL Dataset [39]

1.7.3 CVL

The CVL Database is a public writer retrieval, writer identification and word spotting database. The database contains 7 handwritten texts (1 German and 6 Englisch Texts). There were 310 writers which participated in the dataset. 27 of them wrote 7 texts and 283 writers were compelled to write 5 texts. To each text a rgb color image (300 dpi) with the handwritten text and the printed sample text, and a cropped one (handwritten text only) is delivered. An individual id identifies the author, whereas the Bounding Boxes for each single word are stored in an XML file, the CVL-database comprises images with cursively handwritten german and english texts which have been chosen from literary texts. Every page has a writer id and the dashed text number at the top right corner, succeeded by the printed sample text. The text is placed between two horizontal separatores. Under the printed text subjects have been asked to write the text on a ruled undersheet so that text lines do not curl. The layout simulates IAM database style. The database was updated on 12/09/2013 because one writer ID (265/266) was wrong [40].

1.7.4 IAM

The IAM Handwriting Database is the most utilized benchmark in offline handwriting analysis and writer identification research. Published by Marti and Bunke in 2002, IAM dataset provides a rich collection of handwritten English text samples contributed by a wide population of authors, making it very suitable for uses such as writer identification, word recognition, and handwriting segmentation. The IAM dataset comprises forms scanned at 300 dpi and written by 657 different writers. Each writer was asked to recreate fixed English texts drawn from the Lancaster-Oslo/Bergen (LOB) corpus, providing a linguistically diverse and balanced dataset. The text content is separated into paragraphs, lines, words, and characters, along with ground-truth annotations for writer ID, text transcription, and segmentation boundaries [17] [41].

One of the defining strengths of the IAM dataset is its detailed annotation and hierarchical structure, which enables experiments at multiple levels: paragraph-based, line-based, word-based, or even character-based writer identification. The dataset is stored in grayscale TIFF image format, preserving fine details in handwriting strokes, which is essential for extracting textural and shape-based features [17].

1.7.5 KHAT

The KHATT dataset (KACST Handwritten Arabic Text Database) is an open large-scale dataset and is intended to boost offline Arabic handwriting recognition, writer identification, and other related research. The database was created by the King Abdulaziz City for Science and Technology (KACST) and has 1,000 handwritten pages of 1,000 writers with a balanced mixture of males and females. Four pages of handwritten Arabic per writer were provided with two pages of Naskh and two pages of Ruq'ah script, totaling 4,000 scanned pages. Handwriting material is normalized using predetermined paragraphs of Arabic to possess similar content in the samples. KHATT is very useful for text line segmentation tasks, script style detection, gender identification, and machine learning model training for handwriting-based biometrics. It remains a reference dataset for research in Arabic handwriting [42], Figure 8 shows a sample from the dataset.

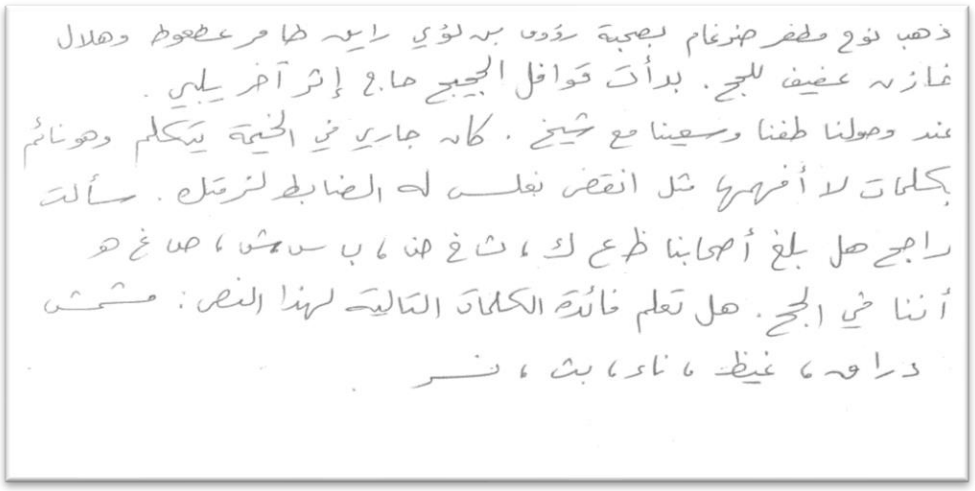


Figure 8 Example from KHATT Dataset

1.8 Image Characteristics and Formats

The effectiveness of a writer identification system is closely tied to the quality and characteristics of the handwriting images it processes. Most handwriting datasets are composed of high-resolution grayscale or binary images, typically scanned at 300 dots per inch (dpi) or higher. Grayscale images capture a wide range of intensity values, preserving subtle variations in pen pressure and stroke continuity. Binary images, produced through thresholding or binarization techniques, simplify processing but require careful handling to avoid the loss of important handwriting details. Images are commonly stored in formats such as TIFF, PNG, JPEG, BMP ...etc, these images may include entire pages or pre-segmented units such as lines, words, or free-flowing

text, depending on the dataset. In well-structured datasets, images are accompanied by metadata that includes writer identifiers, text content, and layout information, enabling controlled experimentation and training of supervised learning models.

1.9 Challenges in Realistic Handwritten Data

Writer recognition systems face numerous challenges when they have to process realistic handwritten data, as opposed to controlled or artificially chosen datasets. In real-life situations, handwriting is exposed to a wide variability influenced by physical, psychological, and environmental factors. This variability poses significant challenges to preprocessing, feature extraction, and classification stages of a writer identification pipeline, one of the significant challenges being intra-writer variability, where a person's handwriting can differ depending on mood, writing speed, fatigue, writing instrument, or writing surface. These variations can obscure consistent writer-dependent features and reduce classification accuracy. Conversely, inter-writer similarity where authors share similar handwriting styles a cause of false identification, particularly in large population databases [43] [44] [25].

Other difficulties are added noise and artifacts introduced in image acquisition, e.g., low-res scanning, inhomogeneous lighting, background textures (e.g., ruled paper), or ink smears. Such artifacts can clash with edge detection, thresholding, and other preprocessing operations, especially in offline scenarios. Besides, the presence of non-standard script patterns, skewed formats, and multi-lingual or multi-script text further forces the system to struggle in identifying the right writer-dependent features for separating and analyzing [45] [46] [18] .

1.10 Conclusion

In this chapter, we have formulated the fundamental concepts and problems of offline writer identification systems. We began with classifying handwriting recognition modes online and offline and defining the difference between writer identification and verification, highlighting their respective applications in application fields such as forensic analysis, biometric authentication, and document processing, and then elaborated on the general architecture of a writer identification system, defining the basic steps that go into preprocessing, feature extraction, classification, and evaluation. All the steps were shown to be crucial in transforming raw handwriting data into structured representations for application with machine learning models, and we elucidated the different types of handwriting image datasets in system construction, and we

surveyed popular databases such as IAM, CVL,BFL and KHATT, and how dataset quality, image characteristics, and real-world factors such as intra-writer variability, inter-writer similarity, and image acquisition artifacts can have profoundly significant impacts on system performance.

Chapter 02

State of The Art

2.1 Introduction

Writer identification offline is the most prominent area of research in handwriting analysis in recent years, prompted by advances in pattern recognition and machine learning. Offline writer identification is the process of identifying the writer of handwritten text using automated systems and is used in forensic science, biometric authentication, and document verification. This chapter presents a well-ordered summary of the major developments in the field, offering background information on how the methods developed and highlighting the importance of handwriting identification [4] [6].

The chapter overviews some of the methods that have been suggested over the years, from early heuristic techniques to state-of-the-art feature extraction and classification methods. The chapter emphasizes the significance of texture-based descriptors, in particular, run-length features, that are able to quite well represent stroke patterns and visual handwriting characteristics. The chapter also describes widely used classification methods such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN), which have worked exceptionally well when trained on well-designed features, also covered is an explanation of standard handwriting datasets that serve as a point of comparison for verifying identification methods. It contrasts their protocols and features, and how challenging they are. Through tracing the chronological sequence of study activity and establishing strengths and weaknesses of existing methods.

2.2 Related works

The text-independent writer identification community has been approached by a plethora of researchers with varying methods, indicative of both the richness and intensity of this community. Compared to fixed content-based comparison-dependent text systems, text-independent systems must operate with no prior knowledge of written documents, and as a result, the task is much more challenging. Researchers have explored an extensive range of feature extraction methods varying from structural features and statistical descriptors to more contemporary deep learning-based features. Traditional methods commonly utilize hand-designed features like texture descriptors, stroke width estimates, and edge direction histograms to characterize writing styles. Deep learning methods, especially Convolutional Neural Networks (CNNs), have been employed to learn discriminative features automatically from raw image data instead. Moreover, the performance of other classifiers such as Support Vector Machines (SVMs), k-Nearest Neighbors (KNN), and

ensemble methods have been evaluated. Despite the progress, writer identification from text-independent handwriting is a challenging task due to factors such as intra-writer variability, inter-writer similarity, and noise in the handwriting data. Thus, constructing robust and generalizable systems is an area of research interest [9] [32] .

C. Djeddi and L. Souici-Meslati (2008) [17] suggested a handwriting recognition approach based on local feature extraction, where the handwriting image is segmented into small imagettes. The segments were recognized with sequential classification and pattern matching techniques. The approach proposed in this paper is defined by the use of local features, so-called "thumbnails", which are small square blocks coming from the handwriting word image. The image is divided first into disjoint writing blocks by vertical and horizontal projections analysis, Figure 09 shows the proposed method for cutting of writing.

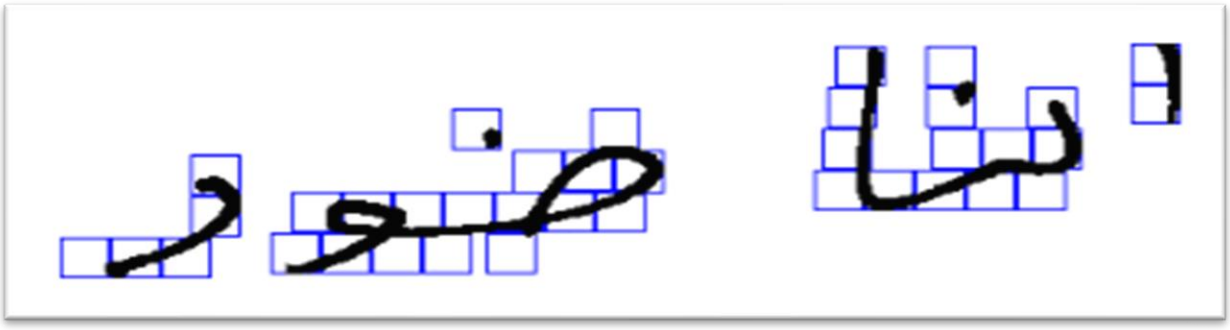


Figure 9 Proposed method for the cutting of writing. [17]

Then each block is subdivided into thumbnails of a given size (e.g. 19×19 pixels), like we see in fig 1 an example of the proposed method for the division of handwriting, and those containing little information are removed. The other thumbnails, which maintain the invariant shapes characteristic of each scripter's hand, are used for learning and classification. A sequential classification algorithm groups similar thumbnails into classes to characterize each individual's handwriting style. The experiments were carried out on two datasets: one with samples from 33 Algerian writers (province names) and the other from the IFN/ENIT dataset (Tunisian place names). The approach was 90% accurate on the Algerian dataset and 93.93% accurate on the IFN/ENIT dataset, proving to be effective for Arabic writer identification.

Then In 2013, Golnaz Ghiasi and Reza Safabakhsh [41], presented an offline text-independent writer identification system based on a codebook-based method. They followed an

approach relying on structural handwriting features such as curves and angles, with K-means clustering for codebook generation and SVM/KNN classification.

The method uses the histogram of shape occurrence in a codebook to create a feature vector for each manuscript. For cursive scripts, a wide variety of different shapes exist in the related components obtained from the handwriting. Small fragments of related components are used to avoid complex patterns. Two efficient methods for extracting codes from contours are presented. One method uses the actual pixel coordinates of the contour fragments, while the other uses a piecewise linear approximation using angles and segment lengths. To evaluate the methods, scripter identification is performed on two English and three Farsi handwriting databases. Both methods showed promising performance, with the latter outperforming the former. The system was tested on IAM (English) and AUT-FH databases and achieved a recognition rate of 96.9% with an error rate of less than 5% in most of the cases, showing that structural features significantly help in writer recognition [41].

That's a block diagram of the writer identification system shown in Figure 10:

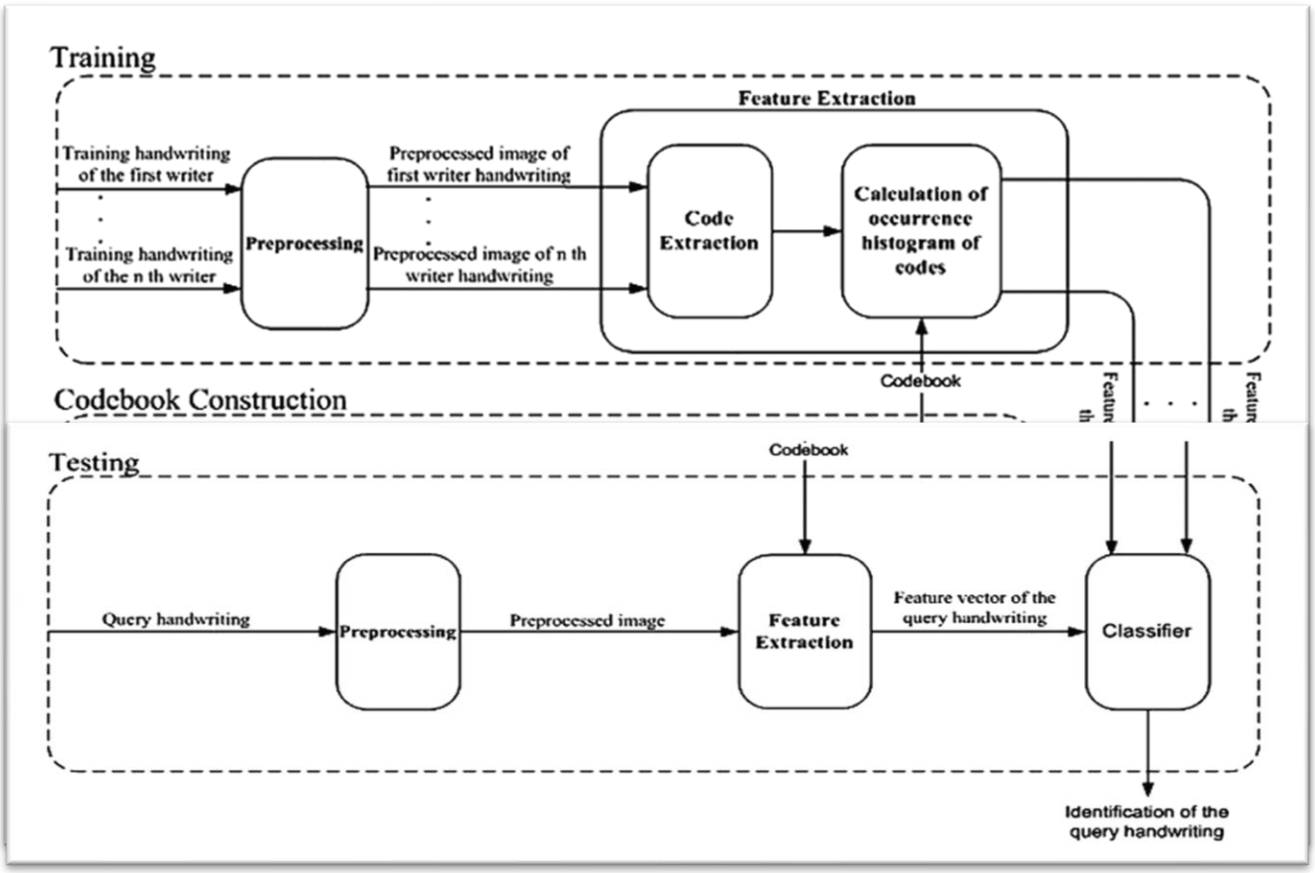


Figure 10 Block diagram of the writer identification system. [41]

Also there is Xiangqian Wu et al. (2014) [20], presented an offline writer identification method based on SIFT wherein they employed an isotropic LoG filter for word separation and a Kohonen SOM for codebook generation. Their technique was designed to extract SIFT descriptors, denoted as SD, as well as a Scale and Orientation Histogram, called SOH, for classification. The approach proposed in this article is distinguished by the use of local features, called “thumbnails”, which are small square blocks extracted from the handwritten word image. The image is first segmented into disjoint writing blocks by analyzing vertical and horizontal projections. Each block is then cut into thumbnails of fixed size (e.g. 19×19 pixels), and those containing little information are eliminated. The remaining thumbnails, which capture the invariant shapes specific to each scripter's hand, are used for learning and classification. A sequential classification algorithm groups similar thumbnails together to characterize each individual's handwriting style, Figure 11 demonstrates the framework of the method proposed in this study.

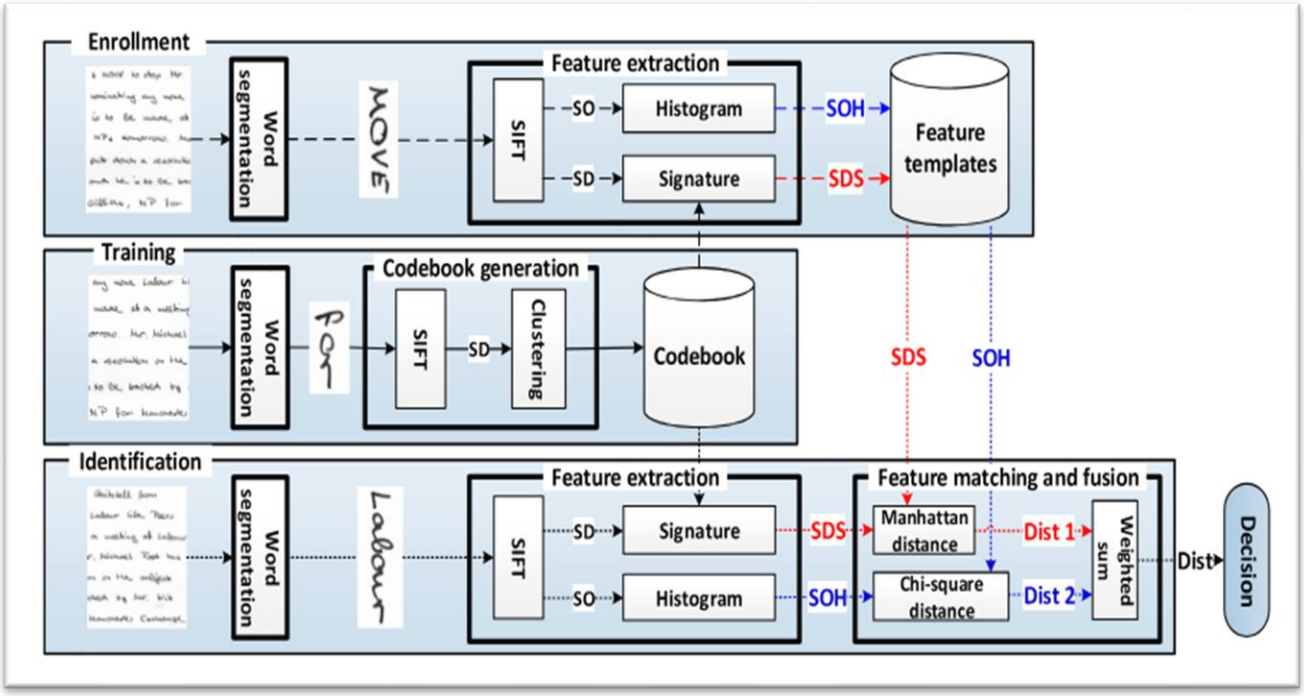


Figure 11 Offline writer identification method based on SIFT [20]

The method tested on various datasets, namely IAM, Firemaker, HIT-MW, ICDAR2011, ICFHR2012, the method obtained a relatively high accuracy, IAM acquiring 98.5% (top-1), 99.1% (top-5), and 99.5% (top-10), and hence proved the resilience of SIFT-based features.

Also Chawki Djeddi's (2014)[18], thesis entitled “Contribution to the analysis and characterization of handwriting” makes significant contributions in three main areas of automatic handwriting analysis. Firstly, it proposes new methods for text-independent recognition of the writer, without imposing a fixed textual content. Secondly, it develops robust approaches for the analysis of multi-script documents, notably Latin and Greek, by demonstrating the independence of certain features from the script used. Thirdly, it presents efficient techniques for the automatic determination of an individual's gender from his or her handwriting. The thesis is based on the extraction of innovative features based on the distribution of segment lengths (run-length) and other features (orientation, curvature, fractal and textural features), tested on several international

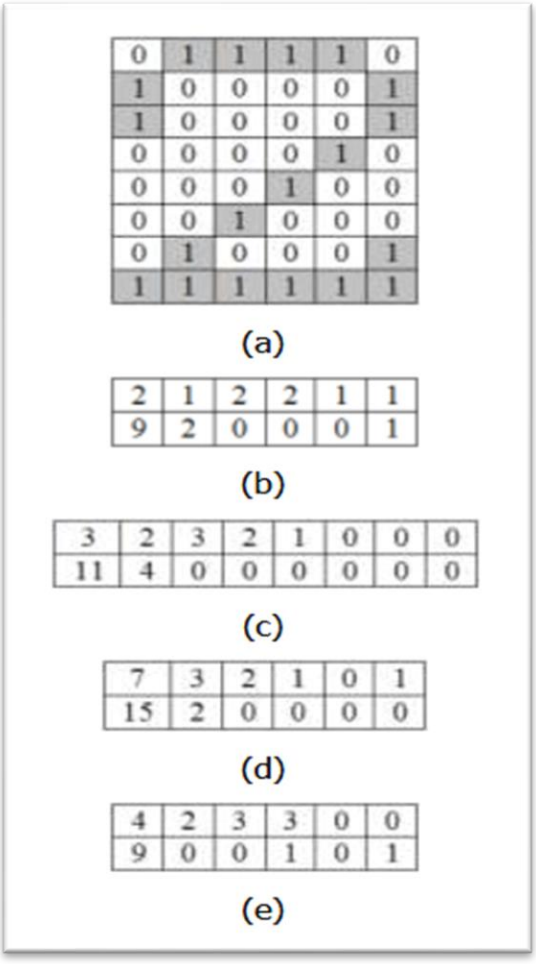


Figure 12 Calculating the run-length matrix (a) An 8×6 image .

databases, Figure13 presents an example of a 8×6 pixels image we applied run-length method on it.

Djeddi's work has been validated by significant results in several international competitions and has contributed to the advancement of research in the field of handwriting analysis. It uses handwriting analysis by Run-Length and Local Binary Patterns (LBP) along with SVM/K-NN classifiers.

We got this results for the different datasets: IFN/ENIT, KHATT, CVL, IAM, CEDAR, BFL, MSHD, ICDAR2011–2013 are presented with high results: 88.07% (top-1) for IFN/ENIT, 70.60% (top-1) for KHATT, and 96.00% (top-1) for CVL. For ICDAR2011, the perfect recognition rate is reported, confirming that Run-Length and LBP can be efficiently applied to characterize handwriting.

Then there is also the work Abdi and Khemakhem (2015) [47], proposes a novel approach to offline, text-independent identification and verification of handwritten Arabic scriptwriters, based on a computational generative model called the beta-elliptic model. Unlike conventional methods which extract real graphemes from large training corpora via segmentation and clustering techniques, the authors here synthesize a set of artificial graphemes directly from the mathematical model, thus generating a codebook entirely independent of the training data. This independence overcomes the limitations of collecting and annotating large databases, while ensuring better generalization. For each handwritten document, features are extracted by template matching between handwriting fragments and codebook graphemes, enabling the construction of robust and discriminating feature vectors.

To optimize the size and relevance of the descriptors, various feature selection methods are applied, such as FDR, cross-correlation and random sub-sampling. The experimental evaluation, carried out on the IFN/ENIT database of 411 scriptwriters, demonstrates the high efficiency of the method: it achieves a top-1 identification rate of 90.02%, a top-5 rate of 96.35% and an equal error rate (EER) of just 2.1% for verification. These performances outperform most existing approaches, whether based on textural, structural or statistical features, and confirm the ability of the synthetic codebook to faithfully represent the diversity of Arabic handwriting styles. The originality of this approach lies in the exploitation of a biomechanical model of motion generation to produce invariant and universal descriptors, making the method particularly suitable for biometric

applications, document security and handwriting authentication. Figure 12 illustrates the framework of the proposed approach.

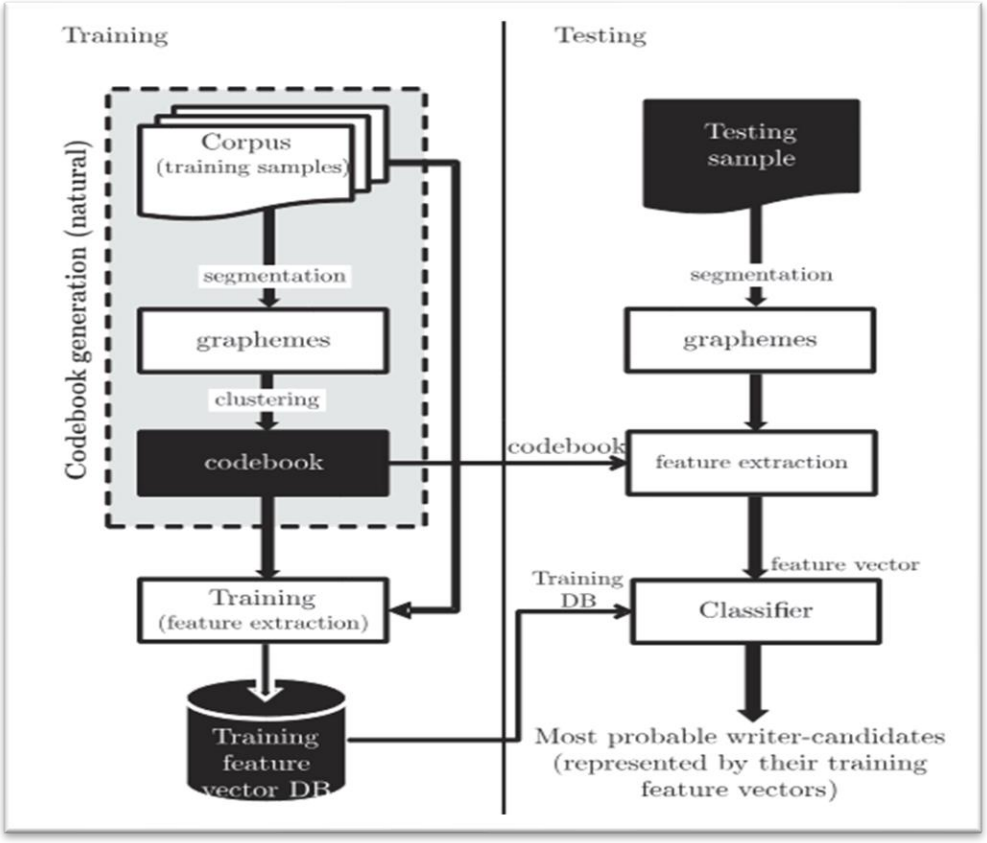


Figure 13 Natural corpus-driven codebook generation and writer identification [47]

Then Bennour et al. (2019) [48], suggested a new method for recognizing writers from images of handwriting, based on the construction of an "implicit codebook of shapes". Unlike traditional approaches that segment text into semantic units (such as characters), the method presented here focuses on extracting small fragments around points of interest detected in the handwriting, using the Harris algorithm. These fragments, or "patches", are then grouped by agglomerative hierarchical clustering to form a visual codebook that captures recurring patterns in writing style, without depending on textual content, Figure 3 illustrates the steps of the proposed method.

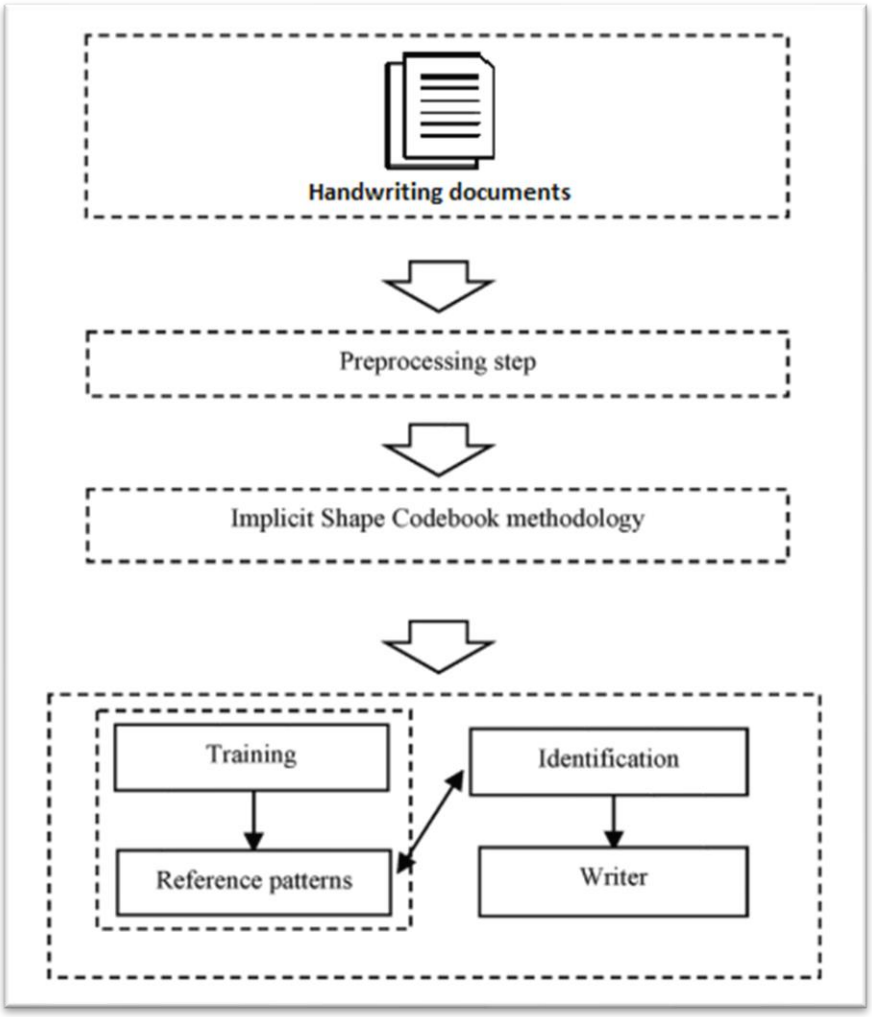


Figure 14 Steps of the offline writer identification base on implicit codebook of shapes [48]

Each writer is then characterized by the probability distribution of production of these codebook patterns in his or her writing samples. For identification and verification, the authors use SVM and k-nearest neighbor classifiers, evaluating their method in text-dependent and text-independent modes on two public reference databases: BFL and CVL. Experimental results show that this approach achieves very promising identification and verification rates, even in complex scenarios, confirming the method's suitability for applications in biometrics, forensics and handwritten document management. The study also highlights the value of IT solutions in assisting forensic scientists, by making it possible to quickly narrow down the list of potential suspects from a handwritten sample. Finally, the authors point out that their approach, based on local patterns

and computer vision techniques, is robust, flexible and suitable for large databases, paving the way for practical applications in forensic analysis and document retrieval.

Experimented on BFL (315 writers) and CVL (311 writers), the system achieved 98% (top-1) accuracy, justifying its robustness with varied handwriting conditions.

There is Faycel Abbas et al. (2020) [30], suggested a texture-based approach with the combination of LBP and oriented Basic Image Features (oBIFs). this new method for the automatic identification of scriptwriters from handwriting images is applicable to both mono-script (a single language) and multi-script (several languages or alphabets). The authors are based on the hypothesis that each individual has his or her own writing characteristics, which remain partly constant even when changing scripts (for example, writing in Arabic and then in Latin). To exploit this specificity, they propose an original textural descriptor based on the cross-use of Local Binary Patterns (LBPs) in different configurations, combined with column histograms derived from Oriented Basic Image Features (oBIFs). These descriptors capture the fine texture of the handwriting, considered as a unique signature of each writer, independently of the written text.

The method involves extracting column histograms of LBPs and oBIFs for each handwritten image, which are then concatenated to form a feature vector representative of handwriting style. A Support Vector Machine (SVM) classifier is used for automatic writer identification. The authors evaluate their approach on five different databases, covering mono-script and multi-script scenarios, to test the robustness of their method in the face of linguistic and graphical diversity.

The experimental results show that the combination of the proposed text descriptors achieves very high identification rates, outperforming state-of-the-art methods, both when the script is in a single language and when it is in multiple scripts, the study also confirms the hypothesis that certain aspects of an individual's writing style persist across different alphabets, paving the way for applications in forensics, behavioral biometrics and multilingual document management. Finally, the authors emphasize the importance of having explicable and robust descriptors, which facilitate the acceptance of these systems by human experts, particularly in the forensic field.Tested on BFL (Portuguese) and KHATT (Arabic), their system achieved 98.63% (top-1) on BFL and 77.10% (top-1) on KHATT, with good cross-script performance.

There is also Djeddi et al. (2021) [44], studied the impact of codebook patterns on the performance of handwriting identification and verification systems. Codebook-based approaches consist in dividing handwriting samples into small fragments (or graphemes), grouping these fragments into recurring patterns (codebook), and then representing each scripter by the distribution of these patterns in his or her writings. Traditionally, the codebook is generated from a subset of the database, while the rest is used for evaluation. The central hypothesis of the article is that the codebook merely provides a representational space for comparing different writing styles, and that the precise choice of patterns within it has no significant influence on identification or verification performance. To validate this hypothesis, the authors generated several codebooks from handwritten scripts in Greek, Arabic and Chinese, as well as from handwritten musical scores, printed texts and synthetic data. Extensive experiments were carried out on three reference handwriting recognition databases (CVL, BFL and IAM), covering different scripts and contexts. The results show that, in general, the nature of the codebook patterns (their linguistic origin or even their handwritten or non-handwritten nature) has no significant impact on the system's ability to recognize or verify a scripter. Performance remains stable, whatever the type of codebook used, as long as the method of fragmentation and representation remains consistent.

This study therefore challenges the conventional wisdom that it is essential to adapt the codebook to the script or language of the writing to be analyzed. The authors conclude that the codebook serves primarily as a statistical support for comparing distributions of writing patterns, and that most of the performance comes from the way the fragments are extracted and the statistical representation employed, rather than from the exact content of the codebook. These findings pave the way for more universal scripter identification systems, less dependent on language or script type, and simpler to implement.

Also Deepa Bendigeri et al. (2022) [27], presents an innovative approach to writer identification from handwritten text images, relying on a hybrid model called the Convolutional Recurrent Neural Network (CRNN). This architecture combines the strengths of Convolutional Neural Networks (CNNs), which automatically extract local and spatial features from handwritten images, and Recurrent Neural Networks (RNNs), which are capable of capturing sequential and contextual relationships between writing fragments. Unlike traditional methods that require manual feature extraction or prior segmentation, the CRNN processes handwriting images end-to-

end, regardless of their size or orientation, making it particularly flexible and robust to the natural variability of writing styles. The study highlights the importance of "offline" identification-based solely on scanned images-for applications in forensics, security, and document management, where rapid and reliable author authentication is crucial.

The authors show that their model achieves 96% accuracy on a set of 690 writers, surpassing the performance achieved by CNN or RNN architectures alone. The article also includes a comparative analysis with other recent approaches and demonstrates that combining convolutional and recurrent layers in a CRNN architecture offers a powerful, automated, and generalizable solution for handwritten text author identification, paving the way for practical applications in forensic analysis and behavioral biometrics..Achieved 96% accuracy with CRNN on the IAM dataset, with the individual CNN (90%) and RNN (93%) models being inferior.

Finally, Ahmed et al. (2023) [9], suggested a codebook of structural features with CPCA and CON3 descriptors. The proposed method extracts the contours of connected components from handwriting images and divides these contours into fixed-length segments. Two simple yet effective structural features are then computed for each segment: the contour point curve angle (CPCA) and the contour point concavity/convexity (CON3). The codebook structural features of this study is shown in the figure below.

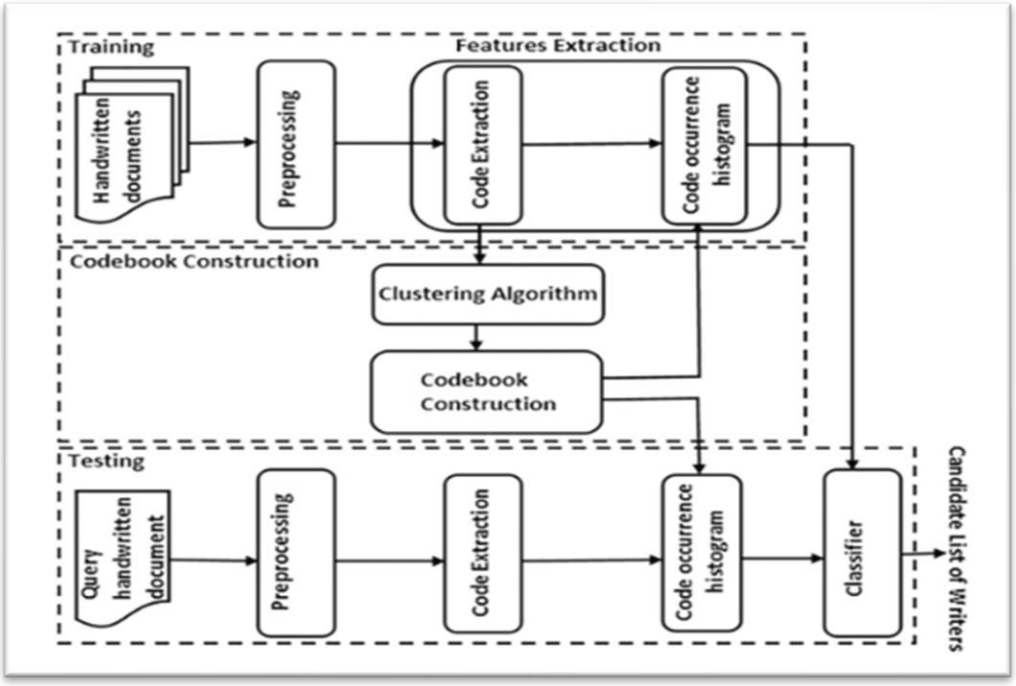


Figure 15 Overview of the writer identification using implicit codebook of shapes [9]

These features are clustered using k-means to create a codebook, and each document is represented by a histogram of feature occurrences within this codebook-a "bag of features" approach. For classification, both nearest neighbor and support vector machine (SVM) methods are employed. The system is evaluated on two large, publicly available datasets: the English IAM and the Arabic KHATT databases. Experimental results show that the proposed method outperforms state-of-the-art approaches on the IAM dataset and achieves competitive results on the KHATT dataset. The study demonstrates the effectiveness of analyzing small handwriting fragments using structural features, confirming that such local analysis can be more discriminative than global texture-based approaches. The authors also provide a comprehensive review of related work and highlight the challenges of writer identification, such as intra-writer variability and the diversity of writing styles, especially in languages like Arabic. Overall, the proposed system offers a robust and efficient solution for writer identification across different scripts and languages, with promising applications in forensics, document authentication, and biometric systems. Their method achieved 98.2% (IAM) and 89.5% (KHATT) accuracy, once again confirming the effectiveness of structural analysis for writer identification.

Collectively, these works present advances in feature extraction, machine learning, and hybrid methods, with continued evolution in text-independent writer identification.

2.3 State of the art Summary

For illustration purposes, the following Table briefly summarizes all the works mentioned in this chapter:

Table 2 Summary of the State of the Art.

Article Title	Author	Year	Method	Classifier	Dataset	Results
Une approche locale en mode indépendant du texte pour l'identification de scripteurs : Application à l'écriture arabe	C. Djeddi, L. Souici-Meslati	2008	- segmentation d'images en « imagettes » - extraction de caractéristiques invariantes	/	- Noms des wilayas algériennes - IFN/ENIT des noms de villes et villages tunisiens.	90% : provinces algériennes. 93,93% : IFN/ENIT des noms de villes et villages tunisiens.
Offline text-independent writer identification using codebook and efficient code extraction methods	Golnaz Ghiasi, Reza Safabakhsh	2013	Codebook method with two code extraction methods (curve fragment and line fragment)	/	English (IAM, Firemaker) and Farsi (Shahabinejad , AUT-FH, FHT) databases	Top1: 91.8% (English), 96.9% (Farsi)

Contribution to the analysis and characterization of handwriting	Chawki djeddi	2014	Run-length LBP	SVM K-NN	-IFN/ENIT -QUWI -KHATT - Other datasets: IAM, CEDAR, CVL, -BFL - MSHD(LAMIS) -ICFHR2012 -ICDAR2011 -ICDAR2013	-IFN-ENIT: Longueurs de segments noirs et blancs: 98.54% on top 10 -KHATT : 84.8% on top 10 -CVL: 98.10% on top 10 -ICDAR2011: 98.6/100/100/100 -ICFHR2012: 89.8/94.3/97.8/98.8 -ICDAR2013: 90.1/93.4/97/97.9
Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform	Xiangqian Wu, Youbao Tang, Wei Bu	2014	SIFT-based method: codebook construction (hierarchical Kohonen SOM),	SIFT method SOH+SDS	IAM Firemaker ImUnipen HIT-MW ICDAR2011 ICFHR2012	(IAM) 99.5% on top 10 (Firemaker) 98.8% top10 (HIT-MW) 99.2% top 10 (ICFHR2012) 99.8% top 10

A model-based approach to offline text-independent Arabic writer identification and verification	Mohamed Nidhal Abdi, Maher Khemakhem	2015	Beta-elliptic model for synthetic grapheme codebook generation	/	IFN/ENIT Arabic database (411 writers)	Top1: 90.02%, Top5: 96.35%, EER: 2.1%
Handwriting Based Writer Recognition Using Implicit Shape Codebook	Akram Bennour, Chawki Djeddi, Abdeljalil Gattal, Imran Siddiqi, Tahar Mekhaznia	2019	Implicit shape codebook	SVM	BFL CVL	BFL : 20 writers 1811 dimension 98.00% on top 1 CVL : 12 writers 954 dimension 94.00% on top 1
Influence of codebook patterns on writer recognition: An experimental study	Chawki Djeddi, Imran Siddiqi, Abdeljalil Gattal, Somaya Al-Maadeed	2020	Codebook-based writer characterization	SVM K-NN	CVL BFL IAM	KNN/SVM/ERR CVL : 93.67%/95.67%/2.50% BFL : 90.00%/89.00%/7.83% IAM : 88.67%/90.33%/3.00%

Texture feature column scheme for single- and multi-script writer identification	Faycel Abbas, Abdeljalil Gattal, Chawki Djeddi, Imran Siddiqi, Ameur Bensefia, Kamel Saoudi	2021	LBP/BIF	SVM	BFL KHATT	98.63% on BFL 77.10% on KHATT
Offline Writer Identification using Convolutional Recurrent Neural Network	Deepa Bendigeri, Gopal A. Bidkar, Jagadeesh D.Pujari	2022	/	CNN +RNN (CRNN)	IAM	CNN : 90% RNN : 93% CRNN : 96%
Offline text-independent writer identification using a codebook with structural features	Bashar Q.Ahmed, Yaser F. Hassan, Ashraf S. Elsayed	2023	Codebook with structural features (CPCA and CON^3)	SVM K-NN	Arabic (KHATT) and English (IAM) datasets	Top1: 98.2% (IAM), 89.5% (KHATT)

The following Table presents the works employed SVM and K-NN classifiers:

Table 3 SVM and K-NN works

Classifier: SVM / K-NN	**Dataset**	**Results**
Influence of codebook patterns on writer recognition: An experimental study	CVL BFL IAM	CVL : 93.67%/95.67%/92.50% BFL : 90.00%/89.00%/87.83% IAM : 88.67%/90.33%/93.00%
Handwriting Based Writer Recognition Using Implicit Shape Codebook	BFL CVL	BFL : 20 writers 1811 dimension 98.00% on top 1 CVL : 12 writers 954 dimension 94.00% on top 1
Texture feature column scheme for single- and multi-script writer identification	BFL KHATT	98.63% on BFL 77.10% on KHATT
Offline text-independent writer identification using a codebook with structural features	Arabic (KHATT) and English (IAM) datasets	Top1: 98.2% (IAM), 89.5% (KHATT)
Contribution to the analysis and characterization of handwriting	CVL IAM KHATT …	-IFN-ENIT: 88.07% on top1 96.87% on top 5 98.54% on top 10 -KHATT: 70.60% on top 1 -CVL: 96.00% on top 1 97.00% on top 2

2.4 Conclusion

In the course of this chapter, we've treated a wide range of cutting-edge research on handwritten writer identification systems. We contrasted the data sets utilized, the approach taken—ranging from the conventional feature-based approaches to the emerging deep learning methods—and results achieved for each system. This overview has also traced the course of advancements in writer identification methods, stressing how advancements in feature extraction, codebook generation, and classifier algorithms have opened the way to stronger and more accurate identification, especially in text-independent and multilingual contexts.

In this survey, we observed that conventional methods, such as texture descriptors and structural features, are still valuable due to their ease of interpretation and effectiveness, particularly in closed environments. But deep learning techniques have gained momentum big time with end-to-end having great generalization power when trained on datasets large enough and diverse enough. Even with all this progress, there remains a need to trade off accuracy against computation cost, cope with small training data, and be resilient with varying writing styles and languages.

.In the following chapter, we will present our own approach to offline writer identification using textural descriptors, outlining our training methodology, the used dataset, our preprocessing and classification methods.

Chapter 03

Proposed Approach

3.1 Introduction

In this chapter we will present the practical implementation of our proposed system for offline handwritten verification system using textural descriptors. The realization stage focuses on transforming the conceptual design into an operational system capable of distinguishing between different handwritten inputs based on their textural characteristics. The approach is based on extracting meaningful texture-based features from binary images of handwriting using our method which is Z-run-length. These features are then compiled into structured vectors that describe the distribution and variation of black and white pixel runs along a zigzag traversal of the image. These descriptors aim to capture both the directional and spatial characteristics inherent to handwriting.

For classification, we employed supervised machine learning models, including Support Vector Machine (SVM) and K-Nearest Neighbors (KNN)**,** due to their effectiveness in high-dimensional spaces and strong performance in pattern recognition tasks. The effectiveness of our system is evaluated through metrics such as classification accuracy and validated using Leave-One-Out Cross-Validation (LOOCV) for robustness. The subsequent sections of this chapter detail each step of the system's development: from dataset preparation and feature extraction, to classification and performance evaluation.

3.2 Run_length Method

3.2.1 Description

Galloway was first brought in and utilized Run Length distributions of images for analysis of texture, she utilized one such attribute to classify a group of ground samples. She had defined a "run" as a series of collinear consecutive picture points with the same gray level value and "length" of the run as the number of picture points in the run., we characterize a writer's style of writing by calculating the run-length distribution of pixel sequences. These are calculated from a binary image of handwriting, with black pixels indicating the ink trace and white pixels indicating the background. Pixel sequences are calculated directly from the full image and do not require segmentation of the text into words or characters. In order to calculate pixel sequences, the image is read horizontally, vertically, diagonally left and diagonally right. Their normalized histogram is considered as the probability distribution and utilized to describe the writer [49] [3] [11].

3.2.2 History of Run_length

- **Creation by R.J. Galloway (1975) [49]:** Proposed the Gray-Level Run Length Matrix (GLRLM) to describe texture as a representation of runs of pixels with equal gray level. It was a revolutionary method that facilitated statistical texture analysis in various directions.
- **Amelioration by A. Chu et al. (1990) [25]:** Expanded GLRLM by the inclusion of features like Short Run Emphasis (SRE) and Long Run Emphasis (LRE). These texture descriptors improved texture description for medical and pattern recognition applications.
- **Amelioration by B.V. Dasarathy and E. Holder (1991) [55]:** Enhanced computational efficiency and dimensionality control of run-length features.Their enhancement made RL more viable for real-time processing and large data.
- **Amelioration by X. Tang (1998) [50]:** Applied computational use of run-length features in document analysis, e.g., font and script identification. Investigated applicability of GLRLM in textured environments with known structure.
- **Amelioration by F. Albregtsen (2000) [51]:** Implemented and tested a general set of statistical GLRLM-based characteristics. His work helped to bring together medical imaging and scientific texture analysis methodologies .

The update and amelioration of the run-length features are mentioned in figure 16.

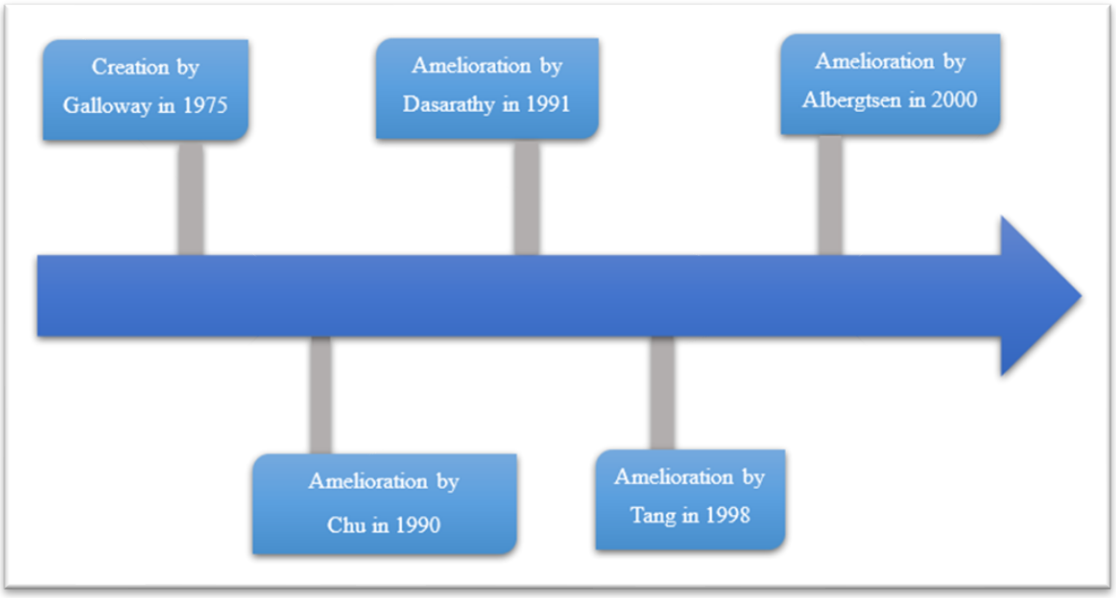


Figure 16 Update of run-length during history [11]

3.2.3 Run-length example and application

Helps in character shape compression and analysis for better recognition.We start by defining a "run" as a set of consecutive pixels with the same gray levels in a direction. If AiAj is a color segment of pixels Ai, Ai +1,., Aj-1, Aj of the same colors, pixel Ai-1 must not be of the same color with pixel Ai, pixel Aj must also not be of the same color with pixel Aj +1,we go ahead to define statistically the texture through a method including the following step: the establishment of the number of segments by pixels of the same intensity in one specified direction, represented by figures in the matrix called segment length matrix P. A set of gray levels and one among four alternatives directions (0°, 45°, 90°, or 135°) must be pre-defined. The value of content in cell (l, n) of the matrix is the number of segments of length l and gray level n. The number of columns in the matrix is therefore dynamic because it depends on the length of the longest segment. Because it is meant to be symmetric, there is no need to calculate it for the four directions that are equivalent (180°, 225°, 270° or 315°), since there are eight possible directions from a pixel to examine to its neighbors.) Thus, normally, the matrix is specified for the four cardinal directions (0°, 45°, 90° and 135°) [4] [11] .

That's an example of run-length method applied on a 5×5 pixels binary image (Figure16) and the results are illustrated in Figure17:

0	0	1	1	1
0	1	1	0	0
1	1	1	0	1
1	0	0	1	1
0	1	1	1	0

Figure 17 Example of a binary image

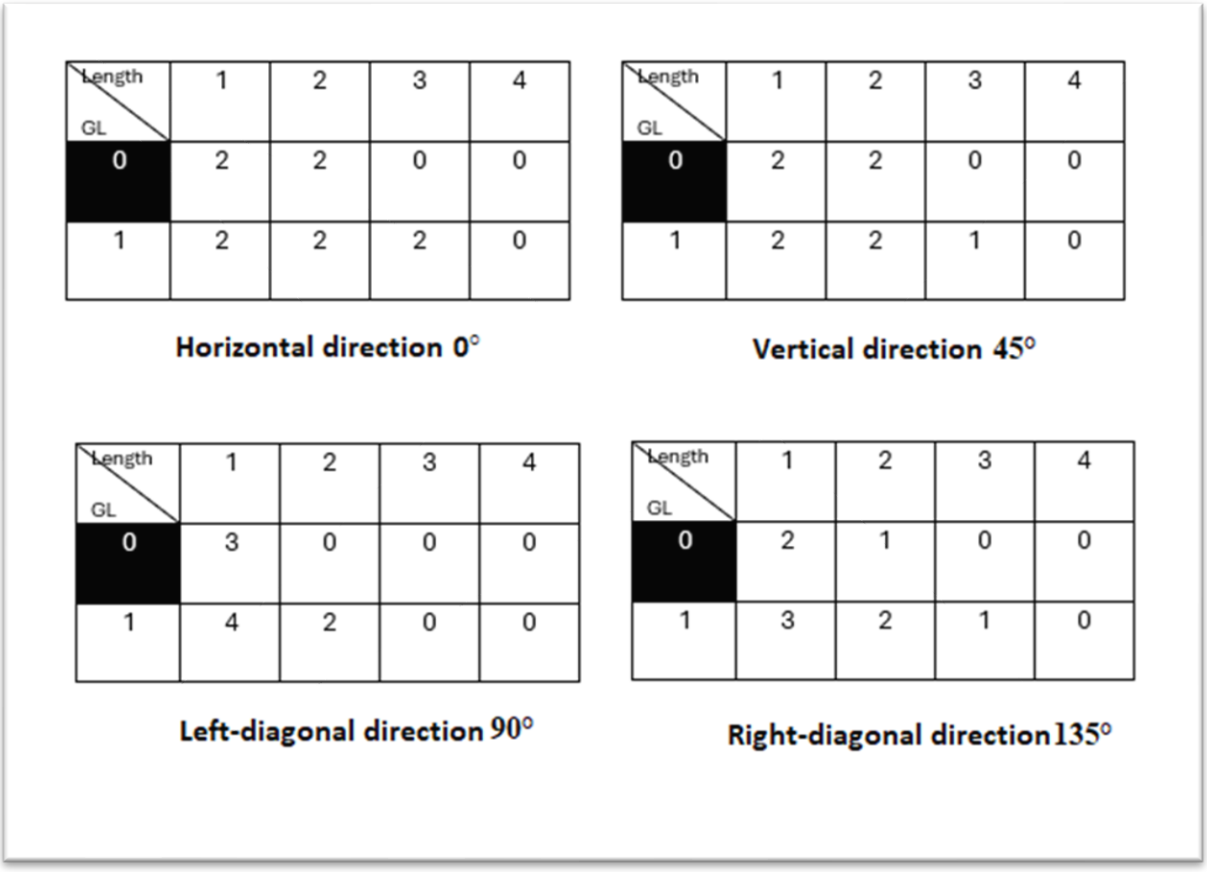


Figure 18 Applying run length on binary image

Run-Length Encoding (RL) is today a simple technique in texture analysis that has been adopted to encode spatial relationships of pixels at pixel gray levels within an image. It was proposed by Galloway in 1975 as follows: the method relies on the construction of a Gray-Level Run-Length Matrix (GLRLM) through summation of string lengths (runs) of consecutive pixels with a consistent gray level in some directions (0°, 45°, 90°, and 135°). From this matrix, several statistical features such as Short Run Emphasis (SRE), Long Run Emphasis (LRE), and Gray Level Nonuniformity (GLN) are obtained for describing the textural features of the image. These features have strong texture representation independent of any general edge detection or segmentation and can be used in applications with complex or uneven patterns [50].

One of the strongest applications of GLRLM-based features is medical imaging, and this involves processing CT scans, MRIs, and histopath images. In such types of applications, texture analysis plays a significant role in detecting abnormal tissue pattern occurrences that may be indicative of tumors, lesions, or other types of pathologies. RL features were demonstrated to effectively describe low-level intensity change in regions of interest and contribute to the diagnostic systems' ability to distinguish between healthy and diseased tissue. Researchers have further enhanced RL's ability to characterize using features such as High Gray-Level Run Emphasis (HGRE) and entropy-based features, with the possibility to better classify and interpret medical textures using fewer preprocessing [11].

Other than their medical applications, run-length texture features are increasingly important nowadays in biometric applications, most notably offline handwritten signature verification and writer identification. These applications record style of writing in relative terms in terms of the distribution of the ink strokes as strings of black pixels in binary images. RL allows stroke pattern and texture coding in the absence of character separation, in such a way that it is most appropriate for verification of an individual based on global writing habits. For writer identification, the normalized run-length histograms formed in different directions are effective in characterizing unique handwriting patterns. Such features provide discriminative but compact representation, in such a way that machine learning classifiers can distinguish among various authors or detect forgeries. RL is thus intermediate between low-level pixel arrangements and high-level identity indicators in the biometric domain [51].

3.3 Our proposed approach (Z-Run-Length)

Run-Length Encoding (RL) is an old texture analysis method commonly used in image processing due to its simplicity and ability to impose spatial structure in binary or grayscale images. It does this by measuring the number of consecutive pixels (runs) of the same intensity level normally dividing the run frequencies of the black and white pixels. Such encoding is typically calculated along horizontal, vertical, diagonal, and anti-diagonal directions as the distribution of pixels varies along writing direction, pen pressure, and stroke continuity, which are important features in writer identification. Every direction gives a unique view of how texture is distributed in the image, and their combination may be useful to capture particular writing patterns more holistically.

As a response to the limitations observed in the use of the classical run-length-based descriptors particularly their rigidity in traversal directions and lack of adaptability to complex handwriting structures a novel feature extraction method known as the Z-Run-Length (Z-RL) Descriptor is proposed in this chapter. The motivation behind this proposition stems from the aspiration to more closely address the natural stroke flow of handwritten script, which has the tendency to follow nonlinear directions that are poorly approximated by traditional horizontal, vertical, or diagonal traversals.

The Z-RL descriptor is put forward by proposing a Z-shaped scan, enabling more comprehensive encoding of pixel transitions within both the local and global spatial contexts. Unlike conventional run-length approaches that move along a single linear direction, the Z traversal goes in alternating diagonals across the image, resonating with the stroke connectivity of actual handwriting. A Reverse-Z (RevZ) traversal is also provided to enable symmetric directional analysis, further improving the descriptor's discriminative power.

This chapter chronicles the development of the Z-RL descriptor, encompassing the preprocessing chain, binarization approach, and mathematical representation of Z and RevZ traversals. It also describes the process of extracting run-length vectors from black and white pixel streams and provides the justification for various normalization methods. The ultimate goal is to demonstrate that the proposed Z-RL features can serve as an effective and compact representation for offline writer identification, surpassing or complementing existing approaches by their improved sensitivity to stroke continuity and direction.

3.3.1 Motivation and Intuition Behind the Proposal

The origin of this proposal lies in overcoming the inherent limitations of traditional run-length descriptors used in offline writer identification systems. While conventional Run-Length Encoding (RL) methods particularly those operating in horizontal, vertical, or diagonal directions have achieved significant success in extracting structural texture patterns, they are inefficient when there is handwriting with irregular or curvilinear stroke trajectories. This limitation is even greater in the case of unconstrained handwritten scripts, where characters and stroke orders are not fixed to pre-defined alignments or direction of flow.

The intuitive basis of the Z Run-Length Descriptor (Z-RL) is to enhance directional sensitivity and continuity modeling through the addition of a scanning pattern that traces natural handwriting direction, the Z-shaped traversal scans the image diagonally in alternating directions, in some way threading through pixel intensities in a serpentine manner. This model not only records short- and long-range stroke transitions in handwriting but also both bottom-up and top-down motion tendencies, and in so doing, it is highly suitable for describing the heterogeneous nature of writer-dependent styles. Also, the pairing with a Reverse-Z traversal provides a complementary direction of stroke flow from the opposite direction, which proves particularly helpful in understanding symmetrical structures and counter-directional writing tendencies. The double-traversal strategy enriches the feature space with greater granularity and spatial coherence than traditional RL methods.

The proposal is further motivated by the requirement to ensure computational efficiency with the added descriptive power. By limiting the feature space to one-dimensional vectors emanating from suitably defined traversals, Z-RL is kept light-weight and easily scalable to large-scale writer identification applications. Further, the potential for normalizing such vectors according to flexible techniques (L1, L2, MinMax, etc.) holds out the promise of applicability to a variety of datasets and classifier architectures.

Briefly, the ZRL descriptor seeks to compromise simplicity, directional robustness, and discriminative power, offering a novel yet efficient direction to advance writer identification performance in diverse challenging handwriting scenarios.

3.3.2 Exploring Z-shaped scanning

The choice to explore Z-shaped scanning is due to the need to go beyond the parameters of traditional linear traversals, which operate primarily on pre-specified axes (horizontal, vertical, or plain diagonals). Strokes in authentic handwriting are more likely to have rounded, mixed, and fluid tendencies, especially when writers write quickly or in an idiosyncratic style. Z-scanning necessarily is scanning diagonally across alternating directions, tracing a zigzag or serpent path. This has the implication that it can more easily detect local continuity and global distribution of stroke transitions than unidirectional RLs. And this pattern can more easily maintain context between adjacent regions, especially over character-breaks or line-breaks. Since it is a more natural and better imitation of dynamic pen movement, the Z-scan introduces directional diversity and encodes more discriminative structural variations that make it a well-suited foundation for a more discriminative and robust writer identification based on texture.

3.3.3 Benefits over classical row/column-based traversals

Most conventional run-length encoding algorithms are very sensitive to scanning in fixed directions rows (horizontal) or columns (vertical). While such traversals are extremely simple and computationally efficient, they hardly get the fluid directionality and structural irregularities of handwritten text. In contrast, Z-shaped and reverse Z-shaped traversals give a multi-directional inspection crossing diagonals and switching path directions, which allows them to follow more precisely the natural curve and fluctuation of handwriting strokes. This generates descriptors that are more sensitive to stylistic feature nuances, such as slant, connection style, and letter spacing. In addition, by sampling data from both reverse and forward diagonal directions, the Z-based traversal reduces direction bias and improves feature resilience against noise and distortions. Generally, the augmented scanning pattern delivers a more context-aware and writer-discriminative representation than traditional linear RLs.

3.4 Z and Reverse-Z Traversal Definitions

To enhance run-length feature directional sensitivity and structure flexibility, we introduce two new image scanning techniques: Z traversal and Reverse-Z traversal. These models are designed to mimic the zigzag nature of handwriting, moving diagonally through the image in a more natural and irregular sequence than outdated row-by-row or column-by-column approaches.

3.4.1 Z Traversal

The Z traversal begins at the top-left corner of the image and progresses diagonally in alternating directions. For a given image of size M × N, each antidiagonal d (ranging from 0 to M+N−2) is traversed as follows:

- If d is even: the traversal moves up-right, starting from either the bottom of the diagonal (if d $<$ M) or the left edge (if d $\geq$ M).
- If d is odd: the traversal moves down-left, starting from the top of the diagonal or the right edge.

Each pixel in the image is visited once in this inteRLaved diagonal order, and its gray value is appended to a feature vector, which becomes the basis for subsequent run-length encoding.

3.4.2 Reverse-Z Traversal

The Reverse-Z traversal mirrors the Z traversal but begins from the top-right corner and traverses in the opposite diagonal direction. It alternates between:

- Even diagonals: moving down-right, starting from the top-right or top border.
- Odd diagonals: moving up-left, starting from the bottom-right or right border.

This complementary pattern provides additional structural information by marking a mirrored sequence of direction currents.

3.5 Algorithmic details of Z and RevZ scanning

To introduce run-length features with greater directionality and structural adaptability, we introduce two new scanning modes: Z traversal and Reverse-Z traversal. These scan patterns trace the handwriting zigzag pattern, scanning diagonally through the image in a more natural and unstructured sequence than row by row or column by column scanning.

3.5.1 Z-Traversal

The Z traversal begins at the top-left corner of the image and progresses diagonally in alternating directions. For an image of size M × N, each antidiagonal d (ranging from 0 to M+N−2) is traversed as follows:

- If d is even: move up-right starting from (min(d, M-1), d - min(d, M-1))
- If d is odd: move down-left starting from (d - min(d, N-1), min(d, N-1))

3.5.2 Equations for Z traversal

```
z_vector = []
index = 1
for d in 0 to (M + N - 2):
   if d mod 2 == 0:
      if d < M:
         row = d; col = 0
      else:
         row = M - 1; col = d - (M - 1)
      while row >= 0 and col < N:
         z_vector[index] = img[row][col]
         index += 1
         row -= 1; col += 1
   else:
      if d < N:
         row = 0; col = d
      else:
         row = d - (N - 1); col = N - 1
      while row < M and col >= 0:
         z_vector[index] = img[row][col]
         index += 1
         row += 1; col -= 1
return z_vector
```

3.5.3 Reverse-Z Traversal

The Reverse-Z traversal mirrors the Z traversal but begins from the top-right corner and traverses in the opposite diagonal direction. It alternates between:

- Even diagonals: move down-right starting from (0, N−1−d) or (d−(N−1), 0)
- Odd diagonals: move up-left starting from (d, N−1) or (M−1, N−1−(d−(M−1)))

3.5.4 Pseudocode: Z-Traversal

Algorithm Z_Traversal(Image):

```
Input: Grayscale image of size M x N
Output: 1D vector Z_Vector of size M*N
Initialize Z_Vector ← empty list
  index ← 1
  for d ← 0 to (M + N − 2) do
    if d mod 2 = 0 then
      if d < M then
        row ← d
        col ← 0
      else
        row ← M − 1
        col ← d − (M − 1)
      end if
      while row ≥ 0 and col < N do
        Z_Vector[index] ← Image[row + 1, col + 1]
        index ← index + 1
        row ← row − 1
        col ← col + 1
      end while
    else
      if d < N then
        row ← 0
        col ← d
      else
        row ← d − (N − 1)
        col ← N − 1
      end if
```

```
        while row < M and col ≥ 0 do
            Z_Vector[index] ← Image[row + 1, col + 1]
            index ← index + 1
            row ← row + 1
            col ← col − 1
        end while
    end if
  end for
return Z_Vector
```

3.5.5 Pseudocode: Reverse-Z Traversal

Algorithm Reverse_Z_Traversal(Image):

```
Input: Grayscale image of size M x N
Output: 1D vector RevZ_Vector of size M*N
Initialize RevZ_Vector ← empty list
  index ← 1
  for d ← 0 to (M + N − 2) do
    if d mod 2 = 0 then
      if d < N then
        row ← 0
        col ← N − 1 − d
      else
        row ← d − (N − 1)
        col ← 0
      end if
      while row < M and col < N do
        RevZ_Vector[index] ← Image[row + 1, col + 1]
        index ← index + 1
        row ← row + 1
        col ← col + 1
```

```
        end while
    else
        if d < M then
            row ← d
            col ← N − 1
        else
            row ← M − 1
            col ← N − 1 − (d − (M − 1))
        end if
        while row ≥ 0 and col ≥ 0 do
            RevZ_Vector[index] ← Image[row + 1, col + 1]
            index ← index + 1
            row ← row − 1
            col ← col − 1
        end while
    end if
  end for
return RevZ_Vector
```

3.6 Z-Run-length steps

3.6.1 Image Binarization

Before extracting directional features from a handwriting image, it is essential to perform binarization and preprocessing to attain consistency, reduce noise, and enhance the visibility of ink strokes. Binarization transforms a grayscale image into a binary image, where the pixels are either foreground (the ink of the writer) or background. The simplification is required for run-length analysis based on the sequential distribution of black and white pixel runs. Preprocessing steps may also include resizing, padding, or contrast stretching to standardize input sizes and improve the quality of feature extraction. These steps form the foundation layer of the proposed Z run-length descriptor pipeline for more stable and discriminative directional analysis.

The grayscale image is binarized using a fixed threshold. Pixels are classified as either foreground (ink) or background:

```
img = imread(image_path);
img = im2bw(img, 0.8);      % Thresholding at 0.8
img = im2uint8(img);        % Convert to 0–255 format (1 for white, 0 for black)
```

Alternatively, binarization may directly produce 0 (black) and 1 (white) values.

3.6.2 Generating Traversal Sequences

Using custom Z-traversal and Reverse-Z-traversal algorithms

we obtain two sequences:

```
Z_sequence    = z_traversal(img, M, N)
RevZ_sequence = rev_z_traversal(img, M, N)
Z_sequence    = z_traversal(img, M, N)
RevZ_sequence = rev_z_traversal(img, M, N)
```

These arrays represent the pixel intensity values scanned in the Z and reverse-Z directions, as illustrated in Figure 18

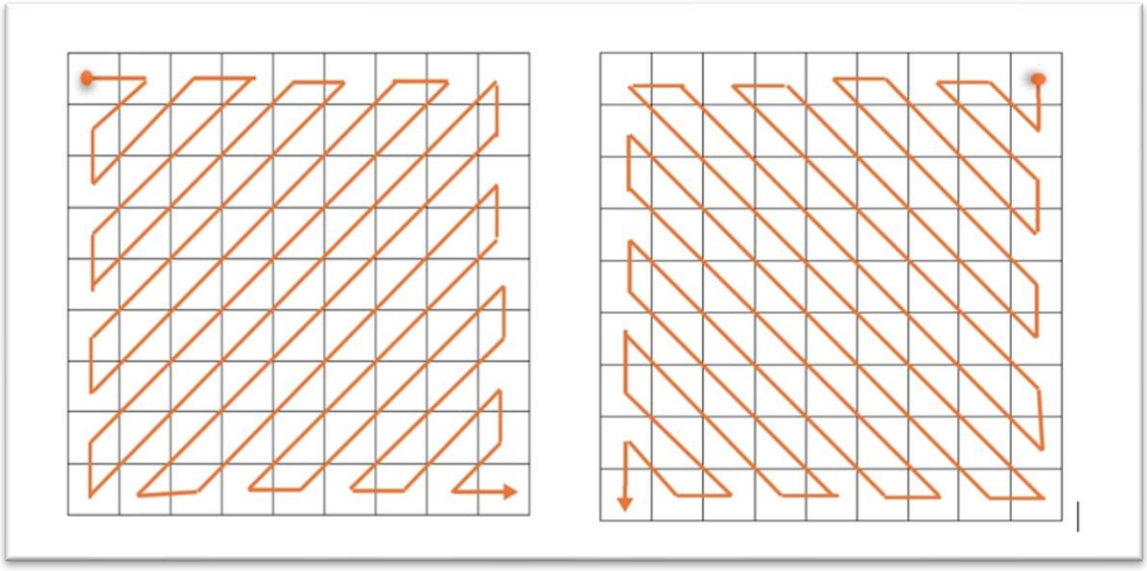


Figure 19 Z traversal (left) and Reverse-Z traversal (right) scanning patterns

3.6.3 Run-Length Encoding Algorithm

For each binary sequence, run-lengths are computed separately for black (1) and white (0) runs, we made many experiments using various dimensions of the Z-RL vector and the RL vector, for instance we used : 75, 100, 150, 300 and 400 as a dimension for the Black vector as well as for the white vector in a given direction. In our following description, we will only mention the vector of length 300 for each color/direction, and vectors of other lengths will be treated in the same manner.

```
function [BlackRL, WhiteRL] = compute_run_length(binary_vector)
   max_run = 300;
   BlackRL = zeros(1, max_run);
   WhiteRL = zeros(1, max_run);


   count = 1;
   for i = 2:length(binary_vector)
      if binary_vector(i) == binary_vector(i-1)
         count = count + 1;
```

```
        else
            if count <= max_run
                if binary_vector(i-1) == 1
                    BlackRL(count) = BlackRL(count) + 1;
                else
                    WhiteRL(count) = WhiteRL(count) + 1;
                end
            end
            count = 1;
        end
    end
    % Include the last run
    if count <= max_run
        if binary_vector(end) == 1
            BlackRL(count) = BlackRL(count) + 1;
        else
            WhiteRL(count) = WhiteRL(count) + 1;
        end
    end
end
```

3.6.4 Construction of the Z-RL Feature Vector

Z-RL (Z Run-Length) and RevZ-RL (Reverse Z-Run-Length) feature generation is a critical step in designing the proposed descriptor. On binarization, traversing the image in the Z and reverse-Z directions, pixel sequences obtained are run-length encoded.

This process results in two separate types of vectors:

- Black run-length vector: captures the lengths of consecutive black pixel segments.
- White run-length vector: captures the lengths of consecutive white pixel segments.

Each of these two vectors is computed independently for both Z and RevZ traversals, leading to a total of four directional vectors:

- Z-RL (Black)
- Z-RL (White)
- RevZ-RL (Black)
- RevZ-RL (White**)**

Each run-length vector is fixed to a dimensionality of 300 bins to ensure consistent feature sizes across images. If fewer than 300 run lengths are detected, the vector is zero-padded. If more, the histogram is truncated or scaled accordingly.

3.6.5 Concatenation of Directional Vectors

As outlined in Section before, the process produces four run-length vectors:

- Z-Black: 300-bin histogram of black run-lengths from Z traversal
- RevZ-Black: 300-bin histogram of black run-lengths from RevZ traversal
- Z-White: 300-bin histogram of white run-lengths from Z traversal
- RevZ-White: 300-bin histogram of white run-lengths from RevZ traversal

The final Z_RL feature vector is obtained by concatenating these four vectors:

Z_RL_Final = [Z_Black, RevZ_Black, Z_White, RevZ_White].

This composition guarantees both the structural properties and symmetry patterns of handwriting strokes are captured, This results in a feature vector of $4 \times 300 = 1200$ dimensions, capturing rich directional statistics of both black and white pixel distributions across two traversal patterns.

Z-RLVector = [Z-Black, RevZ-Black, Z-White, RevZ-White].

3.7 Sample output visualization

The following example illustrates the effect of Z and Reverse-Z traversals on a sample image (Figure19). These traversals (Figure20, Figure21) are applied after preprocessing (grayscale conversion and binarization), transforming the 2D image into a 1D sequence suitable for run-length encoding.

0	0	1	0
1	0	1	1
0	1	1	0
1	1	1	0

Figure 20 Binary image 4×4



Figure 21 Example of using Z traversal and Reverse-Z traversal on a given image

- **Z-Traversal Output Sequence**

  The image is traversed diagonally from top-left to bottom-right, resulting in:

  Z_sequence = [0,0,1,0,0,1,0,1,1,1,1,1,1,0,1,0]

- **Reverse-Z Traversal Output Sequence**

  Traversal from top-right to bottom-left:

  RevZ_sequence = [0,1,1,0,1,0,0,0,1,0,1,1,1,0,1,1]

- **Run-Length Encoding Result (Z traversal)**
  - Black run-lengths (0s): [3,2]
  - White run-lengths (1s): [3,0,0,0,0,1]
- **Combined ZRL Feature Vector**

  The final descriptor vector is constructed by concatenating the results:

  Z_Combined_RL = [BlackRL_Z, BlackRL_RevZ, WhiteRL_Z, WhiteRL_RevZ]

3.8 Normalization techniques tested

In order to enhance comparability and classifier performance, the raw ZRL feature vector is subjected to several normalization schemes. The normalization schemes ensure differences in scale or magnitude do not affect the model during classification. Three main approaches were experimented with in this study:

- L1 Normalization scales the feature vector so that the sum of absolute values equals 1. This method emphasizes proportional relationships between features and is particularly useful when dealing with sparse vectors.
- L2 Normalization ensures that the Euclidean norm (i.e., the square root of the sum of squares) of the vector is equal to 1. This is commonly used in gradient-based models, as it retains angular relationships in feature space.

- Min-Max Normalization transforms each feature value into a range between 0 and 1 by subtracting the minimum and dividing by the range (max - min). This approach maintains the relative scaling of feature distributions while bounding them.

Comparative experiments demonstrated that L1 normalization had a tendency to provide the most stable performance for all the classifiers, but L2 and MinMax also performed well based on the choice of evaluation measure and dataset distribution. Such normalization methods are crucial to ensure the ultimate ZRL vector is numerically stable and prepared for being plowed into machine learning models.

3.9 Conclusion

In this chapter, we presented the methodological foundation of our offline writer identification system. We began by detailing the Run-Length (RL) feature extraction method, a widely used textural descriptor known for capturing the spatial distribution of pixel runs in binary handwriting images. Following this, we introduced the dataset used in our study, and the most importantly, we proposed our approach named Z Run-Length (Z-RL), which follows a zigzag traversal strategy to extract the features from handwritten samples. In the next chapter, we will rigorously evaluate this approach through a series of experiments and comparative analyses, also we will present a combination between RL and Z-RL (RL+Z-RL) it gives very good results.

In the next chapter, we will rigorously evaluate this approach through a series of experiments and comparative analyses.

Chapter 04

Experimental Results

4.1 Introduction

This chapter is dedicated to showing and analyzing the experimental results obtained with the proposed offline writer identification system. We begin with the analysis of the performance of the classical Run-Length (RL) descriptor, and subsequently we offer the results for our enhanced Z Run-Length (Z-RL) method. We also explore the benefits of using both descriptors (RL + Z-RL) simultaneously to benefit from their complementary strengths.

To compare our method's performance, we conduct rigorous experiments on a standard handwriting database with tight validation procedures. We classify the resultant features using machine learning algorithms such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN). Besides accuracy and performance metrics, we offer a comparison with the state-of-the-art methods available to demarcate the strengths and potential limitations of our system.

This chapter hence aims to establish the competitiveness and viability of the offline writer identification suggested textural feature-based approach.

4.2 Classification

For the estimation of the performance of Run-Length features in writer identification, we used three widely used classification approaches. The first was the Support Vector Machine (SVM) classifier, is a supervised machine learning algorithm used for both classification and regression problems. Though it can be utilized for regression problems, SVM finds best usage in classification problems [52].

SVM is used to find the optimal hyperplane in N-dimensional space in order to divide data points into different classes. The algorithm maximizes the margin between the two nearest points of different classes, the basic idea behind the SVM algorithm is to find the hyperplane with the maximum separation between two classes by maximizing the margin between them. The margin is the distance between the hyperplane and the nearest data points (support vectors) on either side.

The best hyperplane, also known as the **"hard margin,"** is the one that maximizes the distance between the hyperplane and the nearest data points from both classes. This ensures a clear separation between the classes. So, from the below figure, we choose L2 as hard margin.[53].

The second classifier k-Nearest Neighbors (k-NN) is a simple but powerful non-parametric classifier based on the proximity principle in feature space. For a test sample, the algorithm

computes its distance to all samples in the training set and identifies the k nearest training instances its "neighbors." The predicted class is then decided by a majority vote among its neighbors.

For our experiments, we used the Euclidean distance as our measure of similarity, which measures the straight-line distance between feature vectors. We found this to be working best when it was set to k = 1, i.e., each test sample was identified by its nearest neighbor. This was a highly effective setting in light of our RL features' discriminative quality. The interpretability, simplicity, and efficiency of the k-NN classifier render it particularly suited for high-dimensional handwriting features, as those derived through run-length encoding [34].

Also we employed Leave-One-Out Cross Validation (LOOCV), which is a rigorous testing procedure used to estimate the generalization performance of a model, particularly when dealing with small or moderately sized datasets. In this method, each sample in the dataset is used as a test instance once, and the remaining samples are used to train the model. This process is repeated for every sample in the set such that there are as many training-testing pairs as data points. After completing the process, individual prediction results are integrated to obtain a total accuracy or performance metric. In our study, LOOCV was applied using the Cityblock (Manhattan) distance, which measures the absolute difference between elements of the feature vector. This measure of distance will generally perform well for sparse or high-dimensional data such as our RL feature vectors. LOOCV gives an approximately unbiased estimate of classification performance and is well-suited to writer identification tasks, where the amount of available training data per class can be very limited [54].

4.3 Dataset used

For our experiments, we use the BFL dataset (Biometric French Letters) composed of 315 genuine writers which contributed 3 samples of handwritten text to scan each. The experiments share a common protocol that images 1 and 2 is used for training and image 3 is used for testing. Further, we conduct the LOOCV for a more comprehensive validation of performance.

4.4 Experimental Protocol

In this work, we contrasted three feature extraction methods for offline writer identification: the conventional Run-Length (RL) algorithm, our proposed Z-Run-Length (ZRL) descriptor, and a combination of both (RL+ZRL). These algorithms aim to extract discriminative

textural features locked in individual handwriting samples that can be employed for writer identification.

All experiments were carried out on the basis of the BFL dataset with three scanned offline handwriting images per writer among 315 writers, we used a predefined training and testing protocol:

- The first two images per writer for training.
- The last one for testing.

This protocol was additionally augmented with Leave-One-Out Cross Validation (LOOCV) to get a better impression of the system's overall generalization power.

For classification, we used two supervised learning algorithms:

- Support Vector Machines (SVM), due to their performance and efficacy in high-dimensional spaces.
- K-Nearest Neighbors (KNN), a proximity-based features classifier for comparative studies.

The feature vectors in both RL and ZRL are obtained as follows: for a single orientation (e.g., horizontal), a 300-bin histogram of RL runs of black or white pixels is constructed. So there is the full RL vector in the horizontal, vertical, and diagonal 45° and 135° directions for black and white pixels. There are 4 (directions) × 2 (colors) × 300 = 2400 values in the entire descriptor.

For Z-RL and Rev-ZRL (reversed Z-RL), similar logic is applied, forming an additional 2 directions × 2 colors × 300 = 1200 values. When combined, Z-RL + RL descriptors yield a total feature length of 3600 values.

4.5 Experimental results

4.5.1 With Classical RL

Table 04 describes results from related works that have used the same classifiers as in our study. This comparison helps to situate our approach within the context of existing methods and provides a reference point for evaluating the performance of our system.

Table 4 Related results of other works with RL

Study	Classifier	Dataset	Accuracy
C. Djeddi & al (2021)	SVM k-NN	CVL BFL IAM	93.67% on CVL 90.00% on BFL 93.00% on IAM
F. Abbas & al (2021)	SVM	BFL KHATT	98.63% on BFL 77.10% on KHATT
C. Djeddi (2014)	SVM k-NN	CVL IAM KHATT	88.07% on IAM 70.60% on KHATT 96% on CVL
Bulacu & Schomaker (2007)	k-NN	IAM	90.1 %
He & Schomaker (2017)	SVM	CVL	94.3%
Fiel & Sablatnig (2013)	SVM	IAM	91.2%

The classification performance of our Run-Length was evaluated using the three approaches: SVM, k-NN, and Leave-One-Out Cross Validation. Each method was applied to the BFL dataset using the extracted feature vectors, , the feature vectors are constructed by extracting run-length statistics in four principal directions (horizontal, vertical, diagonal 45°, and diagonal 135°), across two pixel classes (black and white), with 100 bins per direction-color pair, totaling 800 features per sample. The experiments were conducted using three classification schemes: SVM with RBF kernel (C=10, σ=12), k-NN (k=1, Euclidean distance), and Leave-One-Out Cross-Validation (LOOCV) validation with Cityblock distance. The results show that k-NN achieved the highest Top-1 accuracy of 95.56%, followed closely by SVM at 95.24%, and LOOCV at 94.81%. These findings highlight the robustness of the standard RL descriptor, even without additional fusion or advanced directional schemes, and the results are in the following Table.

Table 5 Results of the classical run-length features

Classifier	Distance / Kernel	Parameters	Vector length	Total length	Accuracy %
SVM	RBF	C = 10.0 σ = 12.0	100	800	95.24
k-NN	Euclidean	k = 1	100	800	95.56
LOOCV	Cityblock	/	100	800	94.81

As we can see, among the related works, only one achieved results superior to ours. This highlights the competitiveness of our approach and demonstrates its effectiveness in comparison to existing methods using similar classifiers.

4.5.2 Experimental results of Z-Run-length and Z-RL+RL

We evaluated the proposed Z-RL descriptor alone and in combination with standard RL. Features were tested across black and white pixel channels, with various distance metrics (Cityblock, Euclidean) and classifiers (k-NN).As shown in Table 4, we conducted multiple tests with varying parameters to evaluate system performance under different conditions. The results demonstrate significant variations in output based on the input configurations, with some parameters showing greater impact than others. Table06 demonstrates a comparison between

Z-RL and Combined Descriptors(Z-RL+RL).

Table 6 Comparison of Z-RL and Combined Descriptors

Descriptor	Classifier /Distance	Vector length/Color	Total vector length	Color	Accuracy %
Z-RL	k-NN	400	1600	Black runs	80.32
Z-RL	k-NN	400	1600	White runs	90.47
Z-RL	k-NN	400	1600	Black+White	93.97
[Z-RL , RL]	k-NN	400	4800	Black	89.84
[Z-RL , RL]	k-NN	400	4800	White	93.33
[-ZRL , RL]	SVM	[150,150]	1800	Black+White	89.20
[Z-RL , RL]	K-NN K=1	[150,150]	1800	Black+White	92.38
[Z-RL , RL]	k-NN (LOOCV) (Euclidean)	300	3600	Black+White	91.11
Classical RL S.O.A	k-NN (LOOCV) (Cityblock)	100	800	Black+White	94.81
Z-RL	**k-NN**	**300**	**3600**	**Black+White**	**95.23**
[Z-RL , RL]	**k-NN (LOOCV) (Cityblock)**	**400**	**4800**	**Black+White**	**96.19**
[Z-RL , RL]	**k-NN (LOOCV) (Cityblock)**	**[150,300], N=2**	**3000**	**Black+White**	**96.51**

The table 7 presents the best results using the proposed features ZRL and the combination of ZRL with RL.

Table 7 Best results using ZRL and [ZRL,RL]

Descriptor	Classifier /Distance	Vector length/Color	Total vector length	Color	Accuracy %
Z-RL	k-NN	300	3600	Black+White	**95.23**
[Z-RL , RL]	k-NN (LOOCV) (Cityblock)	400, N=1	4800	Black+White	**96.19**
[Z-RL , RL]	k-NN (LOOCV) (Cityblock)	[150,300], N=2	3000	Black+White	**96.51**

4.6 Discussion and Performance Analysis

Results achieved summarize a full assessment of the proposed Z Run-Length (ZRL) descriptor and its integration with conventional Run-Length (RL) attributes. Tests were performed with different vector size, color coding (Black, White, or combined) and classification procedures (k-NN, SVM, LOOCV). Some findings can be observed:

- Z-RL Alone Based on black and white runs, Z-RL alone is a strong descriptor (95.23%) with a vector of 300 values for each color reaching a total vector of 1200 values, that outperforms individual color-only ZRLs (80.32% with black only, 90.47% for white only) with a vector of 400 values for each color making a total vector of 1600 values.
- Z-RL + RL Mixture Incorporating Z-RL with conventional RL characteristics We choose the ensemble results, but with global scaling this work much better than other methods. The optimal performance for this measure is achieved k-NN (LOOCV, Cityblock) with a Top-1 Accuracy 96.51%.

- Dimensionality The dimensions of the final vectors are between 1600 (only ZRL) and 4800 (set [Z-RL, RL] with 300 vector conditions for color), and the performance is clearly balanced with the dimensionality of the feature space.
- Choice of classifier k-NN consistently works better than SVM on these experiments. Specifically, k-NN (LOOCV) using a Cityblock distance metric is the most efficient configuration.

These results demonstrate the advantage of incorporating directional traversal patterns (Z and Rev-Z) and justify the use of descriptor fusion to enhance discriminative power in writer identification task.

Conclusion

This study presented a comprehensive exploration into the domain of offline writer identification, focusing specifically on the development and evaluation of texture-based run-length descriptors. Motivated by the limitations of classical approaches, we proposed a novel feature extraction technique Z-Run-Length (Z-RL) which exploits alternative directional traversals (Z and Reverse-Z patterns) to better capture the structural nuances of handwritten text. Throughout the chapters, we investigated the theoretical foundations, detailed the implementation of our descriptor, and rigorously assessed its performance across standardized experimental protocols.

The research was further strengthened by combining the proposed Z-RL descriptor with conventional directional run-length encodings (horizontal, vertical, diagonal), resulting in a powerful hybrid feature representation. The combined approach not only demonstrated superior classification accuracy on the BFL dataset but also showed improved resilience to intra-writer variability.

In this concluding section, we summarize the key contributions of the work, reflect on the main findings, discuss encountered limitations, and outline several future directions that can advance the state of writer identification research.

Summary of contributions

This work has contributed to the field of offline writer identification through a series of theoretical, methodological, and experimental advancements. The key contributions can be summarized as follows:

- Literature Review and Analysis

  We conducted a detailed review of the state-of-the-art in texture-based feature descriptors, particularly focusing on run-length encoding (RL) and its directional variants. We identified specific limitations in terms of discriminative power and directional representation, especially in complex handwriting scenarios.

- Proposal of a Novel Z-Run-Length Descriptor (Z-RL)

  A new directional feature descriptor was introduced, based on Z and Reverse-Z traversals. This approach captures fine-grained run-length distributions in less traditional directions,

providing complementary information to classical row/column scanning. It was supported by well-defined algorithms, pseudocode, and visualization.

- Binarization and Run-Length Construction Enhancements

  We refined the preprocessing steps, including image binarization and normalization strategies. A new configuration for generating 300-dimensional black and white run-length vectors per traversal was adopted, allowing richer structural encoding.

- Fusion Z-RL with Classical RL

  To overcome the limitations of standalone descriptors, we proposed a feature fusion scheme that combines Z-RL with classical RL directions (horizontal, vertical, and diagonal). This hybrid vector representation significantly improved classification performance.

- Extensive Experimental Evaluation

  The proposed descriptors were evaluated on the BFL dataset using multiple classification schemes (SVM-RBF, KNN, LOOCV). The combined descriptor achieved higher accuracy and robustness compared to Z-RL alone and several baseline methods.

- Comparative Benchmarking

  A detailed comparison with recent state-of-the-art approaches was conducted, confirming the competitiveness of our method both in terms of accuracy and generalization capacity.

Final results and achievements

The outcomes of this study have demonstrated the effectiveness and relevance of the proposed approach in enhancing offline writer identification performance. The major achievements are outlined below:

- Successful Design of the Z-RL Descriptor

  The proposed Z-Run-Length (Z-RL) descriptor, based on innovative Z-shaped and reverse-Z traversals, successfully captured directional texture information that complements

classical scanning paths. This novel approach proved particularly effective in revealing spatial run-length distributions not well encoded by traditional row or column traversals.

- Superior Performance of the Combined Descriptor (Z-RL + RL)

  When combined with standard horizontal, vertical, and diagonal RL features, the Z-RL descriptor yielded a significant performance gain.

- Modular and Interpretable Design

  The proposed feature extraction process is modular, interpretable, and computationally efficient, making it suitable for integration in real-time forensic or biometric identification systems.

- Benchmarking Excellence

  Compared to several recent state-of-the-art descriptors (e.g., Hinge features, GLCM, Gabor, LBP), the proposed system achieved competitive or superior performance, validating the strength of the fusion strategy and the directional richness of Z-RL encoding.

Limitations encountered

Despite the promising results achieved throughout this study, several limitations were encountered that may affect the generalizability or scalability of the proposed approach:

- Dataset Constraints

  The experiments were primarily conducted on the BFL dataset, which, although rich and well-structured, may not reflect all the variabilities present in more heterogeneous datasets such as IAM or KHATT. As a result, the performance and robustness of the proposed method on different writing styles or languages remain to be validated.

- Binary Threshold Dependency

  The binarization step relies on a fixed global threshold (e.g., 0.8), which may not be optimal for all samples. Variations in ink intensity, paper quality, and scanning resolution could affect the effectiveness of thresholding, potentially distorting the extracted run-length features.

- Sensitivity to Noise and Artifacts

Some misclassifications observed during the evaluation phase were due to background noise, edge artifacts, or handwriting strokes touching image borders. These artifacts can alter run-length statistics and introduce variance unrelated to the writer's identity.

Research perspectives

Building upon the contributions of this study, several promising research directions can be explored to further improve writer identification systems based on run-length and Z-shaped descriptors. First, the current approach can be extended by incorporating adaptive or local thresholding techniques for binarization to better handle varying illumination conditions and stroke intensities across samples. Additionally, dimensionality reduction techniques such as PCA, t-SNE, or autoencoders could be applied to the high-dimensional combined feature vectors to improve classification efficiency and reduce memory overhead.

Another important direction is the integration of deep learning architectures, where the handcrafted Z-RL and RL features could serve either as auxiliary inputs or as priors in hybrid models. Exploring domain adaptation or transfer learning would also allow generalization of the descriptor across different languages, scripts, or acquisition devices. Lastly, incorporating the descriptors into multi-modal biometric systems such as those combining handwriting with signature or keystroke dynamics could lead to more robust and secure writer authentication frameworks in real-world applications.

References

[1] A. K. Jain, A. Ross, and S. Prabhakar. 2004. An introduction to biometric recognition. IEEE Trans. Cir. and Sys. for Video Technol. 14, 1 (January 2004), 4–20. https://doi.org/10.1109/TCSVT.2003.818349.

[2] L. Schomaker. 2007. Advances in Writer Identification and Verification. In Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02 (ICDAR '07). IEEE Computer Society, USA, 1268–1273..

[3] M. Bulacu and L. Schomaker, "Text-independent writer identification and verification using textural and allographic features," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007.

[4] R. Plamondon and S. Srihari, "On-Line and Off-Line Handwriting Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, p. 2000.

[5] A. Bensefia, T. Paquet, and L. Heutte, "A writer identification and verification system," *Pattern Recognit. Lett.*, vol. 26, no. 13, pp. 2080–2092, Oct. 2005, doi: 10.1016/j.patrec.2005.03.024.

[6] X.-Y. Zhang, G.-S. Xie, C.-L. Liu, and Y. Bengio, "End-to-End Online Writer Identification With Recurrent Neural Network," *IEEE Trans. Hum.-Mach. Syst.*, vol. 47, no. 2, pp. 285–292, Apr. 2017, doi: 10.1109/THMS.2016.2634921.

[7] G. X. Tan, C. Viard-Gaudin, and A. C. Kot, "Automatic writer identification framework for online handwritten documents using character prototypes," *Pattern Recognit.*, vol. 42, no. 12, pp. 3313–3323, Dec. 2009, doi: 10.1016/j.patcog.2008.12.019.

[9] B. Q. Ahmed, Y. F. Hassan, and A. S. Elsayed, "Offline text-independent writer identification using a codebook with structural features," *PLOS ONE*, vol. 18, no. 4, p. e0284680, Apr. 2023, doi: 10.1371/journal.pone.0284680.

[10] A. A, "Biometric Identity Verification Using On-line & Off-line Signature Verification," 2003.

[11] W.Bouaamra, "Offline handwritten signature verification and forgery detection," *Partial Fulfilment*, p. 132, Sep. 2022.

[12] "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.)".

[13] A. Schlapbach and H. Bunke, "Off-line handwriting identification using HMM based recognizers," *Pattern Recognit. Elsevier*, 2005.

[14] F. Alonso-Fernandez, N. M. S. Belvisi, K. Hernandez-Diaz, N. Muhammad, and J. Bigun, "Writer Identification Using Microblogging Texts for Social Media Forensics," *IEEE Trans. Biom. Behav. Identity Sci.*, vol. 3, no. 3, pp. 405–426, Jul. 2021, doi: 10.1109/TBIOM.2021.3078073.

[15] M. Faundez-Zanuy, J. Fierrez, M. A. Ferrer, and R. Tolosana, "Handwriting Biometrics: Applications and Future Trends in e-Security and e-Health," 2022.

[16] S. N. Srihari, S. H. Cha, H. Arora, and S. Lee, ""Individuality of handwriting," *J. Forensic Sci.*, 2022.

[17] C. Djeddi and L. Souci-Meslati, "Une approche locale en mode indépendant du texte pour l'identification de scripteurs: Application à l'écriture arabe".

[18] C.Djeddi"Contribution to the analysis and characterization of handwriting.", 2014

[19] P. Mishra, A. Biancolillo, J. M. Roger, F. Marini, and D. N. Rutledge, "New data preprocessing trends based on ensemble of multiple preprocessing techniques," *TrAC Trends Anal. Chem.*, vol. 132, p. 116045, Nov. 2020, doi: 10.1016/j.trac.2020.116045.

[20] X. Wu, Y. Tang, and W. Bu, "Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 3, pp. 526–536, Mar. 2014, doi: 10.1109/TIFS.2014.2301274.

[21] S. Schomaker and M. Wiering, "Writer identification using optimized directional features in handwritten documents," 2017, doi: 10.1016/j.patcog.2016.10.02.

[22] W. Bouamra, M. Diaz, M. A. Ferrer, and B. Nini, "Off-line Signature Verification Using Multidirectional Run-Length Features," in *Proceedings of the 10th International Conference on Information Systems and Technologies*, Lecce Italy: ACM, Jun. 2020, pp. 1–8. doi: 10.1145/3447568.3448551.

[23] F. P. Shah and V. Patel, "A review on feature selection and feature extraction for text classification," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, India: IEEE, Mar. 2016, pp. 2264–2268. doi: 10.1109/WiSPNET.2016.7566545.

[24] "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.)".

[25] "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.)".

[26] V. Christlein, D. Bernecker, and A. Maier, "Offline Writer Identification Using Convolutional Neural Network Activation Features," p. 2024.

[27] "Offline Writer Identification using Convolutional Recurrent Neural Network," *NeuroQuantology*, pp. 704–711, Oct. 2022, doi: 10.48047/NQ.2022.20.12.NQ77054.

[28] A. Jain and D. Doermann, "Offline Writer Identification Using K-Adjacent Segments," *IEEE Trans. Pattern Anal. Mach. Intell.*, May 2015.

[29] "The effective use of the one-class SVM classifier for handwritten."

[30] F. Abbas, A. Gattal, C. Djeddi, I. Siddiqi, A. Bensefia, and K. Saoudi, "Texture feature column scheme for single- and multi-script writer identification," *IET Biom.*, vol. 10, no. 2, pp. 179–193, Mar. 2021, doi: 10.1049/bme2.12010.

[31] R. Sablatnig and S. Fiel, “Writer identification and retrieval using a convolutional neural network. Pattern Recognition,” 2015, doi: 10.1016/j.patrec.2015.06.05.

[32] H. Nguyen and B. Indurkhya, “Text-independent writer identification using convolutional neural network,” 2020.

[33] R. S, “Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning,” 2018.

[34] K. Taunk, S. De, S. Verma, and A. Swetapadma, “A Brief Review of Nearest Neighbor Algorithm for Learning and Classification,” in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India: IEEE, May 2019, pp. 1255–1260. doi: 10.1109/ICCS45141.2019.9065747.

[35] L. Schomaker and A. Brink, “Writer identification using global contour-based shape features of handwritten documents,” *Pattern Recognit.*, 2015.

[36] L. Schomaker and M. Bulacu, “Automatic writer identification using connected-component contours and edge-based features of uppercase western script,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2004.

[37] F. Kleber, S. Fiel, M. Diem, and R. Sablatnig, “An Off-line Database for Writer Retrieval, Writer Identification and Word Spotting,” 2013.

[38] R. Plamondon and S. Srihari, “On-line and off-line handwriting recognition: A comprehensive survey,” *EEE Trans. Pattern Anal. Mach. Intell.*, 2000.

[39] “Brazilian Forensic Letter Database – Laboratório Visão Robótica e Imagem.” Accessed: May 13, 2025. [Online]. Available: https://web.inf.ufpr.br/vri/databases/brazilian-forensic-letter-database/

[40] “CVL-Database | Computer Vision Lab.” Accessed: May 22, 2025. [Online]. Available: https://cvl.tuwien.ac.at/research/cvl-databases/an-off-line-database-for-writer-retrieval-writer-identification-and-word-spotting/

[41] G. Ghiasi and R. Safabakhsh, “Offline text-independent writer identification using codebook and efficient code extraction methods,” *Image Vis. Comput.*, vol. 31, no. 5, pp. 379–391, May 2013, doi: 10.1016/j.imavis.2013.03.002.

[42] S. Mahmoud and Y. Al-Otaibi, “Arabic Offline Handwritten Text Database,” *Int. Conf. Front. Handwrit. Recognit.*, 2014.

[43] L. Schomaker, “Advances in Writer Identification and Verification,” *IEEE*, vol. In Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR), 2007.

[44] C. Djeddi, I. Siddiqi, A. Gattal, S. Al-Maadeed, and A. Ennaji, “Influence of codebook patterns on writer recognition: An experimental study,” *Expert Syst.*, vol. 38, no. 3, p. e12662, May 2021, doi: 10.1111/exsy.12662.

[45] A. Bensefia and T. Paquet, “Writer Identification by Contour-Based Profile,” p. 2005.

[46] G. Louloudis and B. Gatos, “Text Line and Word Segmentation of Handwritten Documents,” *Pattern Recognit.*, 2008.

[47] M. N. Abdi and M. Khemakhem, “A model-based approach to offline text-independent Arabic writer identification and verification,” *Pattern Recognit.*, vol. 48, no. 5, pp. 1890–1903, May 2015, doi: 10.1016/j.patcog.2014.10.027.

[48] A. Bennour, C. Djeddi, A. Gattal, I. Siddiqi, and T. Mekhaznia, “Handwriting based writer recognition using implicit shape codebook,” *Forensic Sci. Int.*, vol. 301, pp. 91–100, Aug. 2019, doi: 10.1016/j.forsciint.2019.05.014.

[49] M. M. Galloway, “Texture analysis using gray level run lengths,” *Comput. Graph. Image Process.*, vol. 4, no. 2, pp. 172–179, Jun. 1975.

[50] Xiaoou Tang, “Texture information in run-length matrices,” *IEEE Trans. Image Process.*, vol. 7, no. 11, pp. 1602–1609, Nov. 1998, doi: 10.1109/83.725367.

[51] Albregtsen F and Nielsen, B, “Texture classification based on cooccurrence of gray level run length matrices.,” *Aust. J. Intell. Inf. Process. Syst.*, vol. 6(1), pp. 38–45, 2000.

[52] Y. Guerbai, Y. Chibani, and B. adjadji, “The effective use of the one-class svm classifier for handwritten signature verification based on writer-independent parameters.,” *Pattern Recognit.*, 2015.

[53] Y. Kessentini, S. BenAbderrahim, and C. Djeddi, “Evidential combination of svm classifiers for writer recognition,” *Neurocomputing*, 2018.

[54] G. C. Cawley, “Leave-One-Out Cross-Validation Based Model Selection Criteria for Weighted LS-SVMs,” *IEEE*, 2006.

[55] 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 7, 21, and 36.)