Automatic off-line text-independent writer identification from handwriting

Abderrazak Chahi

THÈSE DE DOCTORAT EN CO-TUTELLE

DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

PRÉPARÉE À L'UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD

ET DE L'ÉTABLISSEMENT UNIVERSITÉ IBN TOFAIL DE KENITRA

École doctorale n°37

Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Informatique

par

ABDERRAZAK CHAHI

Automatic Off-line Text-independent Writer Identification from Handwriting

Thèse présentée et soutenue à Montbéliard, le 8 juillet 2021

Composition du Jury:

M. Mohammed El Hassouni	Professeur des universités FLSH, Université Mohammed V de Rabat	Rapporteur
M. Denis Hamad	Professeur des universités Université du Littoral Côté d'Opale	Rapporteur
M. Mohamed SBIHI	Professeur des universités EST Salé, Université Mohamed V	Examinateur
M. Jean-Christophe Burie	Professeur des universités La Rochelle Université	Examinateur
Mme Raja Touahni	Professeure des universités Faculté des Sciences, Université Ibn Tofail	Co-directrice de thèse
M. Youssef EL Merabet	Maître de conférences Faculté des Sciences, Université Ibn Tofail	Co-encadrant de thèse
M. Yassine RUICHEK	Professeur des universités Université de technologie de Belfort Montbéliard	Directeur de thèse

ACKNOWLEDGEMENTS

This dissertation project would not see the day without the unconditional support of my thesis advisor Pr. Yassine RUICHEK. I would like to thank him for his help, advice, and guidance throughout my Ph.D. I express all my respect and appreciation to him for the scientific knowledge and human values he imparted to me. Thanks to him, I was able to carry out my Ph.D. work in suitable conditions, thanks to all the technical resources he provided, his continuous support, and his valuable analysis and suggestions. Without his full help, it would have been more difficult to take the project to a higher level. Thanks to him, today, we have valuable scientific contributions that we have worked out together.

I owe a lot of thanks to my co-advisor Pr. Raja TOUAHNI for her continuous support, suggestions, and encouragement. Her directives had a great impact on the researcher I have become today. I address special thanks to Pr. Rochdi MESSOUSSI for providing the best conditions to work in the LASTID laboratory in Morocco, for his support during my research. Many thanks to Pr. Youssef EL MERABET for his valuable guidance, suggestions, and support for the success of my thesis project.

I would like to express my gratitude to my thesis committee, Pr. Denis Hamad, Pr. Mohammed El Hassouni, Pr. Jean-Christophe Burie, Pr. Mohamed SBIHI for accepting the evaluation of this work.

Many thanks to the members of the CIAD laboratory in Montbéliard and the LASTID laboratory in Kenitra for the favorable working environment, their support, and the friendly moments we shared during these years.

Special thanks to the French Government-Eiffel Fellowship and the Centre National pour la Recherche Scientifique et Technique (CNRST)-Maroc for their financial support.

I address my gratitude, thanks, and affection to my family, who have always been by my side since day one. Without their support, nothing would be possible. My work, contributions, and success are dedicated to them. Special thanks to my friends who have always been by my side.

Finally, I would like to thank everyone who has contributed directly or indirectly to the success of this thesis. THANK YOU!

CONTENTS

I

Context and Problem

1

INTRODUCTION

1.1/ HANDWRITING ANALYSIS

Handwriting is an effective behaviour identifier to portray the uniqueness of an individual. Many factors such as age, schooling, and emotional state can influence handwriting, which is technically called intra-class variance. Handwriting characteristics, i.e., contours, corners, transitions, thickness, etc., serve as discriminative features in biometric identification with the exact representation as in the face, iris, fingerprints, or DNA. Compared to electronic documents, handwriting provides more information about the person who created it.

According to the situation, geographical location, traditional and historical backgrounds, the form, rules, and style of handwriting change. While the learning process begins with copying forms from the standard ”copybook” (see Figure. 1.1), over time, each individual develops his or her own handwriting style as a skill of personal preference in drawing character shapes or combining them. A person will not produce exactly the same writing style twice. It is even impossible or rare for two individuals to have the same writing style. This variation is called interclass variance when it comes to dissimilarities between two texts produced by two individuals. The handwriting was an essential part of communication until the end of the twentieth century. Thus, there is a significant historical stock of handwriting that is the subject of research in many ways. Besides, the mass of handwritten documents continues to grow daily, and more and more industries and services require rapid processing techniques while ensuring the security of these documents. Machines equipped with a handwriting analysis system have been set up to meet the needs of these industries. Handwriting analysis is a challenging research area of pattern recognition that has attracted much attention for psychologists, graphologists, forensic scientists, and historians in recent decades. A system based on handwriting analysis applies the principles of artificial intelligence, more specifically, machine learning or deep learning. The goal is to equip computers to learn to recognize the shape and features of a character, word, or phrase

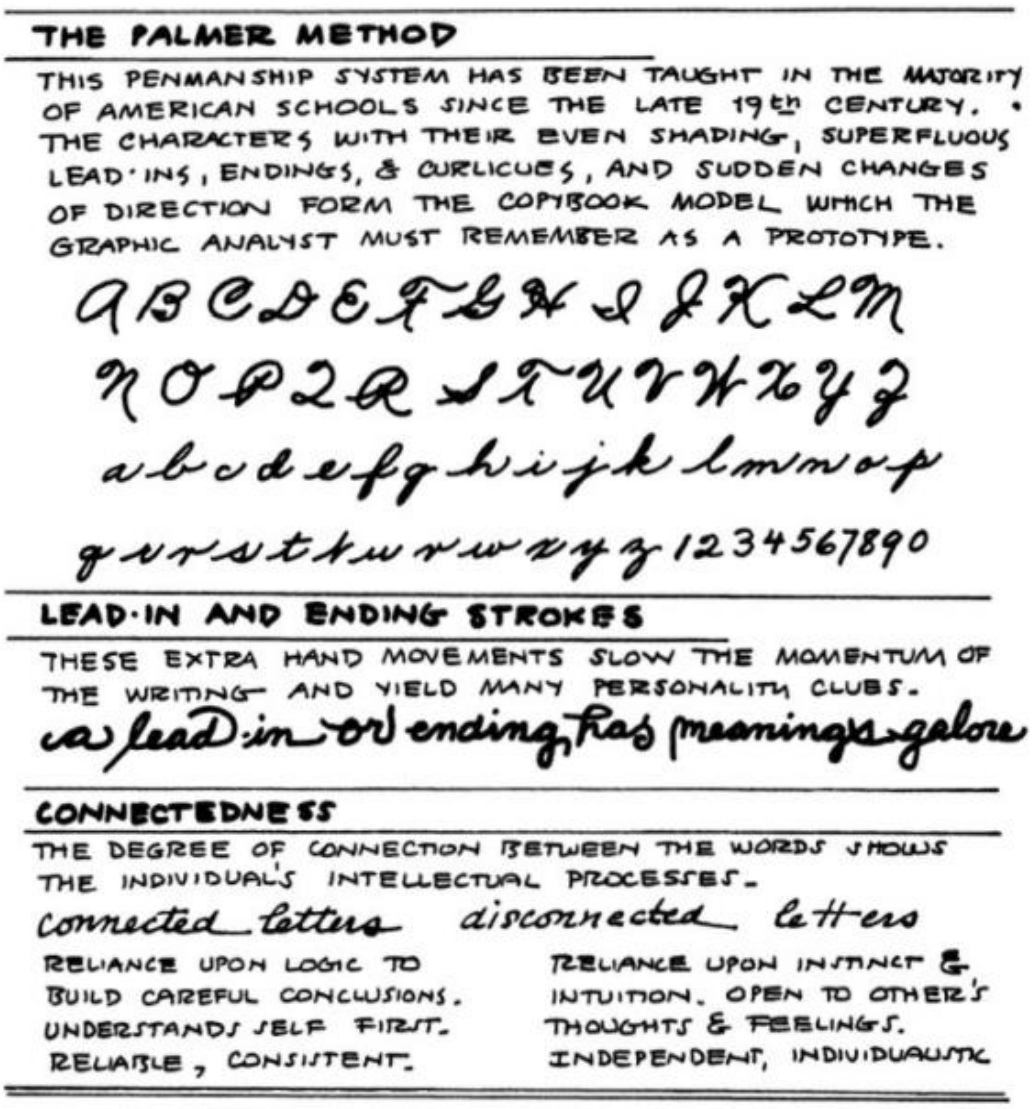


Figure 1.1: Sample copybook form. (Amend and Ruiz (2000))

In general, most of the approaches reported for handwriting analysis have focused on converting handwritten texts into uniform representations that are machine-understandable and easily reproducible. This field has become an important area of research with many scientific/technical locks and application challenges/potentials. The goal is to propose new concepts and reliable solutions for handwriting analysis and develop effective recognition systems that can be applied to different writing styles.

Research in the field of handwriting analysis has many applications in modern life. It covers, in full extension, a wide range of applications. One can cite online/offline verification

of handwritten signatures (Frias-Martinez et al. (2006)), handwritten musical scores for writer identification (Fornés et al. (2008)), online/offline writer identification (Chahi et al. (2020b); Abdi and Khemakhem (2015); Chahi et al. (2018)), classification of ancient documents (Arabadjis et al. (2013)), and smart meeting rooms for writer identification. It also finds application in the writer's gender, age range, and handedness (Liwicki et al. (2006)), forensic to identify the responsible behind fraudulent/threatening letters, ransom notes, and business agreements (Franke and Köppen (2001)). With the development of information security, handwriting is used as a biometric feature useful as a forensic tool for identity verification, validation, and authentication. It is a practical means of identification and has great significance in authenticating authorship of questionable documents, identifying forgeries, detecting alterations, verifying legal documents and cheques, or analyzing indented writings and historical documents.

From the perspective of graphology, handwriting is used to characterize personality traits to capture a person's attitudes, behaviors, and emotions. Therefore, handwriting is also referred to as brain writing because the manipulation of writing is done by the command of the brain, which is delegated to the nervous system, hand, arm and fingers. Thus, handwriting reflects mood swings and characterizes the writer's state of mind at that moment. Nowadays, there are handwriting tests that distinguish between certain medical disorders. These include shaking palsy, and Parkinson's disease (Ünlü et al. (2006)), or hardening of the arteries that supply the brain and those that supply the heart. Handwriting analysis can distinguish between those who suffer from arthritis and those who suffer from hypertension (Amend and Ruiz (2000)).

In summary, we believe that there is a great need for automatic methods to assist handwriting analysts, forensic experts, and scientists in their tasks and research, especially when dealing with large amounts of data. Computational algorithms for handwriting analysis facilitate the search space when comparing and matching questioned handwriting samples and extract useful information from the writing.

1.1.1/ CHALLENGES IN HANDWRITING ANALYSIS

The handwriting analysis is a challenging task because of its high contextual variation with different properties and handwriting form characteristics. "Handwriting is a complex motor skill that combines sensory, neurological, and physiological impulses. Factors such as visual perception and acuity, form comprehension, central nervous system pathways, and the anatomy and physiology of the bones and muscles of the hand and arm interact to produce the desired output." (Harrison et al. (2009)). Three factors cause variability in handwriting: biological, cultural, and interior factors.

1.1.1.1/ BIOLOGICAL AND CULTURAL FACTORS

Two fundamental factors contribute to the individuality of writing: genetic (biological or natural) and cultural (memetic). Several genetic factors influence handwriting and contribute to its uniqueness. First and foremost is the biomechanical structure of the hand (Bulacu (2007)): left- or right-handedness (Francks et al. (2003)), the corresponding sizes of the wrist and finger carpal bones, which strongly influence the pencil grip. Also, there are muscle strength, fatigability, peripheral motor dysfunction (Bulacu (2007)), and central nervous system (CNS) characteristics that affect fine motor control and influence handwriting (Van Galen et al. (1993)). Parkinson's disease, for example, impairs fine motor control, resulting in very shaky handwriting movements.

Cultural factors in handwriting biometrics are the culturally mediated influences on writing forms (allographic variation). Handwriting variability within a population is strongly influenced by the writing techniques taught in school and other factors such as geographic location and time, religion, types of schools, and learning the handwriting of others while imitating the writing style of parents (Bulacu (2007)).

The conclusion of whether the genetic factor or the cultural factor is the main factor has been investigated in several studies. Srihari et al. (2016) showed that children's writing skills increase with continuous learning, time, and practice. They stop copying texts and instead start writing from memory. At this point, children begin to develop their writing style. Besides, the school in which the children learned to write could be identified (Srihari et al. (2012)). Assuming that the genetic factor is the most dominant, the handwriting of the twins cannot be distinguished. Nevertheless, the opposite is true: even in identical twins, it was possible to determine their handwriting (Srihari et al. (2008)).

1.1.1.2/ INTERIOR FACTORS

Several conscious and unconscious factors determine the handwriting variability (Schomaker (1998)):

- *Affine transforms:* Scale, rotation, shear, and translation are transformations controlled consciously by the writer. In particular, slant (shear) is a common parameter determined by the pen grip and the wrist subsystem's orientation relative to the fingers (see Figure 1.2 (a)).
- *Neurobio-mechanical variability:* Handwriting is too sensitive to high intraclass variability depending on the state of the writer (mood, time, and effort). Figure 1.2 (b) shows different handwritings of the same word produced by the same scribe. This variation is technically known as neuro bio-mechanical variation, which is more related to system state than system identity.

- *Sequence variability:* As shown in Figure. 1.2 (c), characters can be reproduced with different stroke sequences. This may have implications for handwriting analysis using temporal data. This factor is also dependent on the instantaneous system state during the writing process and is interpreted by the sequencing variability.
- *Allographic variation:* As shown in Figure 1.2 (d), allographic variation refers to the inter-class variability between writers, considered as a discriminative information for the writer identification task. However, it causes most of the problems in automatic script recognition.

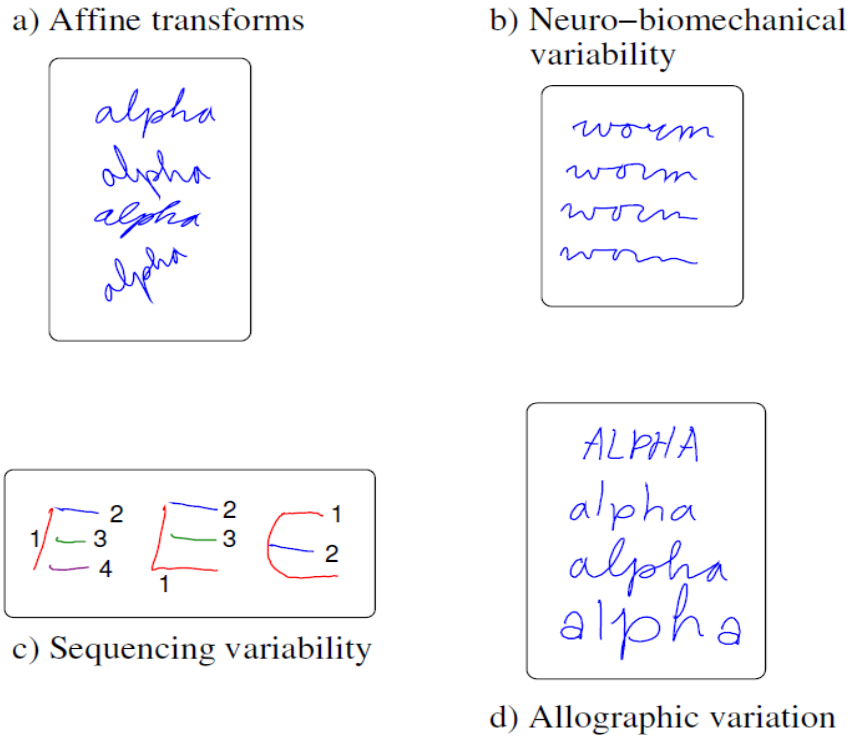


Figure 1.2: Interior factors for handwriting variability (Schomaker (1998)).

1.1.1.3/ WRITTEN LANGUAGE

Of the more than 7,100 languages in existence, there are at most 200 written languages. While there are many approaches to Latin handwriting analysis, this is no less the case with specific languages such as Chinese, Arabic, Hindu languages, etc. For example, very few works have dealt with the Arabic language. It has its characteristics and writing features which pose some difficulties to the existing systems. Arabic writing uses three-letter roots, with vowels not always written. This makes the reconstruction of handwritten words a difficult task. Moreover, every Arabic word is accompanied by diacritical marks.

Therefore, these characters may be automatically removed when using segmentation techniques in the pre-processing phase, affecting the system performance in characterizing the writing variability. Moreover, the same Arabic character may be written in different shapes depending on its position within the word or syllable, and it is visually written obliquely on the line rather than vertically as in most other languages. These various challenges further complicate handwriting analysis, as there are a variety of problems that need to be solved.

1.1.2/ DEFINITIONS

To investigate these challenges and develop a suitable, reliable, yet effective approach based on handwriting analysis, we need to define:

- The type of available data (How is the data collected and acquired ?)
- The writing content (how is the handwritten text present in the database ? Is it always the exact text ?)
- For what task is the eventual handwriting-based system to be used?

Based on these questions, handwriting analysis systems can be categorized into different groups: online or offline data, text-independent vs. text-dependent, as described below.

1.1.2.1/ ON-LINE VS. OFF-LINE DATA

On-line handwriting analysis systems use temporal and spatial characteristics of the writing captured through digitizing acquisition devices at the writing's real-time (e.g., Anoto pen). These characteristics are transmitted to computers for analysis using a particular transducer device, i.e., converting dynamic writing movements (see Figure 1.3) such as strokes, trajectory, height, speed, writing time, and pen pressure, etc., into a sequence of signals processed by computers. Handwriting analysis based on *on-line* data is expected to perform better than offline data as many significant features of writing are available during data acquisition. However, *offline* approach- based handwriting analysis remains a challenging and complex research topic. It can be defined as a static process that typically uses digitized handwritten images as input samples (which present allographic and textural variation). *Off-line* data can be derived relatively well from the image pattern of the writing. Thus, the image of the handwriting contains pattern features that are needed to characterize the handwriting style. To capture these features, various image processing and segmentation methods can be used. Research on handwriting analysis using offline data has focused on the pattern recognition and computer vision community

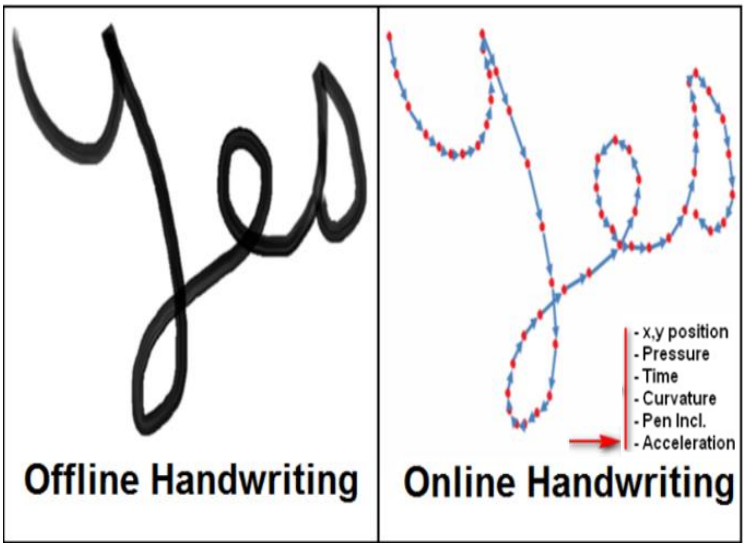


Figure 1.3: Example of online vs. offline handwriting word.

in recent decades. This is mainly due to its practical applications in security verification, behavioral biometrics, forensic document examination, and ancient document analysis.

1.1.2.2/ TEXT-DEPENDENT VS. TEXT-INDEPENDENT METHODS

Depending on how the writing content is present in the database, handwriting analysis systems can be further divided into two main types: *text-dependent* and *text-independent*. *Text-dependent* methods deal with the textual content of the writing, asking different scribes to produce the same fixed handwritten texts (e.g., signature verification). In general, these methods are not applicable in many practical cases due to their limitation to textual content (e.g., historical document analysis, forensic and identity verification).

In contrast, the *text-independent* method has no condition or restriction on the textual content, and any text can be analyzed and evaluated. It addresses the variation of the image writing texture with arbitrary texts and different character shapes, making the study of this mode more challenging.

1.1.2.3/ WRITER RECOGNITION

Relevant tasks for handwriting analysis are writer identification, verification, and retrieval. Writer recognition is the most general term that combines identification, retrieval, and verification. The current thesis does not focus on what is written. Instead, we study the problem of writer recognition, more specifically, writer identification and retrieval using offline data, in a text-independent manner.

For *writer identification*, the system works on the basic principle of the ”one-to-many” search technique within a large handwritten database. It uses multi-class ranking, where the output is a predicted list, sorted by class, of writers whose writing style matches that of the query sample. Identifying a person by their handwriting or signature is a form of behavioral biometric recognition. Thus, handwriting-based *writer identification* is considered as an important area of research that is valuable as a forensic tool for identity verification, validation, and authentication. For *writer retrieval*, the system searches and retrieves all document samples produced by a particular writer within an extensive database according to the similarity of handwritings (see Figure 1.4). As shown in Figure 1.5, *Writer verification* systems only compare and match a query handwriting sample with another sample. It answers whether two query handwriting samples are from the same person or not (e.g. signature verification).

Writer identification, verification, and retrieval rely on pattern recognition and machine learning techniques to characterize the writing variability. Note that in the remainder of this thesis, we follow the published vocabulary of this field and refer to writer identification for both identification and retrieval. This is because writer identification and retrieval can be mapped to the same process by ranking the most similar reference samples to the query one.

1.2/ WRITER IDENTIFICATION: MOTIVATIONS AND OBJECTIVES

A person’s writing is often recognizable, like faces or fingerprints. This feature attracts interest and presents a challenge for researchers to explore this area. It is an essential part of forensic document understanding and pattern recognition. This work addresses the problem of text-independent writer identification using handwriting images (off-line data). The motivation of this work stems from the need to improve behavioral biometric tasks, which have been mainly used for writer identification, to enhance security and forensic applications in today’s world. This can be achieved by developing near real-time, effective, and robust systems based on machine learning approaches. Some advantages and reasons sustain the ongoing study of handwriting patterns for writer identification. From the application point of view, one of the main advantages of handwriting-based writer identification is that it minimizes human intervention. Thus, the oldest techniques used by forensic examiners are tedious. With the advent of computerized handwriting analysis systems, writer identification is improved, and the search space for comparing, matching, and identifying the authenticity of unknown documents is reduced. Another important need for writer identification arises from the field of security and biometric verification. This refers to the potential use of handwritten words or small phrases to enhance real-world security applications in mobile and internet-based environments. To investigate

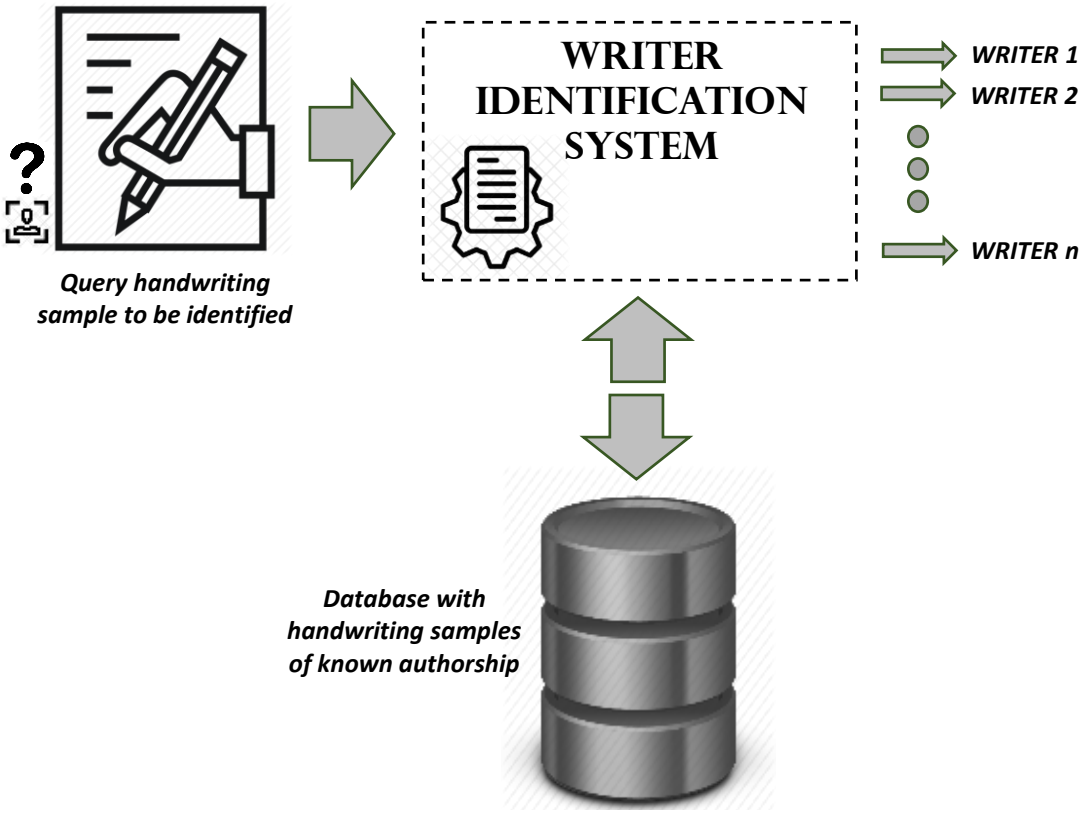


Figure 1.4: Writer identification system. The handwriting samples that are most similar to the query are retrieved from the reference database in a hit list.

Figure 1.5: Writer verification system. It compares two handwriting samples and automatically decides whether or not the same person wrote the input samples.

the task of writer identification, established benchmarks described in the literature are used to evaluate algorithms for writer identification. These databases contain extensive handwriting data with different scripts and languages.

Writer identification is a challenging task that has been considered in several application fields, ranging from preprocessing of handwriting images to biometric measurement and classical handwriting classification methods. The present work contributes to the

solution of numerous challenges that arise in these different stages. Writer identification systems face several challenges, mainly due to (i) the diversity of languages used worldwide. Each language has its own character form with different and complex writing styles; (ii) the variability of writing (cf. Section 1.1.1) and degraded documents with noise background and accidental writing traces; (iii) the collection and reorganization of handwritten databases is a difficult task. The goal is to develop efficient, real-time, robust, and generic approaches to correctly handle the identification task's writing style. Computational algorithms facilitate writer identification by assisting scribe analysts and forensic experts in reducing the search space to compare and match specific handwritten patterns within an extensive reference database.

Addressing these challenging problems raises some important research issues in computer vision: (1) How can we characterize the writing variability using automatic methods? (2) What kind of feature representations are most appropriate, and how can we combine them? (3) How can we exploit the feature representation of writing to identify its writer? (4) What performance and results can be achieved with these algorithms?

Our research aims to develop an effective and state-of-the-art writer identification system by exploiting theoretical and technical advances in image analysis and artificial intelligence. Our contributions concern all the main steps of an automatic system for identifying writers from handwriting:

- Image preprocessing step to reduce noise and unwanted details in scanned handwriting images.
- Segmentation of the preprocessed image into entities (words, characters, connected components, etc.) reduces the complexity of the subsequent processing modules.
- Feature extraction. It consists in defining, from the representation of the image, a synthetic description of the shape to be recognized in a space with multiple dimensions. In this work, texture descriptors and deep learning methods are proposed to characterize the variability of the writing. A dimensionality reduction step would be necessary to reduce the computational cost of the recognition process.
- Writer identification process. The query documents to be identified are automatically compared and matched in a large handwritten database. Our goal is to improve this step by developing new and robust distance-based approaches and appropriate learned models. In the second case, our goal is to design effective methods based on deep learning.

In this context, institutes such as the National Institute of Standards and Technology (NIST) and the International Association of Pattern Recognition (IAPR) offer many pro-

grams that promote pattern recognition and computer vision research. They regularly hold competitions to award the best performing learning systems. This work proposes to place itself in the middle of these international competitions for writer identification, gaining interest on scientific, social, and economic levels.

1.3/ THESIS OUTLINE

This thesis is structured in six chapters to give the readers a comprehensive presentation of the main contributions and prepare their background for a fluent experience.

Chapter 2 presents a comprehensive literature review of existing works for writer identification, which are compared thoroughly in our performance evaluation in Chapter 5.

Chapter 3 introduces the main stages of our proposed framework for writer identification. It details the different feature extraction methods we propose: Block Wise Local Binary Count (BW-LBC), zones-based handcrafted, Cross Multi-Scale Locally encoded Gradient Patterns (CLGP), and Local gradient full-Scale Transform Patterns (LSTP). This chapter also explains the image preprocessing and segmentation method used and the classification step employed to perform writer identification.

Chapter 4 shows that deep learning methods can be used to achieve further improvements in writer identification performance. It includes a comprehensive explanation of two proposed deep CNN models named $DeepWINet$ and $WriterINet$ for writer identification and a CNN framework for image retrieval for historical handwritten fragments (ICFHR2020 competition).

Chapter 5 is devoted to the study of experimental results obtained with our various approaches to writer identification and image retrieval for historical handwritten fragments. It describes in detail the handwritten databases used with the standard protocol setup to evaluate our proposed approaches, compares and discusses our achieved performance with SOTA systems. This chapter also highlights additional experiments conducted to further investigate the effectiveness and stability of the proposed methods.

Chapter 6 summarizes the research results presented in this thesis and outlines the overall conclusions with future research directions opened by the work reported here.

2

Writer Identification: Literature Review

2.1/ Introduction

Writer identification based on handwriting style recognition is considered as one of the most common research areas in pattern recognition and biometrics. It has received a lot of interest and attention in recent decades, as it is a challenging task considering the large within-writer and between-writer style variability. Several reviews (Rehman et al. (2019); Dargan and Kumar (2019)) have extensively addressed writer identification SOTA. The interest is to compare, evaluate and build reliable, near real-time and robust approaches that would provide high identification performance. In this research area, one needs to compute abstract and discriminative writing style features and extract details that reflect personal writing habits. This poses a great challenge due to the high sensitivity of the writing variability. Dealing with such extreme variations greatly improves the identification task and reduces misclassifications. Most existing work generally considers the following pipeline: pre-processing, feature extraction, and classification stages.

In this chapter, we survey well-known writer identification approaches proposed in recent years, as a result of the renewed interest in the scientific community for this research topic.

2.2/ Related Works

Here we summarize research work on writer identification. These approaches can be categorized into four groups: texture-based, grapheme-based, contour-based, and auto-learned methods.

2.2.1/ TEXTURE-BASED METHODS

Texture-based methods have been widely used for writer identification. Based on the assumption that handwriting can be considered as a texture image, texture features can be extracted within blocks, regions, fragments, words, lines of text, or the entire image. He and Tang (2004) presented a texture-based approach for Chinese off-line handwriting writer identification. They considered each character as a texture from which each feature vector is computed. The Weighted Euclidean Distance classifier (WED) is used to match and compare the extracted features for writer identification. Local binary pattern (LBP), local ternary pattern (LTP), and local phase quantization (LPQ) have been used in several recent approaches (Bertolini et al. (2013); Nicolaou et al. (2015); Hannad et al. (2016); Singh et al. (2018)) to capture texture features from handwriting and have shown promising results (writer identification performance) on benchmark handwriting databases.

Bertolini et al. (2013) proposed a texture-based system for writer identification using LBP and LPQ texture descriptors. These descriptors capture texture information from normalized blocks of writing. The system extracts small blocks and fills up the line while removing unwanted components considered noise. The normalized blocks are constructed from the filled lines as shown in Figure 2.1. The SVM classifier was used to identify the authorship of the query documents. This approach has produced remarkable results on a large database of 650 writers.

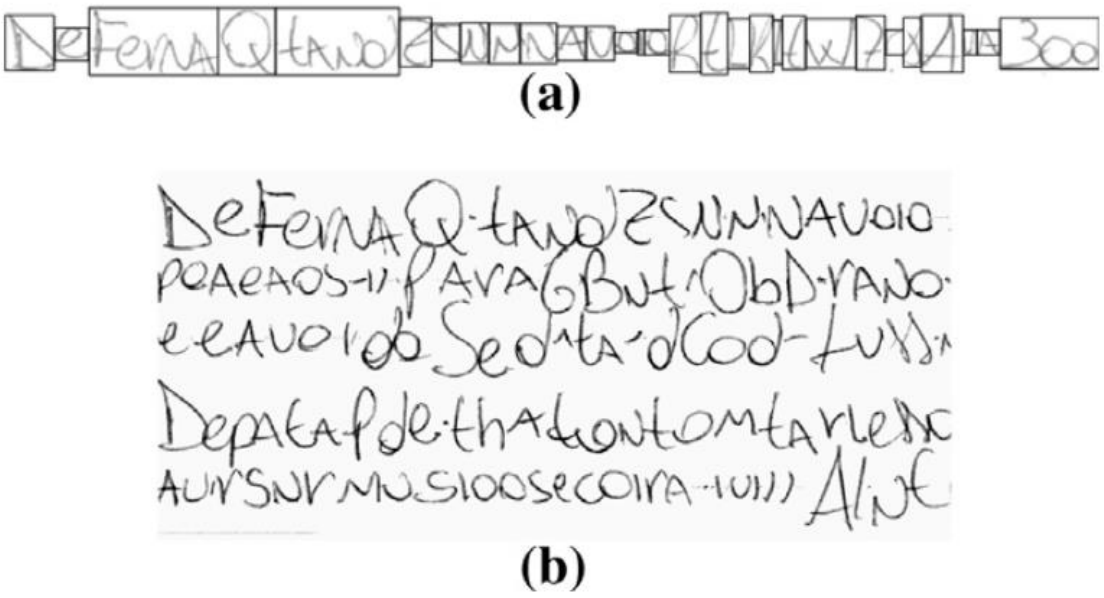


Figure 2.1: Example of text block generation in (Bertolini et al. (2013)). (a) Filling the line. (b) Construction of a normalized texture block.

Nicolaou et al. (2015) used a new variant of the LBP descriptor, known as Sparse Radial Sampling LBP, for writer identification. SR-LBP feature vectors are computed from handwriting images and forwarded to the Nearest Neighbor classifier to identify query writers. The proposed method showed SOTA performance on CVL and ICDAR2013 databases. Hannad et al. (2016) presented a texture-based approach for writer identification of handwritten documents. As shown in Figure 2.2, the authors segmented the handwriting into small fragment sub-images and computed a feature vector for each fragment using LBP, LPQ, and LTP texture descriptors. The query documents are classified based on the extracted feature vectors using the nearest neighbor classifier (1-NN) with Hamming distance metric. They achieved correct performance on IFN/ENIT and IAM databases.

Similarly, Singh et al. (2018) conducted a comparative study of six textural descriptors to identify writers. These include LBP, LPQ, Discrete Wavelet Transform -based Local Extrema Pattern (DWT + LEP), Discrete Wavelet Transform -based Directional and Local Extrema Pattern (DWT + DLEP), Center Symmetric Local Binary Co-occurrence Pattern (CSLBCoP), and Local Tri-Directional Pattern (LTriDP) methods. The authors extracted a set of nine texture blocks for each handwritten document using the algorithm presented in (Hanusiak et al. (2012)). It aligns the text and reduces the spacing between characters, words, and lines of text. An example of extracted texture blocks is shown in Figure 2.3. Four different classifiers, including k- Nearest Neighbor (kNN), Support Vector Machine (SVM), Multilayer Perceptron (MLP), and Random Forest (RF), are used to compare and match texture features within the extracted texture blocks to identify unseen documents from the test set. Singh et al. (2018) have experimentally shown that LBP and LPQ methods perform best on three handwriting databases (Arabic, English, and Devnagri

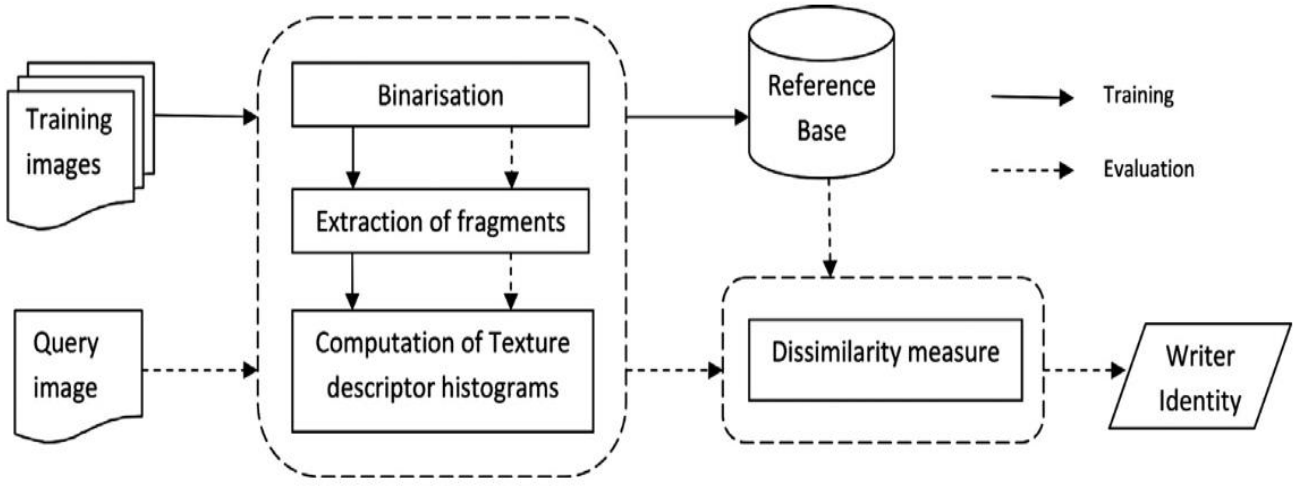


Figure 2.2: Overview of the proposed system in (Hannad et al. (2016))

scripts).

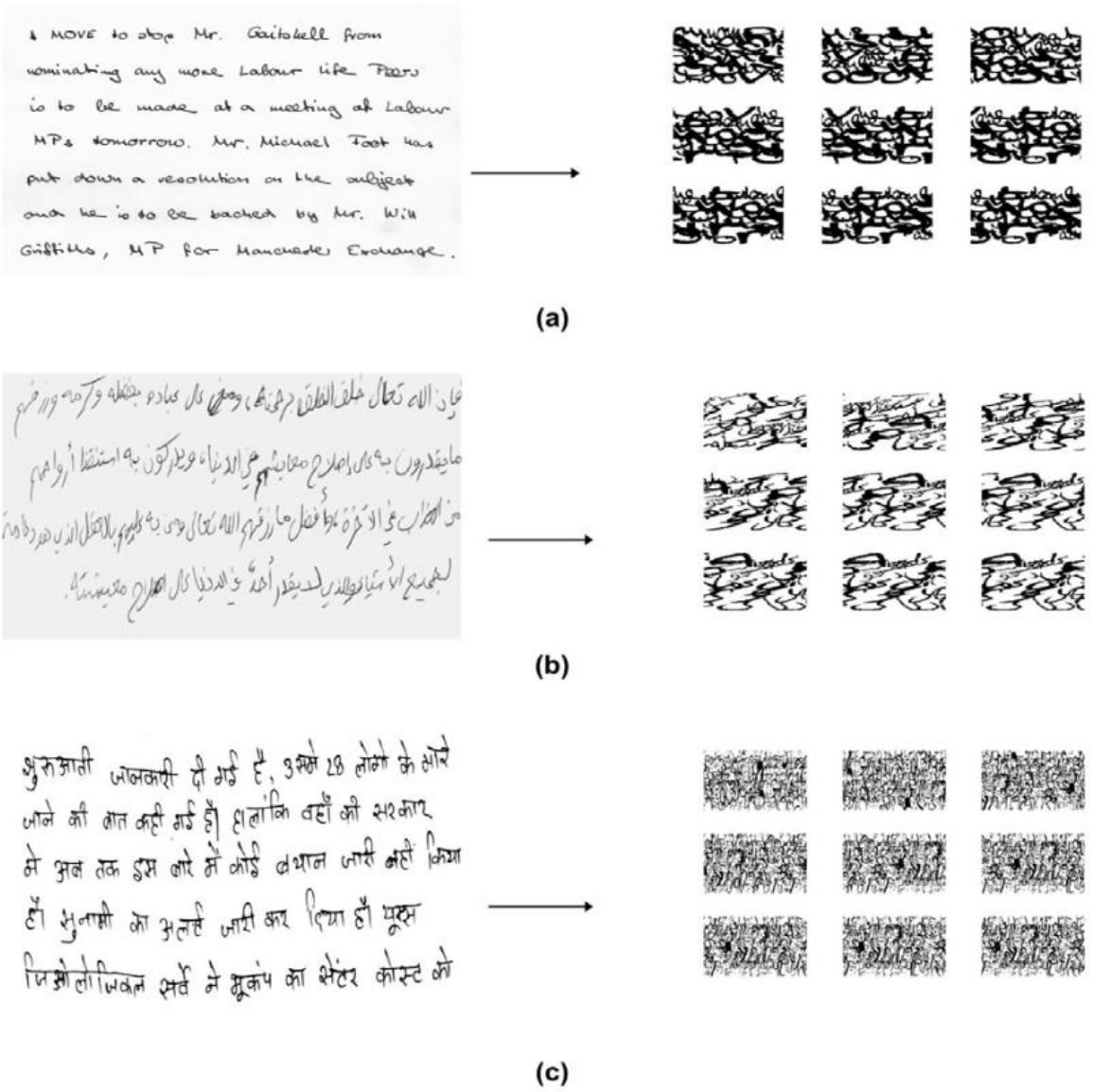


Figure 2.3: Example of input samples with the corresponding texture blocks for (a) IAM (b) KHATT (c) Devnagri databases (Singh et al. (2018)).

Run-length histograms are a trendy feature method primarily because they are conceptually simple and effective in characterizing handwriting variability. In (Chawki and Labiba (2010)), the gray-level run-length matrix (GLRL) and the gray-level co-occurrence matrix (GLCM) are used to capture the textural information of the writing. The proposed system uses the standard nearest neighbor rule with the Euclidean distance metric to perform writer classification. An identification rate of 82.62% is reported for 130 different writers. Later, Gordo et al. (2013) proposed multiscale run-length histograms for large-scale document image identification. They successfully applied Principal Component Analysis (PCA) for dimensionality reduction while maintaining (or even improving) their

discriminative qualities. They reported state-of-the-art results on document classification tasks. Djeddi et al. (2013) investigated text-independent writer identification and verification in various scripts. They evaluated some SOTA features and found that run-length histograms performed much better than other methods in the multi-script environment. They showed encouraging results in two different languages, Greek and English.

He and Schomaker (2016) presented an original texture-based approach for writer identification. They introduced a feature method called General Pattern Run-Length Transform (GPRLT) based on the run-length patterns of handwriting. It can be performed on binary handwriting images (GPRLT bin) or grayscale images (GPRLT gray) without using any binarization or segmentation techniques. Figure 2.4 shows an example of the run length (computed from the binary image of a handwritten character) with three scanning lines and the corresponding converted string lines of the three patterns $(1, 0, 1), (0, 1, 1)$, and $(1, 1, 1)$. Using the GPRLT (General Pattern Run-Length Transform) method, handwritten documents are mapped into feature spaces containing the required properties of the writing style. Then the normalized features are fed to the chi-square-based classifier to perform the writer identification. Experimental results on the English CERUG-EN database validate the effectiveness of the overall system in identifying query writers.

Later in (He and Schomaker (2017b)), two novel curvature-free features are proposed: the run lengths of Local Binary Pattern (LBPruns), which is the run-length histogram of local binary patterns (LBP), and the cloud of line distribution (COLD), which is the distribution of line segments from contours of handwritten texts in polar coordinate space. The

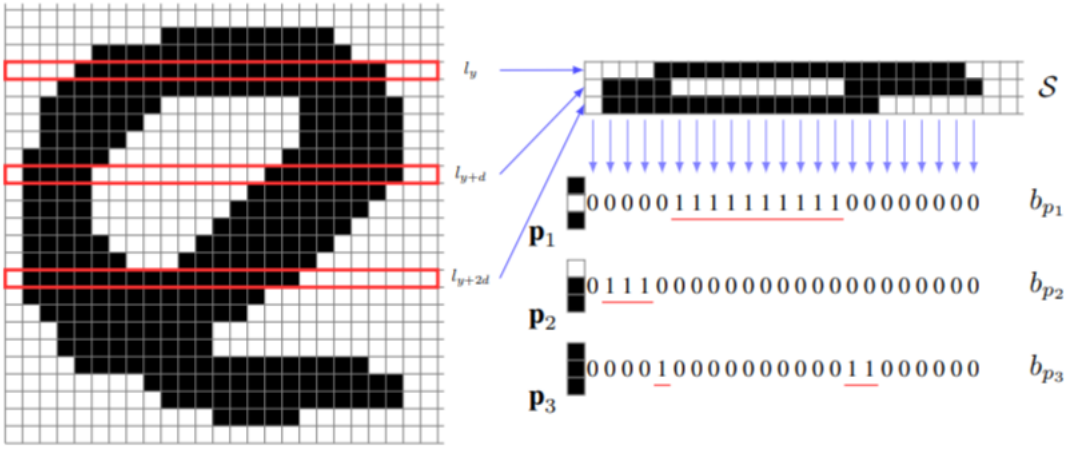


Figure 2.4: The run-length of the more complex patterns p_1, p_2, and p_3 on the scanning line S formed by the three lines l_y, l_{y+d}, l_{y+2d} with distance d. The run length of the pattern p in the scanning line S is computed by the run length of the value **"1"** in the converted string line $b_p(x)$ (x is the index of the sequence)(He and Schomaker (2016)).

overall method achieved good performance on writer identification in three benchmark datasets.

Texture filter-based methods were also studied and evaluated for writer identification. The work in (Said et al. (2000)) was the first to introduce a texture filter-based method to study writer identification in a text-independent mode using 1000 test images written by 40 writers. The overall approach normalizes the input handwriting and generates standard blocks according to the following steps: 1) horizontal projection for line detection, 2) standardization of spacing between lines and words, 3) ”padding” of the text, and 4) block segmentation. Figure 2.5 shows an example of document normalization. Said et al. (2000) computed texture features captured from the preprocessed blocks using the grayscale co-occurrence matrix (GSCM) and multichannel Gabor filtering (MGF). They used the weighted Euclidean distance (WED) and nearest neighbor method (K-NN) to classify the writers. The best results are obtained with an identification rate of 96%.

In another work presented by Shahabi and Rahmati (2006), four different texture filter-based methods for characterizing writer variability were evaluated. These include symmetric Gabor filters, the Gabo sigmoidal transform, the Gabor energy feature, the Fourier

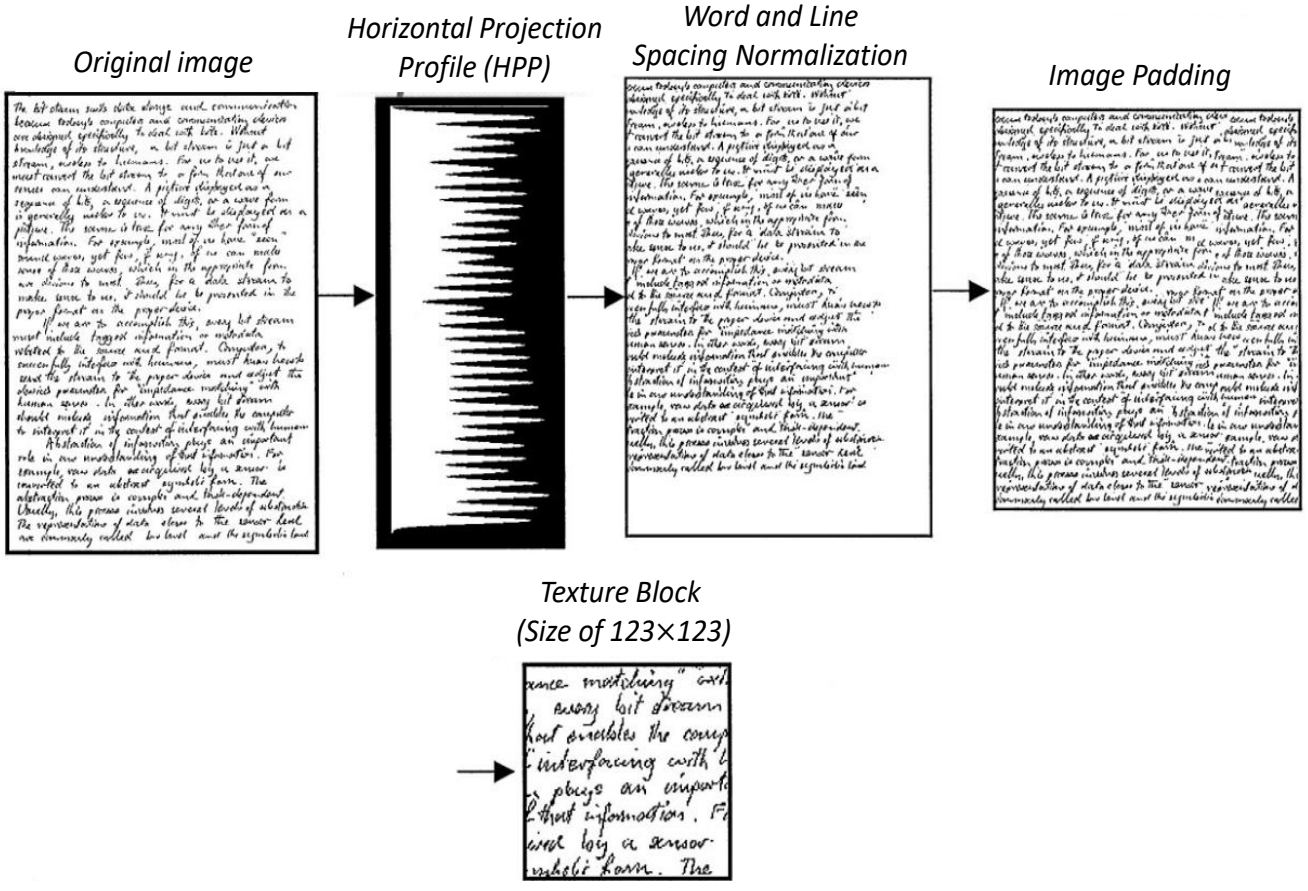


Figure 2.5: Example of handwritten document normalization (Said et al. (2000)).

transform of the Gabor output, and the co-occurrence matrix features. The weighted Euclidean distance (WED) is used to identify unseen writers. Shahabi and Rahmati (2006) reported that Gabor energy and Fourier transform of Gabor output performed better than other methods. Later, Nejad and Rahmati (2007) proposed a system for off-line writer identification of Farsi handwriting. A bank of Gabor filters is applied to images of Farsi handwriting to extract texture features, which are fed to weighted Euclidean and Chi-square classifiers to perform writer identification. Nejad and Rahmati (2007) achieved correct performance on Farsi handwriting of 40 writers.

He et al. (2010) combined Gabor filter, wavelet decomposition, and fractal dimension to improve the performance of writer identification. They generated multiple Gabor sub-bands from the handwriting images, which were expanded into data sequences and decomposed into a series of wavelet subpatterns by wavelet transformation. They used normalized Euclidean distance to perform writer identification. Their method outperforms existing Gabor-based methods and reports SOTA results. Helli and Moghaddam (2010) introduced a feature relation graph (FRG) to encode the directions of Gabor and XGabor filters based on some fuzzy variables. The Gabor filter is mainly used to quantify and characterize frequent patterns. It responds to single lines and depends only on the gradient of the line. Figure 2.6 shows an example of this response. XGabor filter is another variant of the Gabor filter that responds to the writing curves. A 2D convolution operation is used to obtain the response of an XGabor filter to an image. Figure 2.7 shows the results of applying a circular and an elliptical XGabor filter to an image. The authors used a graph similarity measure in the classification phase to identify query writers. Experimental results showed better SOTA performance on 100 writers.

The oriented Basic Image Feature (oBIF) column is an effective texture-based method that has been used for character recognition (Newell and Griffin (2011)) and texture

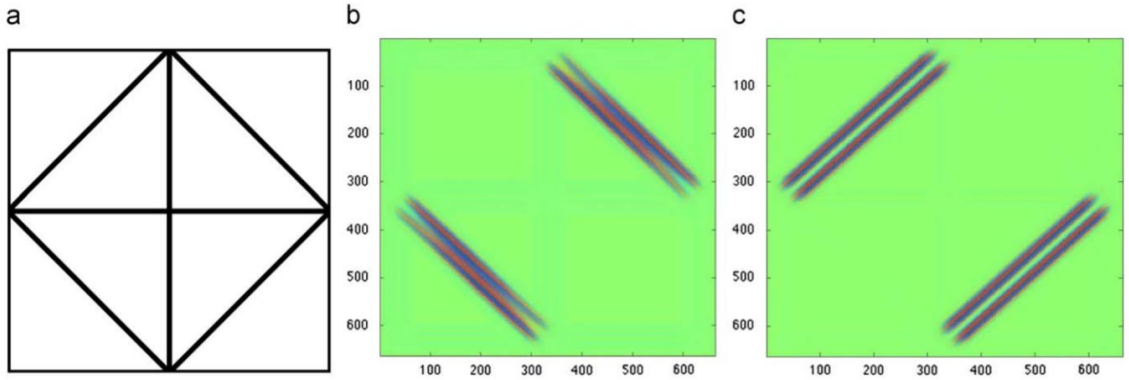


Figure 2.6: (a) A sample image, (b) response of (a) to a Gabor filter with $\theta = \pi/4$ and (c) response of (a) to a Gabor filter with $\theta = 3\pi/4$ (Helli and Moghaddam (2010)).

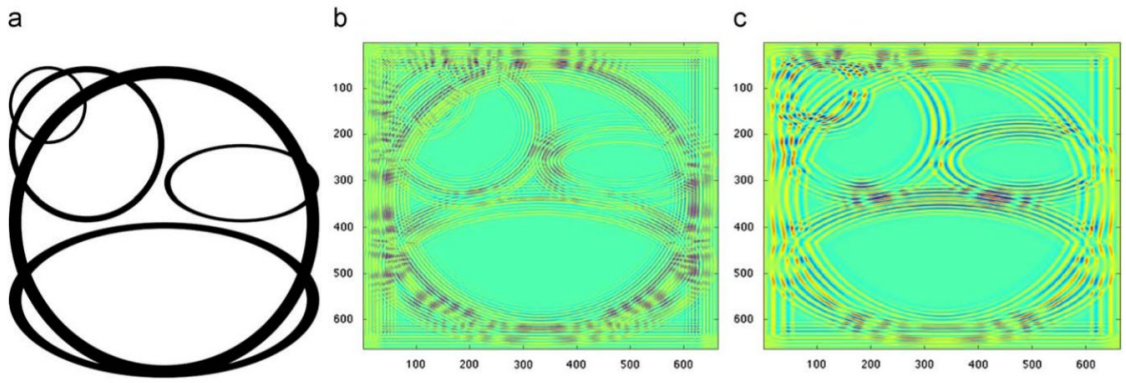


Figure 2.7: (a) Example image consists of 9 different elliptic shapes, (b) result of convolving an elliptic XGabor with (a) and (c) result of convolving a circular XGabor with (a) (Helli and Moghaddam (2010)).

recognition (Timofte and Van Gool (2012); Newell et al. (2010)). It was first adapted for writer identification by Newell and Griffin (2014) and used by the winning team of the ICDAR2017 Historical-Writer identification competition (Fiel et al. (2017); Abdeljalil et al. (2018)). Newell and Griffin (2014) showed that the oriented Basic Image Feature could achieve better writer identification performance even when there are no common handwritten characters between training and test data. As shown in Figure 2.8, the oBIF-based system in (Newell and Griffin (2014)) encodes the handwriting image into oriented Basic Image Features (oBIFs) at two different scales. The oBIFs column features in the encoded image are computed at each location by combining the oBIF type found at the two scales. Note that there are seven possible symmetry types: slope, dark line, light line, dark rotation, light rotation, saddle-like, and flat. The oBIF column features are then counted across the image to form the final normalized feature histogram. The Nearest Neighbour classifier was used to evaluate the oBIF column method using the IAM database. The classification results demonstrated the ability of the oBIF column scheme in characterizing the writing variability with a writer identification score of 99% (tested on 300 writers). Similarly, Abdeljalil et al. (2018) used different configurations of oBIF columns to extract texture information from handwriting. Feature matching and classification were performed using standard distance metrics, including Euclidean, city block, correlation, cosine, and Spearman. Experiments were performed on 720 different writers from the ICDAR2017 database. The approach provides better SOTA performance for writer identification.

Kumar and Sharma (2019) presented a modeling approach based on descriptive distribution curves (DDC-) and cellular automata (CA-). The DDC algorithm uses the pixel distribution of handwritten text images to generate a unique curve as a feature vector fed to a support vector machine (SVM) for writer identification. They also evaluated similarity-

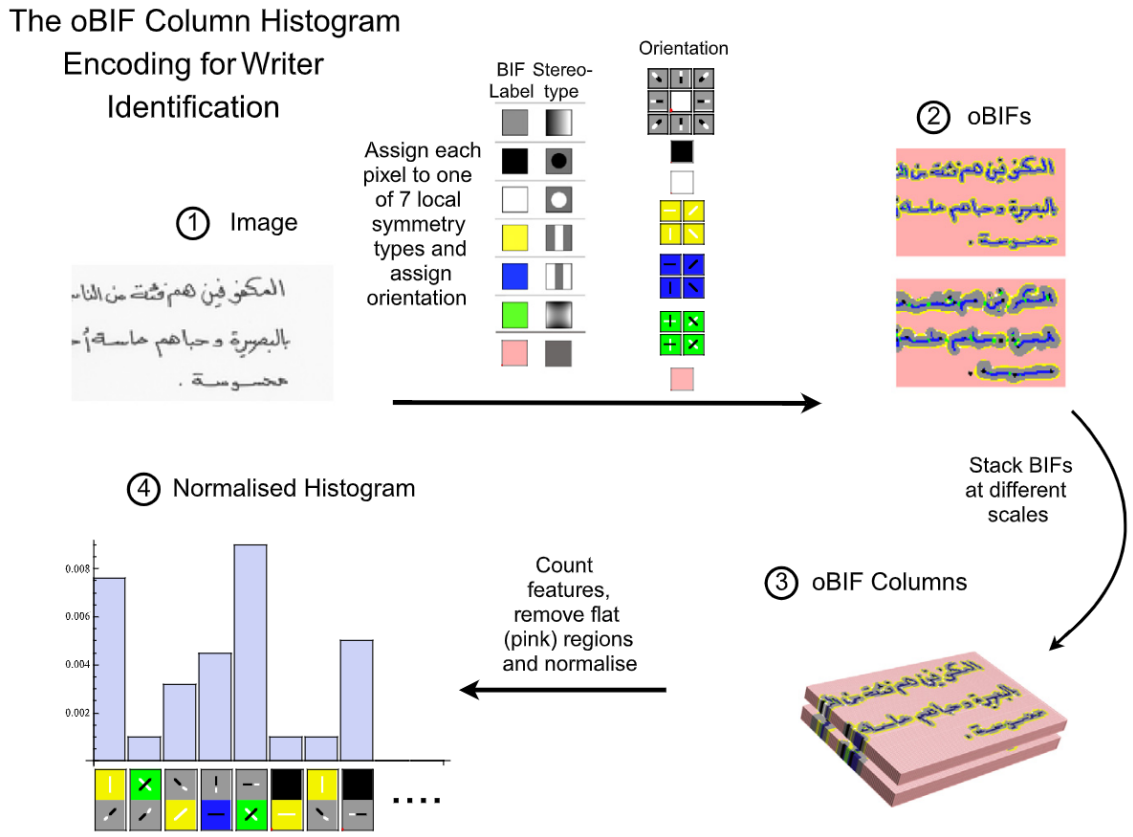


Figure 2.8: The various steps in the oBIF column encoding scheme as applied to writer identification. An image is first encoded into oBIFs, with a local symmetry type and orientation assigned to each location in the scale space. The oBIFs at two scales are then stacked to form a pair or column of oBIFs at each position. To remove whitespace, any column feature that contains a flat oBIF at either scale is discarded. The remaining column features are counted to form a histogram, which is normalised by dividing by the total number of non-flat columns. (Newell and Griffin (2014)).

based classifiers (SBC) for classification. Both approaches (DDC + SVM and CA + SBC) are combined into one system called DCWI to improve the overall writer identification performance.

2.2.2/ GRAPHEME-BASED METHODS

Grapheme-based methods focus on extracting features within small writing traces called graphemes. A robust segmentation algorithm is needed to crop the handwriting into trace lines, and then the segmented lines are further fragmented into small segments (small strokes of handwriting). Every segment might contain zero, one, or more than one grapheme. Encoding techniques like the bag of words are used to locally encode the writ-

ing graphemes over a prototype codebook produced by clustering. Bensefia et al. (2002) first used graphemes for writer identification. They proposed a morphological grapheme-based analysis to characterize each pattern's redundancy, known as writer invariants (cf. Figure 2.9). Two experiments were conducted to evaluate the effectiveness of their method. Based on 88 writers, Bensefia et al. (2002) reported a 97.7% hit rate for writer identification using large compressed handwriting samples. In the second experiment, they investigated the effect of the number of graphemes on overall performance. The score of 92.9% in identification rate is reached for 88 writers using only 50 graphemes of each handwriting sample. The same authors further improved the same database results (88 writers) by using graphemes with an information retrieval paradigm to compare each unseen sample to the reference base (Bensefia et al. (2003)). They also evaluated a set of concatenated graphemes (bi- and tri-gramme) as features to improve the task.

Van Der Maaten and Postma (2005) compared and tested Kohonen-trained grapheme codebooks with grapheme codebooks constructed by random selection to improve writer identification performance. They selected a random number of graphemes from training to form the feature codebook. The authors showed in experiments that random grapheme codebooks performed better compared to Kohonen-based codebooks. Bulacu and Schomaker (2005) compared and analyzed three different clustering methods for grapheme codebook generation. These include k-means, Kohonen self-organizing map (SOM) 1D and 2D (cf. Figure 2.10). Extensive experiments were conducted on both the Firemaker and ImUnipen datasets to compare the three clustering methods over a wide range of codebook sizes. The results for writer identification are consistent with those reported in their previous work in (Schomaker and Bulacu (2004)).

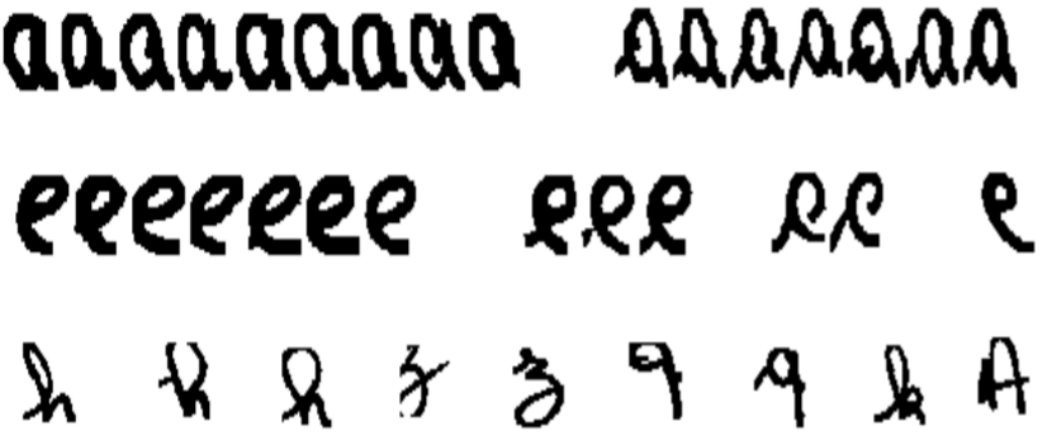


Figure 2.9: Example of invariant clusters (graphemes) extracted from a handwritten page (graphemes) (Bensefia et al. (2002)).

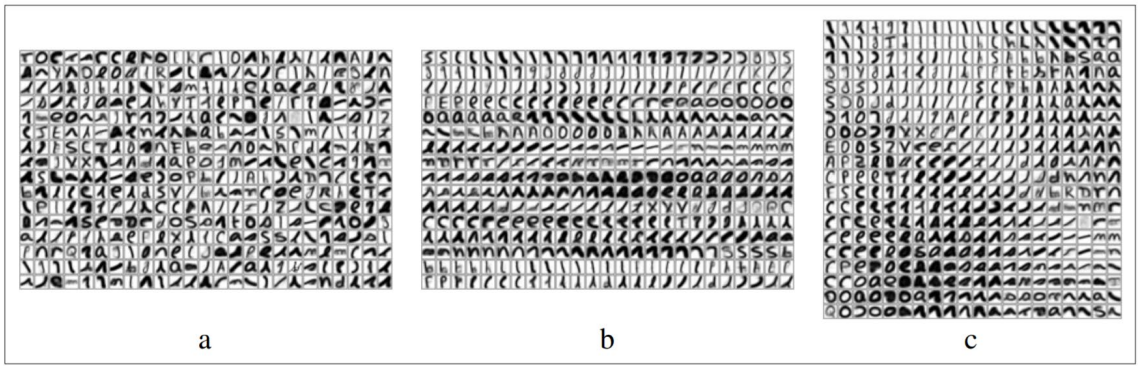


Figure 2.10: Examples of codebooks with 400 graphemes. For K-means (a) and K-SOM 1D (b), the graphemes were arranged 25 in a row, while for K-SOM 2D (c) the original 20x20 SOM organization was retained (Bulacu and Schomaker (2005)).

Bensefia et al. (2005) used a cursive handwriting segmentation algorithm to extract graphemes as local features for writer identification. PSI (88 writers) and IAM (150 writers) databases were used to evaluate their overall system. The Vector Space Model (VSM) was used as a classifier to compare the feature vectors and then identify the unseen documents. Figure 2.11 illustrates an overview of the proposed system. They successfully recorded a score of 96.41% on the PSI database and 97.33% on the IAM database.

Bulacu and Schomaker (2006) combined several features (directional, grapheme, and run-length probability distribution functions) extracted from handwriting images. Otsu's algorithm was used to binarize the input grayscale images, considering three main representations of the document for feature computation: the binary image, the connected components, and their extracted contours. Bulacu and Schomaker (2006) have shown that the feature fusion scheme improves writer identification and verification performance. Gaceb et al. (2006) presented an approach to characterizing the writer's style invariants based on Gabor-based directional features and a complete grapheme signature. The method reported good results in writer identification.

In (Bulacu and Schomaker (2007)), original work on writer identification and verification using an effective allographic feature method was presented. They used contour-based joint directional probability distribution functions (PDFs) that highly encode orientation and curvature information to characterize writing style. They also used a stochastic pattern generator of ink-trace fragments or graphemes and computed the probability distribution functions (PDF) of these shapes (in a given handwriting sample) using a typical shape codebook obtained by grapheme clustering. Experimental results showed that combining multiple features (directional, contour-Hinge PDF, grapheme, Autocorrelation, and run-length PDFs) allows high performance in writer identification.

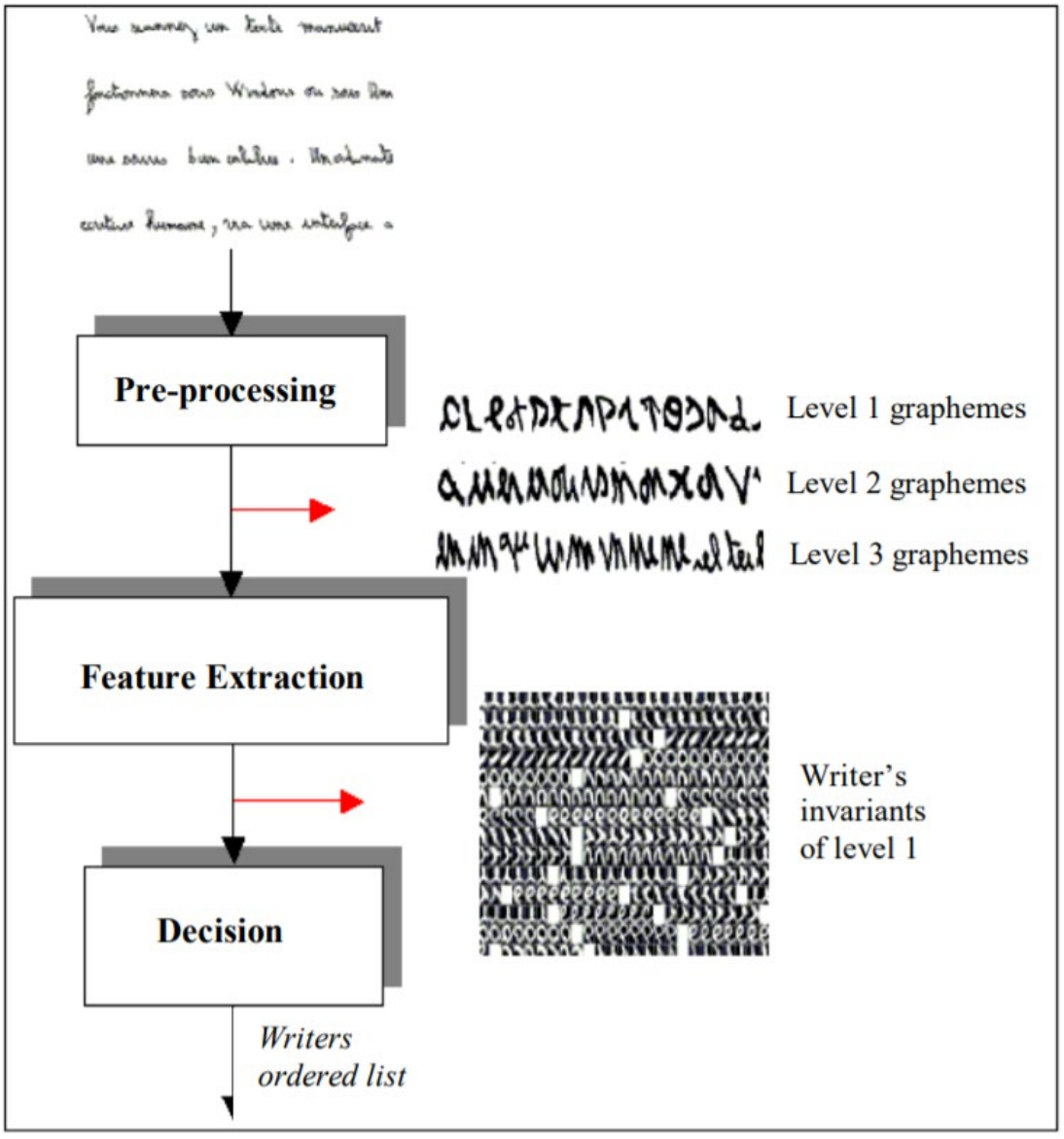


Figure 2.11: An overview of the approach proposed in (Bensefia et al. (2005)).

Pervouchine and Leedham (2007) used microstructural features extracted from characters and graphemes to identify query writers. They used a generic algorithm called wrapper (John et al. (1994)) to search for the optimal features and trained a neural network as a classifier for writer identification. Experiments showed that graphemes are more efficient than micro-features in characterizing writing variability. In (Khalifa et al. (2015)), an ensemble of multiple codebooks was proposed for writer identification. They used spectral regression with kernel discriminant analysis (SR-KDA) as a dimensionality reduction technique to avoid over-fitting. A Nearest-Neighbor (NN) classifier was used in a leave-one-out strategy to evaluate the performance of writer identification. Figure 2.12 shows the foremost steps used in this approach. Experiments on the IAM (Marti and Bunke

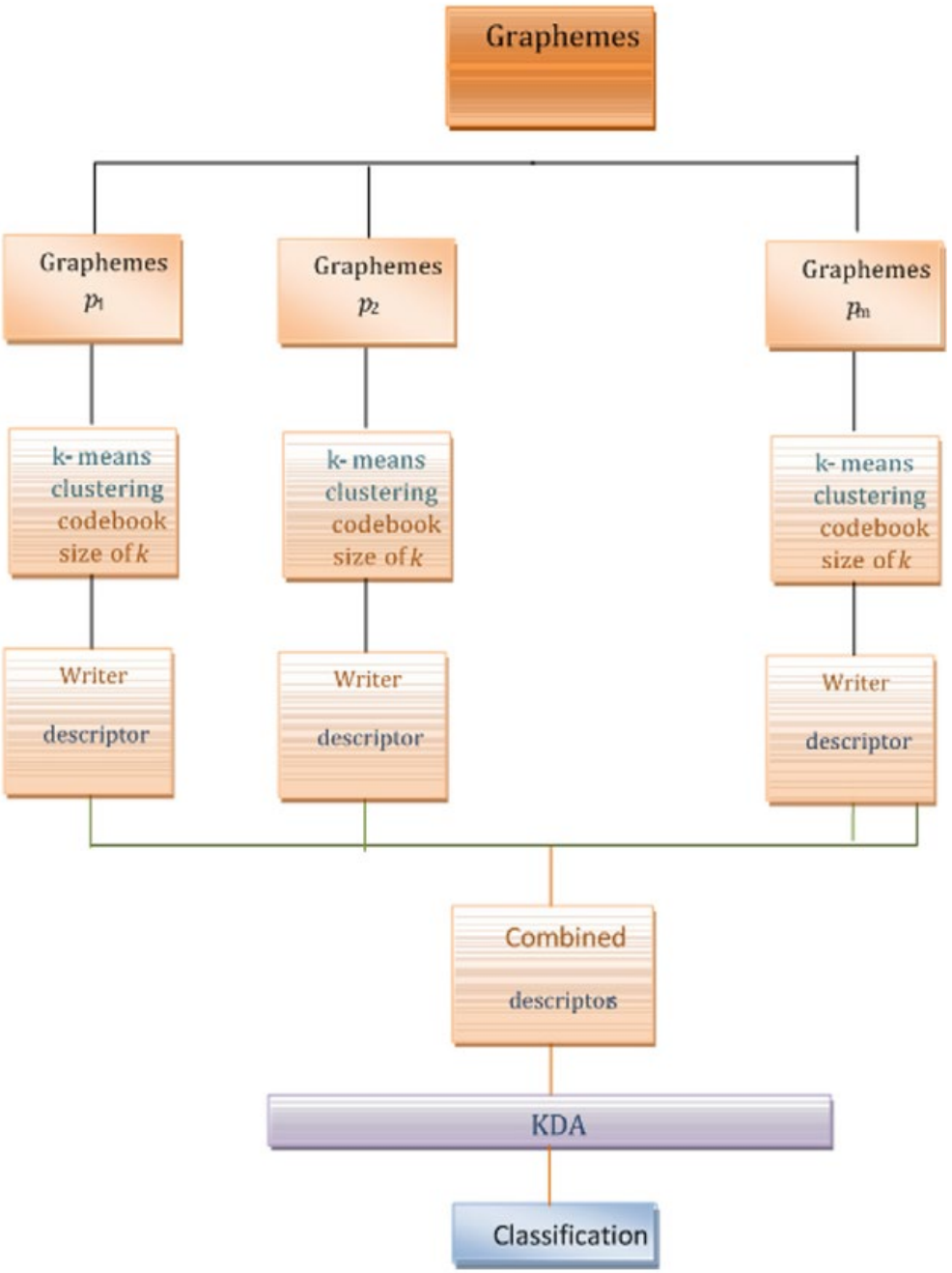


Figure 2.12: The main steps of the system proposed in (Khalifa et al. (2015)).

(2002)) and ICFHR2012 (Hassaïne and Al Maadeed (2012)) datasets showed that the fusion of multiple codebooks gives better performance in writer identification than a single codebook approach.

Abdi and Khemakhem (2015) generated synthetic codebooks for feature extraction using the beta-elliptic model. Their work reported that they were the first to address the capability of model-based synthetic codebooks in writer identification and verification. Instead

of extracting natural graphemes from a training set using segmentation and clustering, their approach synthesizes their own graphemes based on the beta-elliptic model. The grapheme-based approach generates one full and four partial codebooks, which are reduced according to specific criteria using a feature selection procedure. Experiments are performed on the IFN/ENIT database (Pechwitz et al. (2002)) with a total of 411 writers using 60 feature vectors extracted by template matching. The results show a good generalization ability of the synthetic codebooks for writer identification.

Garz et al. (2016) proposed a conceptually fast and straightforward approach for writer identification. It captures orientation distributions at multiple scales and geometric relationships between grapheme strokes, junctions, endings, and loops. Experimental results on the IAM database showed the effectiveness of combining these methods, with a result of 86.9% in the top-1 writer identification rate. In (Miller et al. (2017)), isomorphic graph class and shape are embedded in a generic graph-based system to improve automated handwriting identification. Miller et al. (2017) used topological and geometric classification of graphemes with ”like-with-like” comparisons of similar features across different writers.

Khan et al. (2017) used universal codebooks with bagged discrete cosine transformed (BDCT) descriptors to identify writers from handwriting. DCT features are computed from overlapping blocks (graphemes) extracted from the original writing. Dimensionality reduction of the extracted features was performed using Kernel Discriminant Analysis with Spectral Regression (SR -KDA), and classification was performed using the nearest-center rule. As illustrated in Figure 2.13, multiple SR -KDA predictor models are generated for each writer (codebook) using a training set. The authors used a majority voting strategy to identify unseen documents in the test set, as shown in Figure 2.14. The overall system achieved superior SOTA performance on four different benchmark databases.

Pandey and Seeja (2018) proposed a grapheme-based approach to text-independent writer identification. Graphemes are extracted from handwriting and represented as projection profile representations. They performed dictionary learning (codebook) from handwriting samples using k-means clustering. The k-nearest neighbor classifier is used to compare and match feature vectors to identify query writers. Pandey and Seeja (2018) reported better results compared to other grapheme-based methods. Durou et al. (2019) proposed a feature fusion approach combining Oriented Basic Image features (oBIF columns) and grapheme codebooks for writer identification. They used Kernel Principal Component Analysis (kPCA) to reduce the high dimension of the resulting feature vector. The K- Nearest Neighbour, Support Vector Machine (SVM), and Neural Networks are used to classify unseen writers from IAM and ICFHR2012 databases. Their method showed better SOTA performance compared to similar techniques.

The Scale-Invariant Feature Transform (SIFT) descriptor is considered an efficient

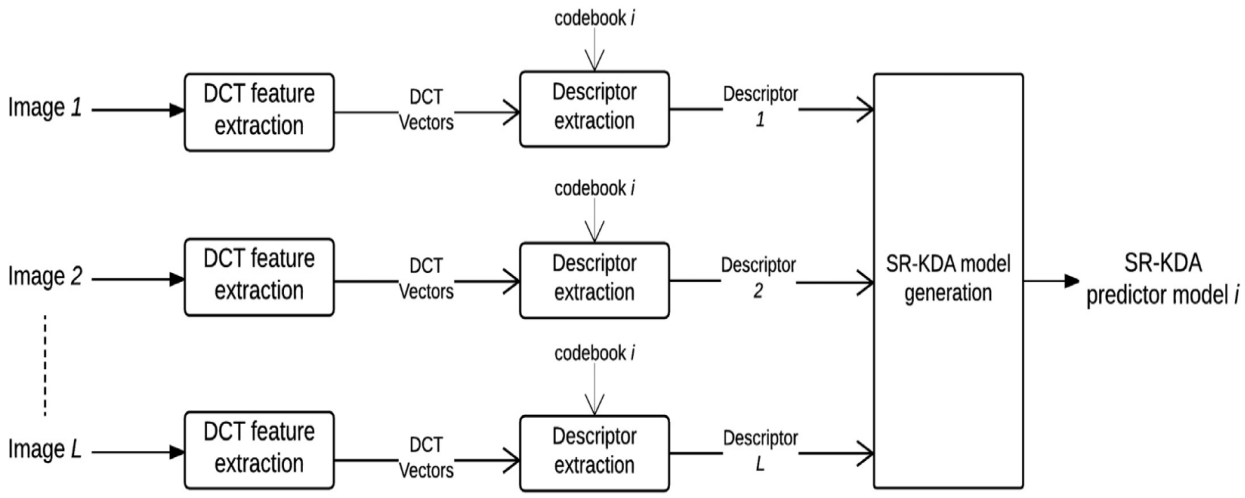


Figure 2.13: Training step - SR-KDA predictor model *i* is generated for codebook *i* (Khan et al. (2017)).

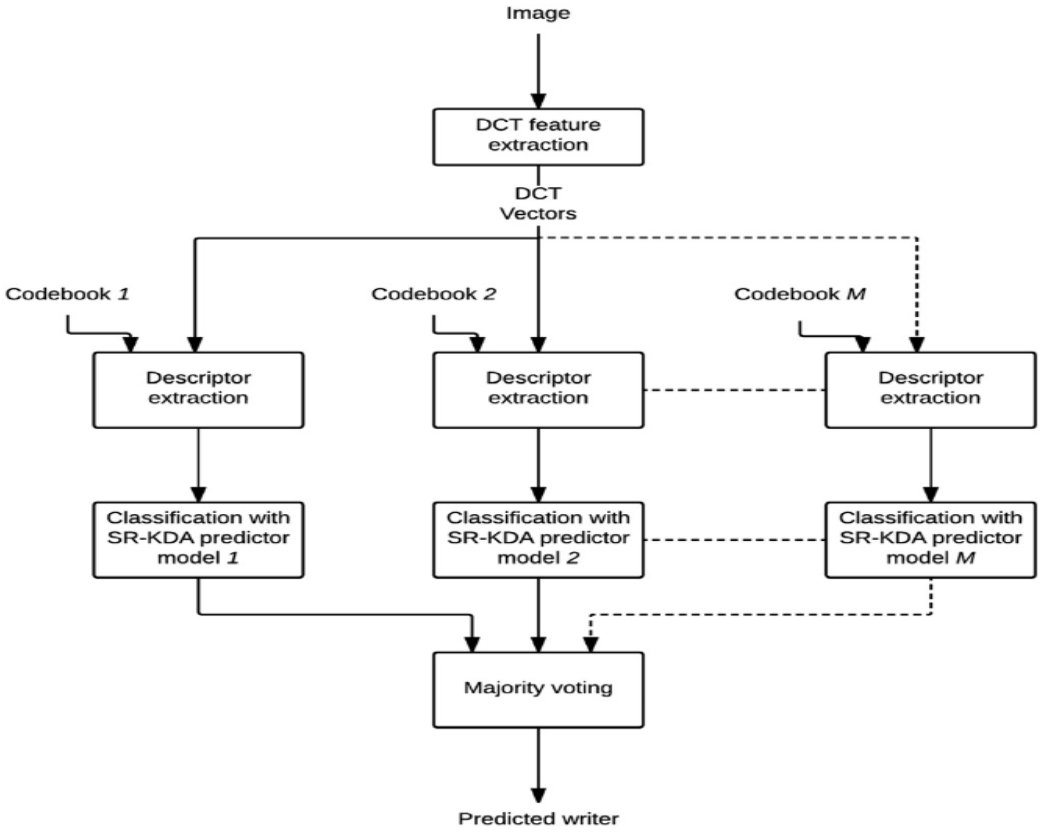


Figure 2.14: Testing step of the system proposed in (Khan et al. (2017)).

grapheme-based feature method for all pattern recognition problems, especially for characterizing salient local structures. In writer identification and verification, the SIFT descriptor has been widely used to capture local structures of writing variability. It can be extracted locally for feature matching, as in (Mohammed et al. (2017, 2018)), or encoded into a global vector or score, as in (Fiel and Sablatnig (2013); Christlein et al. (2017a); Khan et al. (2019)). The most recent typical work using this type of feature is the approach presented in (Lai et al. (2020)). The authors combined two feature methods to capture salient information of local structures of handwriting: they proposed a novel contour-based method called Pathlet feature and used a grapheme-based method known as the unidirectional SIFT feature to describe corners and junctions of handwriting. Also, an encoding method called bagged vector of locally aggregated descriptors (bagged-VLAD) was introduced to encode both Pathlet and SIFT features effectively. The overall system was evaluated on historical benchmark databases for writer identification and achieved excellent SOTA performance. An overview of the overall system is shown in Figure 2.15.

2.2.3/ CONTOUR-BASED METHODS

Contour-based methods have also been studied for writer identification. These methods capture features from handwriting contours rather than image pixels, which are a probability distribution of local attributes, such as angles and ink widths. Text image segmentation techniques are required to enhance the identification performance. The most

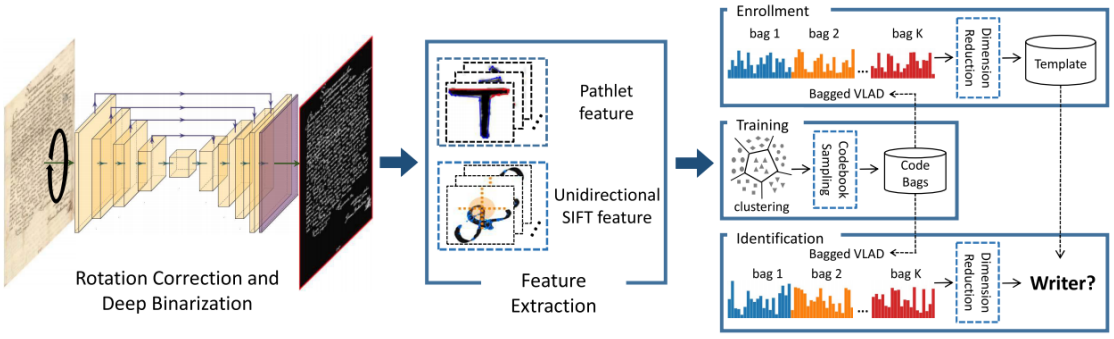


Figure 2.15: The historical writer identification system proposed in (Lai et al. (2020)). First, document images are rotation corrected and binarized using deep U-Net model. Second, pathlet features and unidirectional SIFT features are extracted. Third, for each document, the pathlet features and SIFT features are encoded using the proposed bVLAD, followed by dimensionality reduction and l_2 normalization. The resulting feature vectors are then used to identify the unseen document (Lai et al. (2020)).

well-known work that used this approach is the one presented in (Schomaker and Bulacu (2004)). The authors introduced an important theory-based approach to use connected-component contours (CO3s) and edge-based features for automatic writer identification. A stochastic pattern generator model is used to characterize a family of connected components of the Western uppercase script. Based on the CO3s training codebook of 100 writers, Schomaker and Bulacu (2004) calculated the probability density function (PDF) of the CO3s test set of 150 unseen writers. Their experimental results showed a high sensitivity of the CO3-PDF for identifying individual writers based on a single sentence of capital letters. They also combined the CO3-PDF with an independent edge-based orientation and curvature PDF to improve the performance of writer identification. Similarly, Schomaker et al. (2004) improved the task of writer identification using fragmented connected-component contours (FCO3) in mixed handwritten samples of limited size. In a stochastic model, they generated a family of character fragments called fraglets and determined their probability distribution for an independent test set. Later, Bulacu and Schomaker (2007) captured the joint probability distribution functions (PDF) of orientations of the two legs of the so-called ”contour-hinge.” Bulacu and Schomaker (2007) also used a stochastic pattern generator of ink-trace fragments or graphemes and computed the PDF of these shapes (in a given handwriting sample) using a typical shape codebook obtained by grapheme clustering. The authors experimentally demonstrated that combining multiple features (directional, contour-Hinge PDF, grapheme, and run-length PDFs) leads to high performance in writer identification. In another work presented in (Bulacu et al. (2007)), the same combination of features, i.e., PDFs-based features and graphemes, was used to evaluate writer identification from Arabic handwriting. Figure 2.16 shows an example of extracted direction and run-length PDFs. The overall system achieved an identification rate of 88% on an Arabic database of 350 writers. The results prove the possibility of applying the same approach to other script languages, except that the authors affirmed that Arabic writing presents more difficulty and a real challenge for writer identification systems.

Abdi et al. (2009) presented a contour-based system for writer identification from Arabic handwriting. The approach computes the joint probability distribution functions (PDF) of a combination of contour features. As with other writer identification systems that use PDFs, a preprocessing stage is required for the document image. This step, shown in Figure 2.17, consists of normalizing the word image by removing diacritical marks, dilating the image, and extracting word contours. Relevant word contours are approximated and extracted using a minimum perimeter polygon (MPP) algorithm with different pixel grid sizes. Abdi et al. (2009) used six features based on length, direction, angle and curvature measurements extracted from the relevant contours (cf. Figure 2.18).

An interesting work presented by Siddiqi and Vincent (2010) proposes a contour-based approach for writer identification. It extracts, as shown in Table 2.1, a set of 14 visual

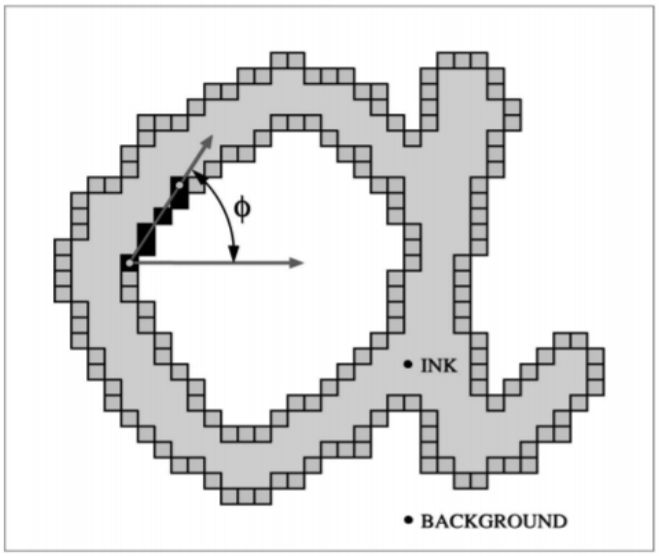


Figure 2.16: Schematic description for the contour-direction PDF feature extraction method. (Bulacu et al. (2007)).

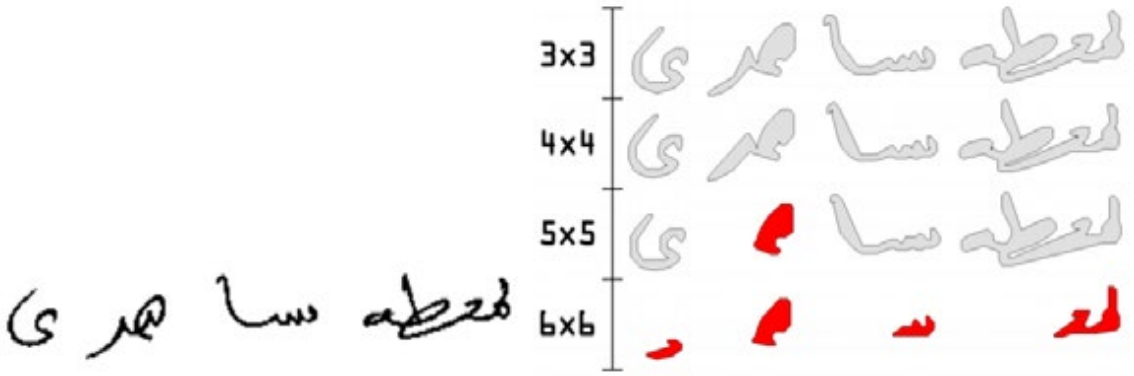


Figure 2.17: An example of relevant word contours obtained using the minimum perimeter polygon (MPP) algorithm with different pixel grid sizes. Contour projections disappear as the grid size increases. MPP contour edges that no longer meaningfully represent their respective word outlines are filled with a darker color (Abdi et al. (2009)).

features (orientation and curvature information) from redundant patterns of the writing. The extracted contour-based features, called chain-code-based probability distribution and polygon (cf. Figure 2.19), provide competitive performance to the Hinge feature proposed in (Bulacu and Schomaker (2007)). The overall system combines codebook

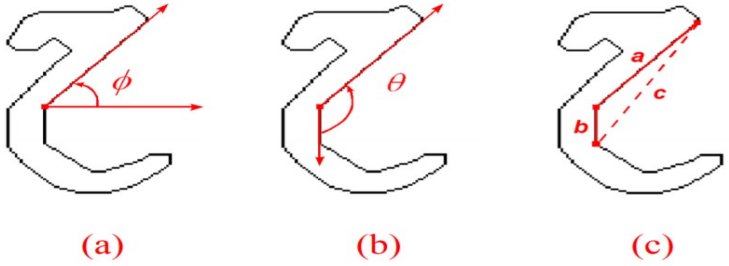


Figure 2.18: (a) Edge direction ϕ relative to the horizontal straight line. (b) Angle θ formed by two adjacent edges. (c) Curvature C (Abdi et al. (2009)).

Table 2.1: Summary of contour features. $f1 - f9$ are chain code-based features and $f10 - f14$ are polygon-based features (Siddiqi and Vincent (2010)).

Feature	Description	Dimension
$f1$	Distribution of chain codes	8
$f2$	Distribution of 1st order differential chain codes	7
$f3$	Distribution of 2nd order differential chain codes	8
$f4$	Distribution of chain code pairs	44
$f5$	Distribution of chain code tripplets	236
$f6$	Distribution of curvature indices	11
$f7$	Local stroke direction distribution	80
$f8$	$f2$ computed locally	70
$f9$	$f3$ computed locally	80
$f10$	Distribution of segment slopes	8
$f11$	Length-weighted distribution of segment slopes	8
$f12$	Distribution of curvatures	8
$f13$	Length-weighted distribution of curvatures	8
$f14$	Distribution of segment lengths	10
Total		586

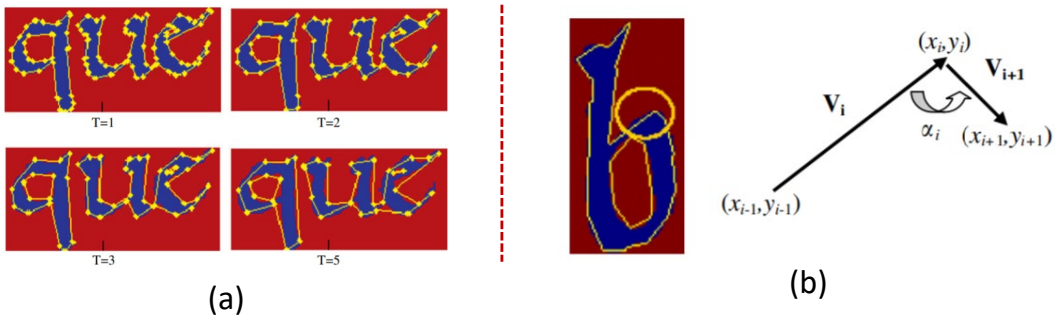


Figure 2.19: (a) Polygonization at different values of T. T is a user-defined parameter that controls the accuracy of the approximation. Larger values of T produce longer segments at the expense of character shape degradation, and vice versa. (b) Curvature (angle) between two connected segments (Siddiqi and Vincent (2010)).

and contour features. The identification step in this system is based on the aggregation of the computed chi-square distances (related to extracted features) to identify the query writers. The system performance has been evaluated on different databases. The highest rate of 91% is reached on 650 writers.

Another work presented by Jain and Doermann (2011) exploits the K-adjacent segments feature (KAS), which is used to represent the relationship between groups of adjacent (neighboring) edges in an image for object detection (Ferrari et al. (2007)). As the name implies, K-adjacent segments describe any number K of adjacent segments, where two segments are considered adjacent if they share a common endpoint. As explained in (Jain and Doermann (2011)), the primary line segment is defined as the line whose midpoint is closest to the center of the midpoints of all lines. The remaining lines are ordered by their midpoints from left to right and then from top to bottom. Figure 2.20-(a) shows an example of KAS feature extraction ($3AS$ with $K = 3$). Before proceeding with the computation of K- AS features, the approach proposed in (Jain and Doermann (2011)) extracts contours and edges of the writing, performing the following preprocessing steps: 1) binarization of the document image, 2) extraction of contours capturing the shape and curvature, 3) decomposition of smooth curves into a set of lines using a line fitting algorithm. This process is illustrated in Figure 2.20-(b). Then, a feature vector is extracted from the codebook K- AS computed for each writer and compared using Euclidean distance to identify the unknown author of the test document. Classification results are comparable to SOTA systems, with an identification rate of 93% for 350 writers.

Brink et al. (2012b) reported an interesting work on writer identification using directional ink trace width measurements. The authors developed efficient pixel contour-based feature methods called Quill and QuillHinge (a variant of Quill). The proposed approach captures the information of probability distribution between ink direction and ink width.

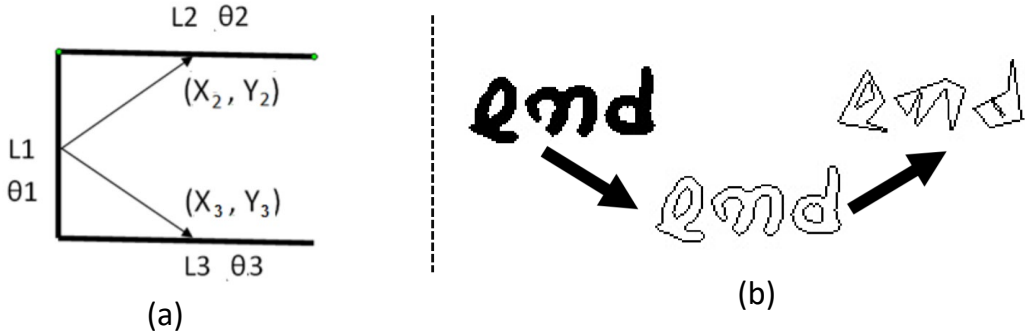


Figure 2.20: (a) The segment ordering and recorded features for a 3AS, where the primary segment is numbered 1. θ and L are the orientation and length of a particular segment that makes up the feature KAS. (b) Extraction of contours and edges (Jain and Doermann (2011)).

The Quill feature calculation (contour tracking, angle measurements, and width measurements) is shown in Figure 2.21. In the classification process, the nearest-neighbor rule was used to perform writer identification. The overall system was evaluated on two datasets of medieval handwriting, the Dutch Charter dataset and the diverse English dataset, and on two datasets of contemporary handwriting, the Firemaker and IAM databases. The experimental results confirm the effectiveness of the QuillHinge-based system for identifying writers.

Ghiasi and Safabakhsh (2013) presented two different codebook-based methods for extracting codes from COnnected-COmponent COntours (CO3), which are a significant improvement over early CO3 (Schomaker and Bulacu (2004)) and FCO3 (Schomaker et al. (2004)). The first method uses the pixel coordinates of contour fragments, while the second is based on a linear piecewise approximation using segment angles and lengths. Both methods use the frequency histogram of component shapes in a codebook to create a feature vector for each handwriting sample. Evaluations on two English and three Farsi handwriting databases showed promising performance in identifying unseen writers.

The approach presented in (Awaida and Mahmoud (2013)) captures gradient and contour chain code features from handwriting samples. Five feature selection methods were used for data reduction. These include principal component analysis (PCA), linear discriminant analysis (LDA), multiple discriminant analysis (MDA), multidimensional scaling (MDS), and forward-backward feature selection algorithms. The nearest neighbour (NN) method was used to classify the query writers (from the test set). The authors evaluated different distance metrics and reported that Euclidean distance performed better with a top 1 result of 75.0% for 250 writers (using 54 out of 83 features with the backward fea-

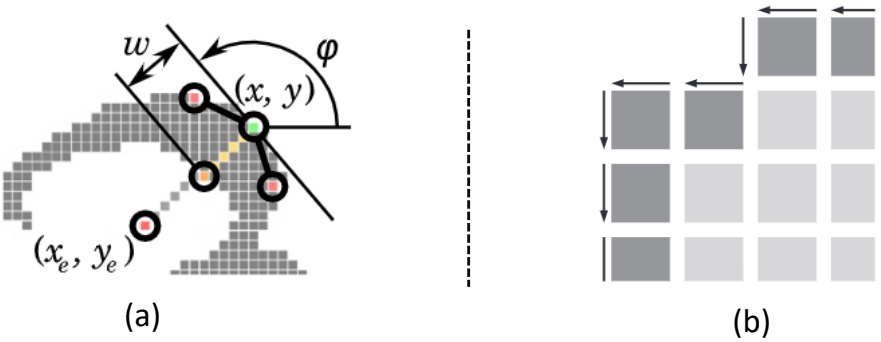


Figure 2.21: (a) ϕ and w are determined at each contour pixel (x, y). ϕ (trace direction) is measured by averaging the angles with two adjacent contour pixels at distance r. w (trace width) is computed using the so-called Bresenham width: the distance to the first background pixel hit when following a Bresenham path, perpendicular to ϕ, in the direction (x_e, y_e). (b) Contour tracing by tracking crack edge contours, shown as arrows. Foreground pixels are shown as blocks; pixels in the resulting trajectory are shaded dark (Brink et al. (2012b)).

ture selection algorithm). Jain and Doermann (2014) presented a contour-based system to address the writer identification problem, which is an improvement over the early system in (Jain and Doermann (2011)). In feature extraction, they used the combinations of three different feature methods: K- Adjacent Segments (K- AS), Speeded Up Robust Features (SURF), and Contour-Gradient Descriptors (CGD) (cf. Figure 2.22). A linear combination of the Fisher Vector distances is used (feature pooling), and feature matching is performed using the Gaussian Mixture Model supervector (GMM). This approach provides SOTA performance on three different datasets. The authors pointed out that the combination of local features consistently performs better than single features.

He and Schomaker (2014) proposed a system for writer identification based on the use of the same feature method presented in (Bulacu and Schomaker (2007)), known as the Hinge feature. They proposed a new variant of the Hinge feature, called Delta-n Hinge, that incorporates the derivative between several writing points along the ink contours to extract rotation-invariant features. The Nearest-Neighbor (NN) classification with a "leave-one-out" strategy was used to perform writer identification. The overall system was evaluated on two different databases, showing promising SOTA performance.

Christlein et al. (2015a) used contour Zernike moments as local features by decomposing the handwriting image using Zernike polynomials. The extracted Zernike moments are encoded into a global descriptor using the Vectors of Locally Aggregated Descriptors (VLAD) algorithm. The feature vectors are decorrelated using Principal Component Analysis (PCA), and dimensionality reduced to 256 components and finally matched and compared using a cosine distance metric. The system outperforms existing methods in two benchmark databases for writer identification. In (He et al. (2015)), a generic approach to junction detection for writer identification is proposed. The system performs a junction detection using Junclets to identify query writers. Later, in (He et al. (2016b)), they designed two methods to characterize handwriting style: local contour fragments (kCF) and stroke fragments (kSF) features. These methods were further explored to

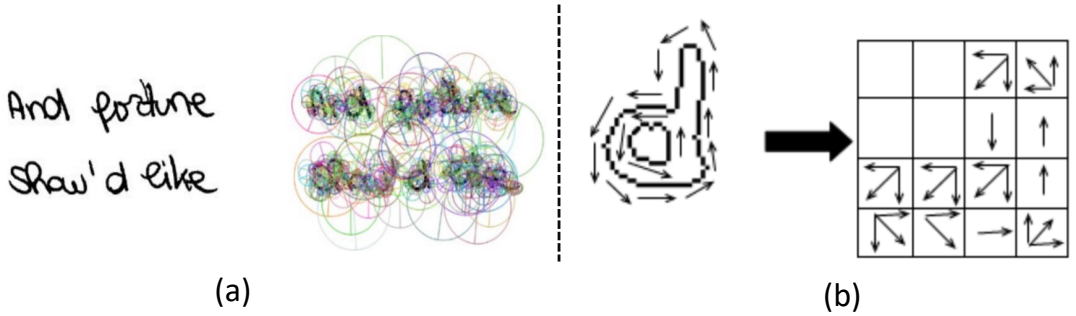


Figure 2.22: (a) SURF Features extracted from handwriting. (b) : Contour gradients and the resulting feature (Jain and Doermann (2014)).

evaluate other applications, such as dating historical documents and multi-faceted tasks, including writer identification (He and Schomaker (2017a)).

He and Schomaker (2017a) proposed the principle of Joint Feature Distribution (JFD) to develop feature algorithms for writer identification, script recognition, and historical manuscript dating and localization. They introduced seventeen features, including twelve texture-based and five contour-based features. For writer identification, the features are matched and compared using the Chi-square distance measure. The best experimental results (on five different databases) were obtained using the QuadHinge and CoHinge pixel contours as features.

2.2.4/ DEEP LEARNING-BASED METHODS

Convolutional neural networks (CNNs) have recently emerged as state-of-the-art tools for large-scale image classification and pattern recognition problems. With enough training data and good optimization, Deep learning can provide an accurate solution for identifying writers based on their handwriting. CNN models were first used as activations of fully connected layers to capture deep features for writer identification (Fiel and Sablatnig (2015); Christlein et al. (2015b, 2017b); Xing and Qiao (2016)).

Fiel and Sablatnig (2015) used Convolutional Neural Networks (CNN) to extract learned features from segmented words and text lines. The CNN model named Caffenet was trained on different databases of known writers, and the penultimate fully connected layer was used for feature activation. Figure 2.23 shows the structure of the Caffenet network. The Nearest Neighbor (NN) classifier was performed to compare and match these deep features to identify unseen writers. The overall system obtained SOTA results in the ICDAR2013, ICDAR2011, and CVL databases with 50, 26, and 309 writers, respectively.

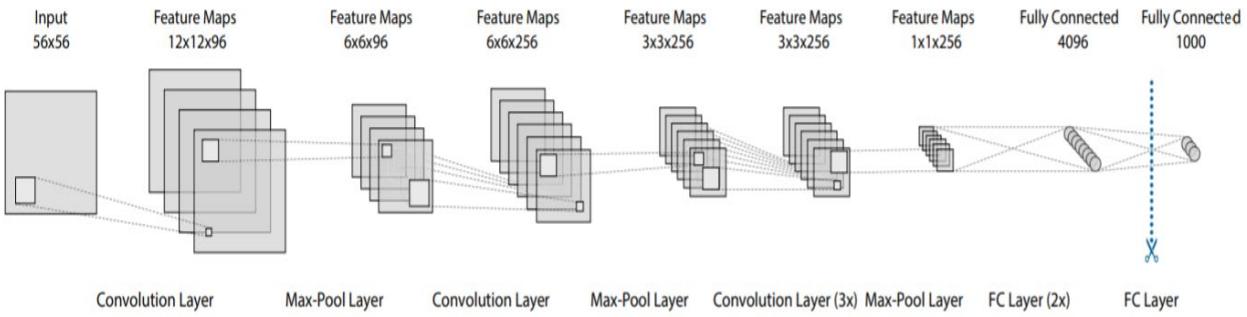


Figure 2.23: Structure of the Caffenet CNN model (Fiel and Sablatnig (2015)).

Christlein et al. (2015b) used activation features learned from a deep CNN model as local descriptors to characterize handwriting style. They used zero-phase component analysis whitening (ZCA) to decorrelate the activation features, followed by global $L2$ normalization. Deep local features were extracted from patches in a handwriting sample and encoded using a modified variant of the Gaussian Mixture Model (GMM) supervector. The resulting GMM supervector was normalized to the global descriptor using the KL kernel (Kullback Leibler) presented in (Xu et al. (2010)). Figure 2.24 illustrates the main steps of the approach proposed in (Christlein et al. (2015b)). The authors compared the performance of CNN features with Speeded Up Robust Features (SURF) and Enhanced Scale-Invariant Feature Transform (RootSIFT) methods using the same classification method, i.e., the Nearest Neighbor classifier with cosine distance as similarity metric. Experimental results on two different databases showed superior SOTA performance.

Xing and Qiao (2016) designed a multi-stream CNN architecture called DeepWriter, which consists of two branches sharing the same CNN layers for writer identification. Local handwritten patches of size 113×113 (cf. Figure 2.25) are the input of the DeepWriter model, which was directly trained (or pre-trained) on two benchmark databases with Softmax classification loss. A patch-scan strategy is introduced to handle text images with different lengths. Since two adjacent CNN streams construct the DeepWriter model, the two output vectors of the fully connected layer (FC7) are merged by an element-wise sum operation. Figure 2.26 shows the construction of the DeepWriter model. In the test

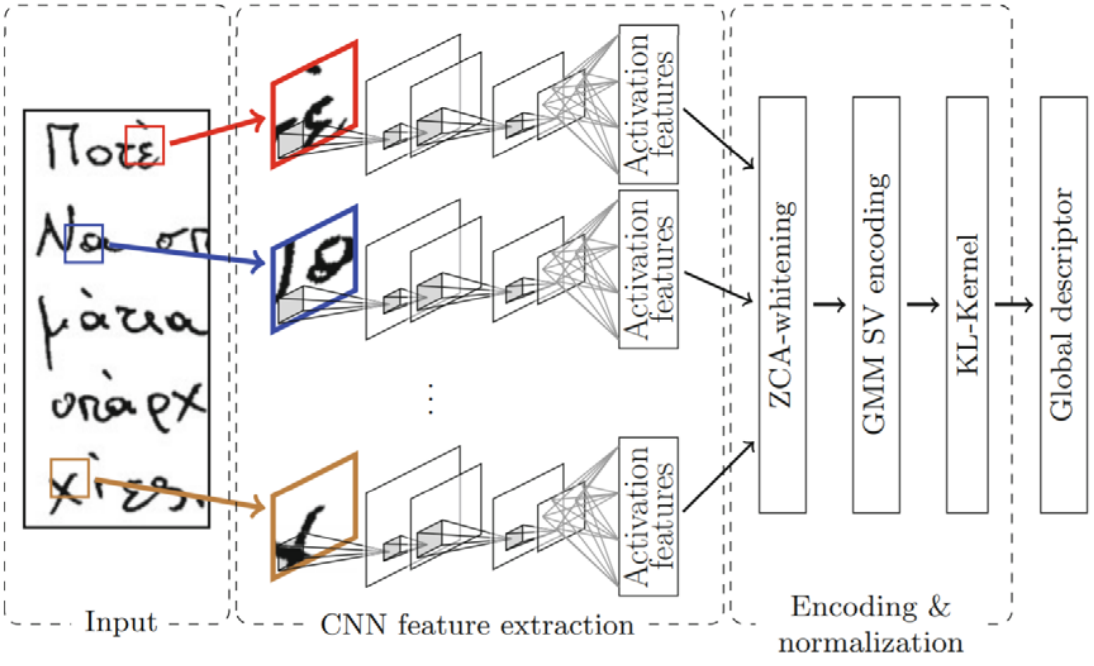


Figure 2.24: Diagram of the approach proposed in (Christlein et al. (2015b)).

Figure 2.25: Image patches cropped from the IAM dataset (Xing and Qiao (2016)).

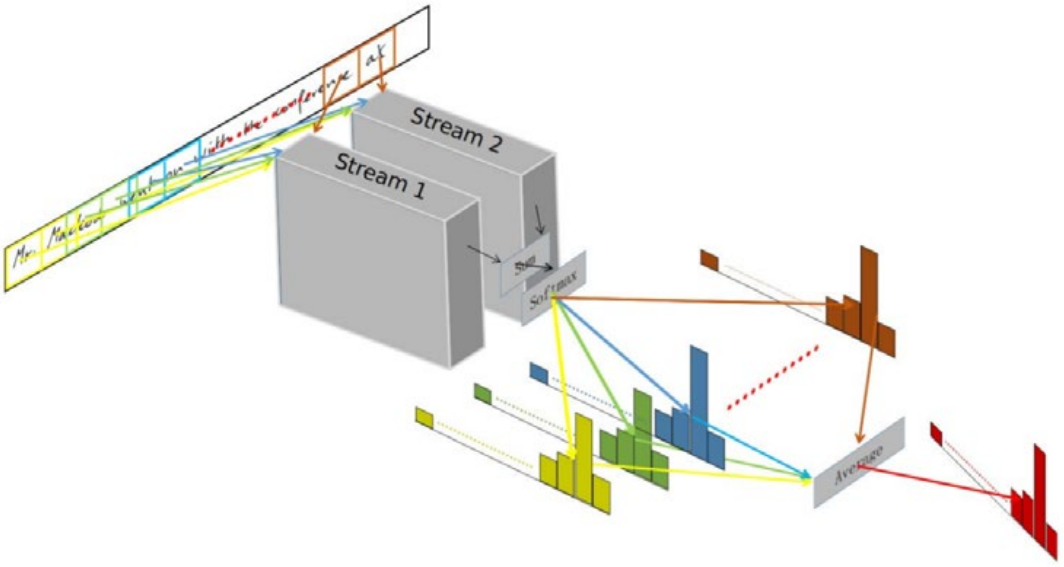


Figure 2.26: Pipeline of testing. Stream 1 and stream 2 share the same parameters (Xing and Qiao (2016)).

phase, the unseen documents are classified according to a similarity score averaging rule. The score vectors are computed using the softmax classification loss of the Deep-Writer model relative to each patch image. As a final decision, the overall system returns the writer of the query document with the highest average score. Experimental results on IAM and HWDB datasets showed superior SOTA performance. The authors found that different languages such as English and Chinese can have common patterns and that joint training can improve writer identification performance.

Later in (Christlein et al. (2017b)), the same authors improved the writer identification task by introducing an effective approach to learn CNN activation features in an unsupervised manner without requiring labelled data. As shown in Figure 2.27, scale-invariant key-point descriptors (SIFT) are first computed on the training database and clustered with surrogate classes. The clustered index represents each surrogate class. Then, a deep

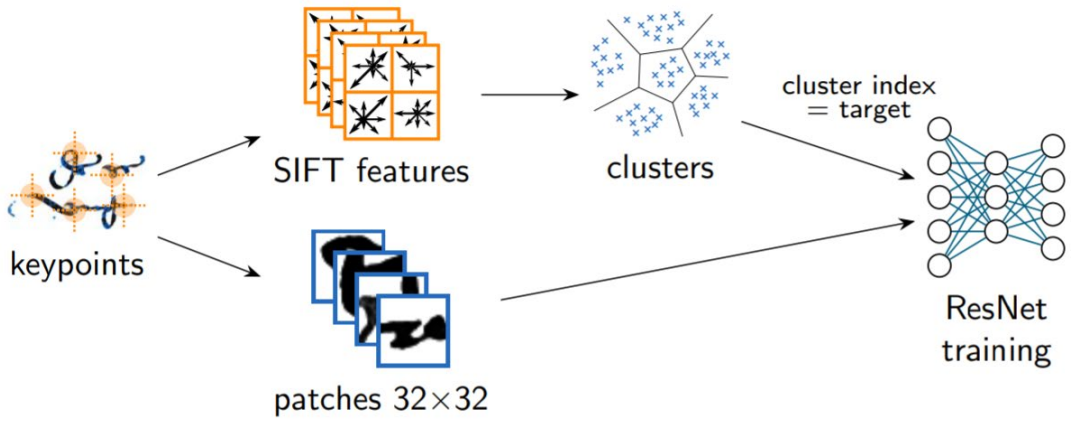


Figure 2.27: Diagram of the approach proposed in (Christlein et al. (2017b)).

residual network (ResNet) is trained with surrogate classes using patches extracted from each SIFT location (from the same SIFT keypoints). The deep features are extracted from the penultimate CNN activation layer, encoded, and classified using the linear exemplar support vector machines (E-SVM). The approach performed all SOTA methods on the ICDAR17 competition dataset for identifying historical document writers (Historical- WI).

The work presented by Christlein and Maier (2018) investigates how VLAD (Vector of Locally Aggregated Descriptors) encoding of CNN activations compares to triangulation embedding methods proposed in (Jégou and Zisserman (2014)) (cf. Figure 2.28). They evaluated different CNN network topologies to learn deep features from handwritten patches of size 32×32. These include LeNet, two different residual networks (ResNet), and models with different depths. The CNN activation features are then encoded into a global representation using triangulation and VLAD embeddings. The authors also investigated Generalized Max Pooling (GMP) as an alternative to Sum Pooling (SP), and they found that the GMP method slightly outperformed the SP method. The authors compared the Exemplar Support Vector Machines (ESVM) and PCA whitening when using VLAD or triangulation as the encoding approach in the classification phase. Experimental results on three benchmark databases (ICDAR13, CVL, and KHATT) showed the effectiveness of combining deep CNN activation features, VLAD encoding, normalization, and Exemplar SVMs for better writer identification.

An interesting work presented in (Keglevic et al. (2018)) proposes a learning feature representation for writer identification using the triplet CNN architecture of the DenseNet model. Instead of using the SoftMax layer and a Mean Square Error loss function, the authors employed three CNN branches with common weights in which the $L2$ distance

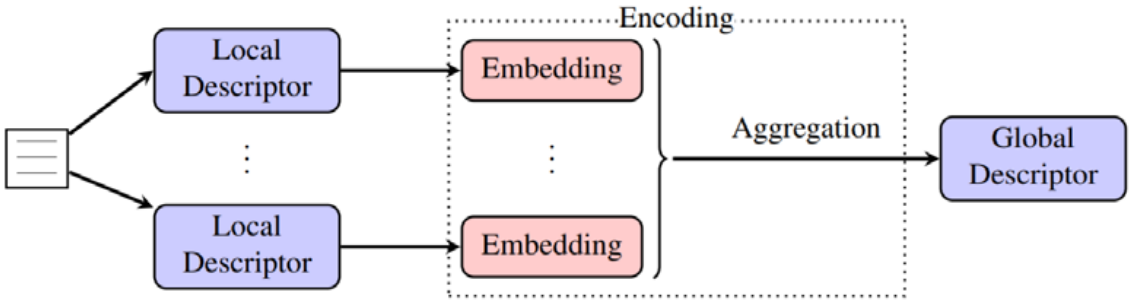


Figure 2.28: Encoding local descriptors to form a global representation that can be compared (Christlein and Maier (2018)).

was used to measure similarities. Then, the loss function with negative and positive distances is evaluated simultaneously in the three CNN streams (see Figure 2.29). As in (Christlein et al. (2017b)), image patches of size 32×32 are extracted based on the positions of the SIFT keypoints in the handwriting (cf. Figure 2.30). The triplet CNN is trained on these image patches by maximizing the inter-class distance and minimizing the intra-class distance. The deep activation features (corresponding to each document image) are encoded into a global descriptor using the Vector of Locally Aggregated Descriptors (VLAD). The evaluation of the writer's identification is performed in a leave-one-out strategy. A ranked hit list is generated according to the Euclidean and Cosine similarity of each document (taken once as query sample and represented by its feature vector) with other documents (taken as a reference base).

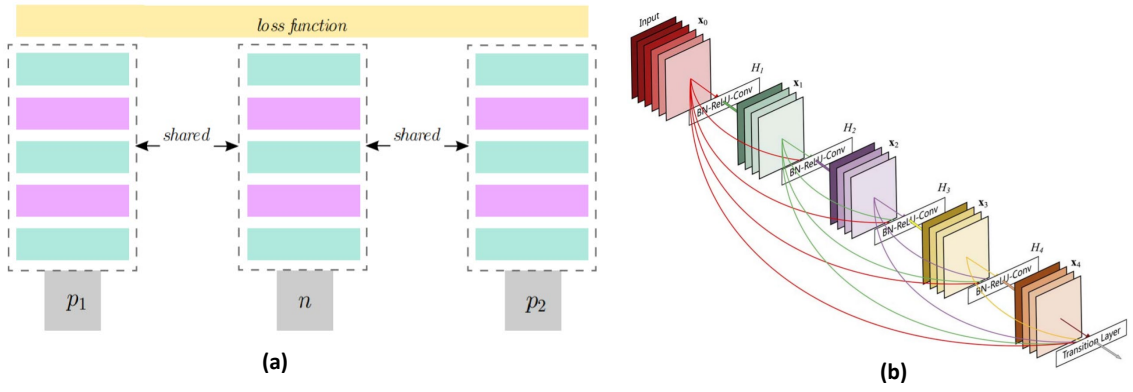


Figure 2.29: (a) Triplet CNN architecture. (b) Dense block with 5 layers (Keglevic et al. (2018)).

Figure 2.30: Sample patches of size 32×32 extracted at the SIFT keypoint locations (Keglevic et al. (2018)).

Recently, Chen et al. (2019) developed a semi-supervised feature learning for writer identification, trained with additional unlabeled data and the original labeled data simultaneously. For this purpose, they introduced a data augmentation called Weighted Label Smoothing Regularization (WLSR) method. It assigns a weighted uniform distribution of labels to the additional unlabeled data to regularize the CNN model to learn more discriminative features. As shown in Figure 2.31, the semi-supervised feature learning network, which is the baseline of ResNet-50, is trained from mixed original labeled data and additional unlabeled data (image patches extracted from handwritten documents). In the test phase, deep activation features (related to the image patches extracted from the unseen document) are computed from the penultimate fully connected layer, PCA-whitened, and encoded into a normalized global feature vector representing the query document. Euclidean distance was used as a similarity metric to compare and match the documents based on their feature vectors. Experiments conducted on the ICDAR2013 and CVL databases demonstrated the effectiveness of semi-supervised feature learning in accurately identifying the writers.

Kumar and Sharma (2020) introduced a CNN model called SEGmentation-free Writer Identification (SEG -WI) to improve writer identification. A training strategy without segmentation and preprocessing steps is the main contribution of this work. Unsegmented handwritten documents are the input for the SEG -WI model to be trained. It extracts deep features at different depth levels of the network, further selected by a region selection mechanism. A weighted voting mechanism is used to compute the loss of the network for writer identification. The main steps of the SEG-WI model are given in Figure 2.32.

In (He and Schomaker (2020)), a deep CNN model named FragNet is proposed for writer

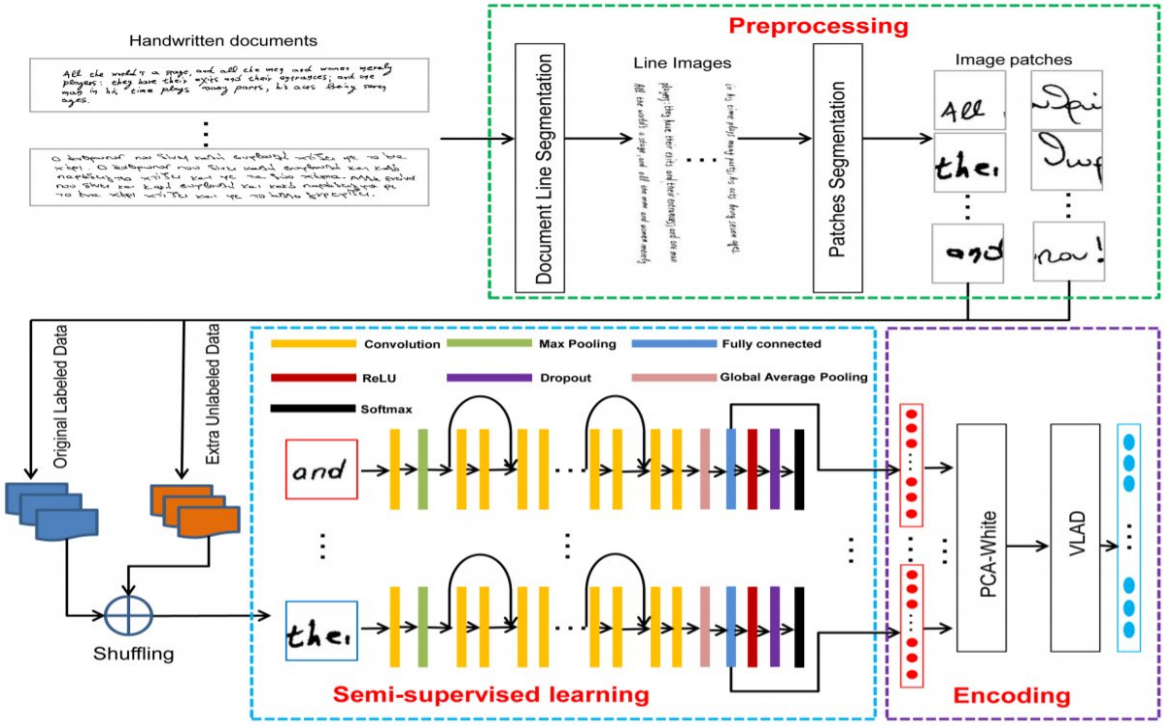


Figure 2.31: The pipeline of semi-supervised feature learning, which consists of three parts: preprocessing (green dotted box), semi-supervised learning (blue dotted box) and encoding (purple dotted box) (Chen et al. (2019)).

identification. As shown in Figure 2.33, the FragNet architecture is structured into two streams: a feature pyramid stream used to extract feature maps from the whole word image, and a CNN fragment stream (fragments as input) trained to identify the writer based on the predicted scores of fragment images and feature maps (computed from the feature pyramid). The authors have shown that the FragNet model trained with word images and fragments performs better (for writer identification) than networks trained with word images only. Experimental results on four benchmark databases have shown the effectiveness of the FragNet model in writer identification. Javidi and Jampour (2020) presented an offline writer identification system based on the concatenation of deep and local descriptor features. The CNN model is a residual network (ResNet) with 18 layers, where deep feature vectors of dimension 512 are extracted from the Flatten layer. As shown in Figure 2.34, the proposed local descriptor named Handwriting Thickness Descriptor (HTD) captures the handwriting thickness information by counting the all-black patches in the binary image, resulting in a feature vector with a size of 45 dimensions. Deep and HTD activation features are combined to form an extended end-to-end version of ResNet-18 for writer identification. Classification results (on four benchmark datasets) demonstrated the effectiveness of the system in identifying writers.

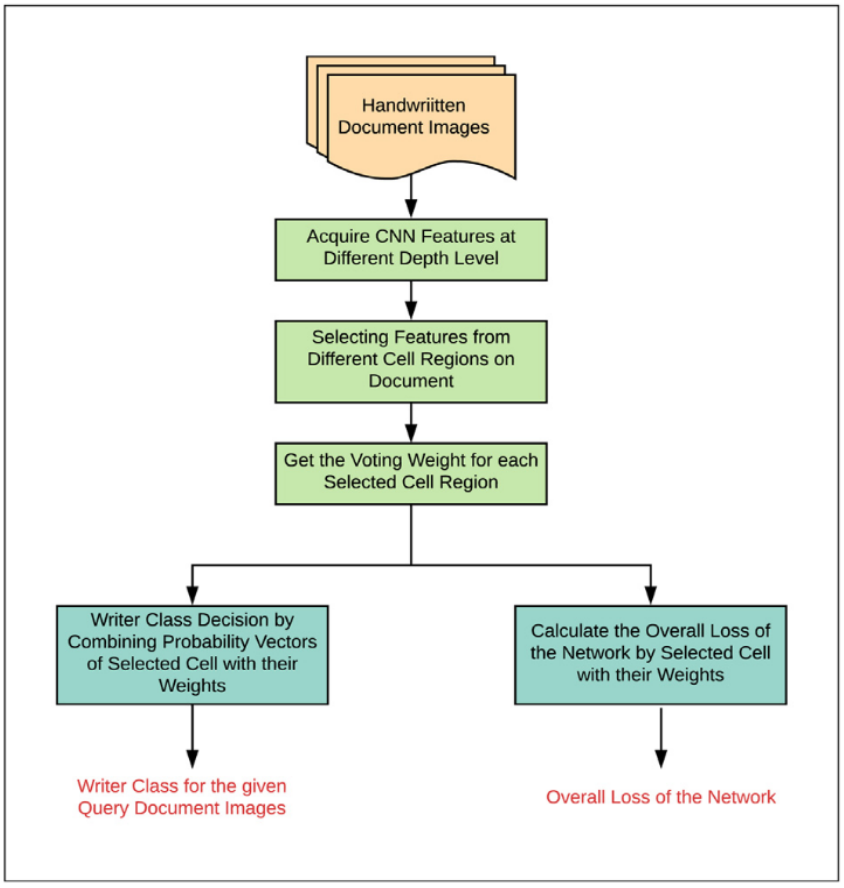


Figure 2.32: The main steps of the approach presented in (Kumar and Sharma (2020)).

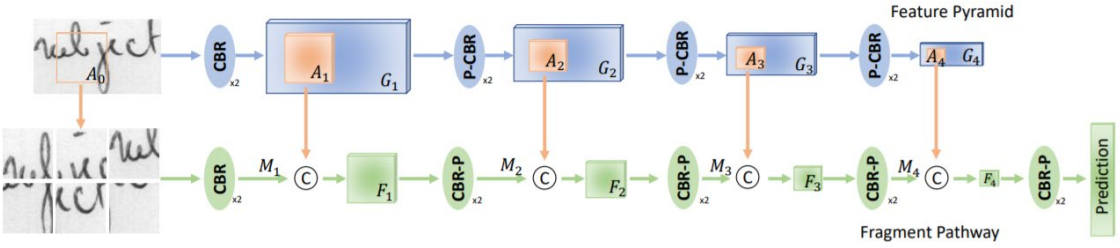


Figure 2.33: A FragNet network has two paths: feature pyramid (blue color), which accepts the whole word image as input, and fragment path (green color), which accepts the fragment as input. $(P)-CBR$ means the sequence of P: max-pooling, C: convolutional, B: batch normalization and R: ReLU layers. C with the circle is the concatenation operation. $\times 2$ means two blocks are stacked together. G_i and F_i are the ith feature maps in the feature pyramid and fragment path, respectively (He and Schomaker (2020)).

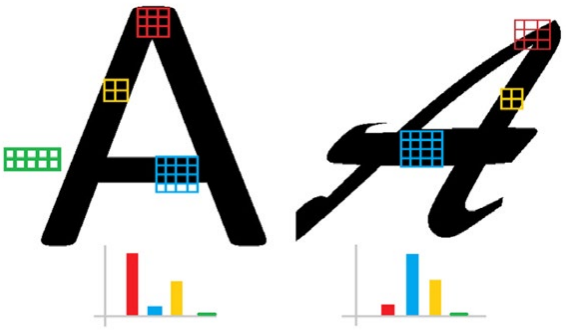


Figure 2.34: Handwriting Thickness Descriptor (HTD), counts fully black patches with different size (Javidi and Jampour (2020)).

2.2.5/ DISCUSSION

Texture features are easy to compute from handwriting without going through a binarization step. However, one of the drawbacks of texture-based methods, when used in conjunction with a classifier, is the higher number of parameters and computations required for image classification. This limitation, i.e., texture features are costly in processing time, is due to the high number of bins of extracted feature vectors. To solve this problem, post-processing methods such as principal component analysis (PCA), linear discriminant analysis (LDA), and generalized discriminant analysis (GDA) are mainly used as tools to reduce the higher dimensional space, whether it is 1D, 2D, or 3D data. However, these techniques may affect the overall performance in identifying the query authors. Texture-based methods detect redundant writing patterns and achieve correct performance, which can be further improved using robust classifiers.

Grapheme- and contour-based methods mainly rely on binarization and image segmentation techniques to characterize local writing patterns. Therefore, the recognition and identification performance is somewhat limited and dependent on the ability of these techniques to capture local structural information of the writing. Texture, contour, and grapheme approaches depend on features computed by algorithms and basically designed by humans. Currently, these features can be learned automatically using Deep Learning (DL). It provides an easier way to obtain the desired features for the task under consideration. However, deep learning requires large labeled training data to learn how to classify images of a particular application. For writer identification, large handwriting training data is needed to learn and characterize the writing style, which is not always present in some scenarios. In this case, traditional methods perform better or equivalent to deep learning.

2.3/ CONCLUSION

In this chapter, we have presented, in chronological order, a comprehensive literature review of the known approaches that have been proposed for writer identification. We have categorized these approaches into texture-based, grapheme-based, contour-based, and deep learning methods. We have also discussed the challenges, factors, performance, and criticisms of these approaches to better characterize the handwriting style. To improve writer identification, researchers agree that the feature extraction step is one of the most critical modules in a writer identification system. Indeed, using a very powerful classifier cannot compensate for a poorly matched feature representation.

In the following chapters, we discuss and present our proposals to improve state-of-the-art performance to better characterize the writing style and accurately classify and identify the authorship of handwritten documents. Note that our proposed approaches in this thesis belong to the texture and deep learning categories.

II

CONTRIBUTION

3

TEXTURE FEATURES-BASED WRITER IDENTIFICATION

3.1/ INTRODUCTION

Most existing work for writer identification generally considers the following pipeline: image pre-processing, feature extraction, and classification. These approaches mainly focus on feature extraction, as the aim is to capture necessary and discriminative information from the image that reflects and characterizes the writing style. Feature engineering is an essential part of a pattern recognition system that can improve or decrease classification performance. Features, also called descriptors, attributes, variables, measures, are discriminative compared to the original image pixels. Different types of feature methods have been reported in the literature, and each has its limitations and advantages in terms of applicability, suitability, performance, and computational complexity. It is challenging to meet all these criteria in the feature extraction step, especially when dealing with handwritten documents with complex writing styles. At this level, the goal is to accurately identify the writer using useful features extracted from handwriting images in a text-independent manner.

Handwriting can be viewed as a texture image that contains grayscale or binary variations that form specific recurring patterns. Extracting such features defines a synthetic characterization of handwriting in a 2D dimension. A well-designed and defined feature extraction method improves writer identification through appropriate pre-processing and classification methods. To this end, we propose several approaches for writer identification based on texture features. Our contributions address all main steps of an automatic writer identification system, including image pre-processing and segmentation, feature extraction, and classification methods. The following four texture-based systems are proposed to help to solve some challenges encountered in these different steps: (1) Block Wise Local Binary Count (BWLBC)-based system. The BWLBC operator characterizes the variability of writing style by capturing pixels' distribution within small binary

blocks; (2) Handcrafted feature-based system. The handcrafted descriptors Local Binary Patterns (LBP), Local Ternary Patterns (LTP), and Local Phase Quantization (LPQ) are applied to small regions (zones) of interest in the writing to extract texture features. We also introduce a dimensionality reduction technique to reduce the computational cost of the subsequent classification process; (3) An effective approach based on the Cross multi-scale Locally encoded Gradient Patterns (CLGP) descriptor to capture texture information of the writing image. It consists of encoding CLGP features using the Histograms of Oriented Gradients (HOG) method; (4) The fourth approach computes local intensity gradients of the writing within non-overlapping blocks using the Local gradient full- Scale Transform Patterns (LSTP) method. This feature gives the overall system the ability to extract more relevant information to improve the identification task. The writer classification is performed using the nearest neighbor with a new strategy to compute similarities between the handwritten documents.

This chapter presents in detail the theoretical description of our proposed texture-based systems for text-independent offline writer identification. Note that the experiments and evaluations to validate the effectiveness of these systems are discussed later in Chapter 5.

3.2/ Overall Pipeline

Our proposed system for writer identification based on texture features involves three main steps: 1) image preprocessing and segmentation, 2) feature extraction, and 3) classification process. In our methodology, handwritten documents are considered as texture images, and their respective features are extracted from different regions of interest (i.e., connected component sub-images). The proposed system for writer identification is shown in Figure 3.1. Each of the main steps of the system is described in detail in the following subsections.

3.2.1/ Image pre-processing and segmentation

To enhance the performance of writer identification, we perform some basic image preprocessing techniques on the handwritten input image to make it suitable for the segmentation phase and reduce the complexity of the subsequent processing modules. First, the handwritten images, which are in RGB format, are converted to grayscale images. Then, except for the IFN/ENIT database, which contains handwritten patterns in binary format (cf. benchmark and experimental setup in Chapter 5), global thresholding is performed on the handwritten grayscale images using Otsu's method described in (Otsu (1979)) to convert them into binary images. For a grayscale image, binarization consists of first

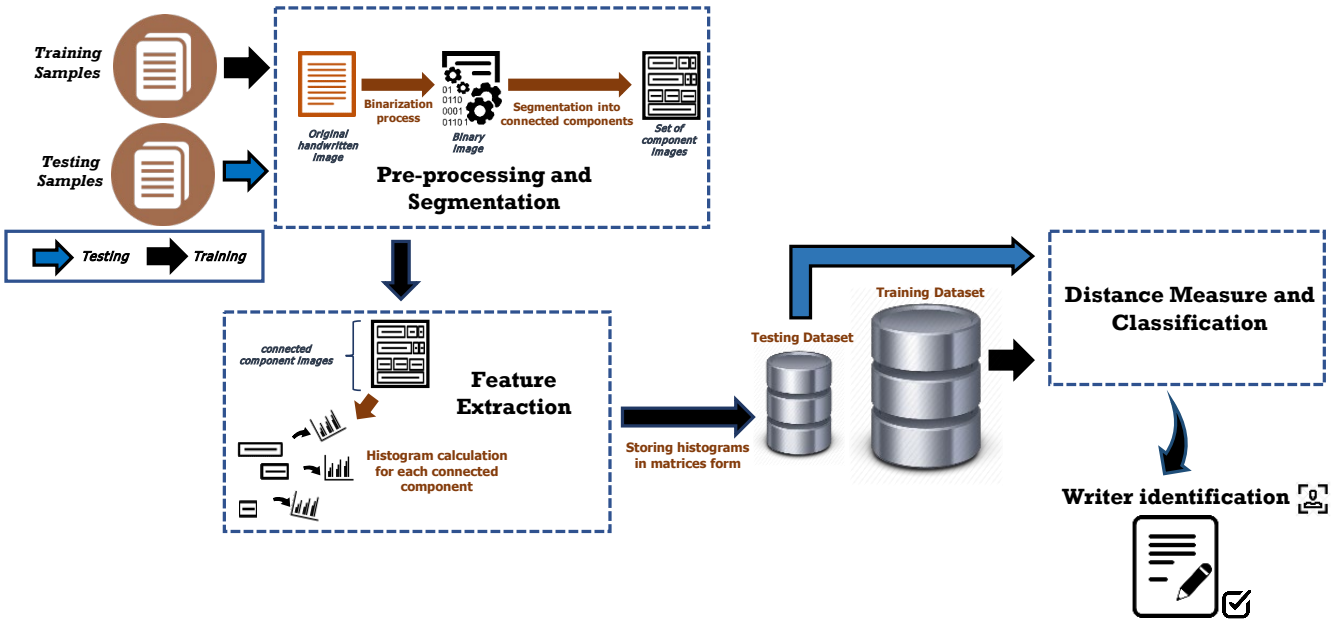


Figure 3.1: Flowchart of the proposed writer identification system.

determining a normalized threshold and minimizing the intraclass variance of the black and white pixels. Figures 3.2 and 3.3 show that this binarization process is necessary to clearly distinguish the background from the foreground (writing pixels). In general, the grayscale threshold should be closer to the intensity of the foreground writing to remove unwanted background noise while maximizing the useful information in the scanned image. As can be seen from the examples in Figures 3.2 and 3.3, the complement of the binary image is computed to obtain the same fixed binary format, i.e., black as the background and white as the foreground (white pixels correspond to the ink). The obtained preprocessed image is then segmented into a set of isolated components using a labeling procedure that assigns a label to each connected object in the binary sample. Only the significant and representative connected components are retained, i.e., those with minimal amounts of text pixels such as diacritics and random writing traces (random patterns produced during the image acquisition) are considered as unwanted details and discarded. In the next step, a bounding box is assigned to each labeled region. It is defined as the smallest rectangle that completely encloses and circumscribes the corresponding connected component. At this stage, each labeled region (connected component) is considered within its bounding box. Figure 3.4 shows an example of connected component extraction from an Arabic handwritten word, where small components are suppressed as they are considered as unwanted details. The algorithm of the preprocessing and

segmentation stage is summarized in Algorithm 1.

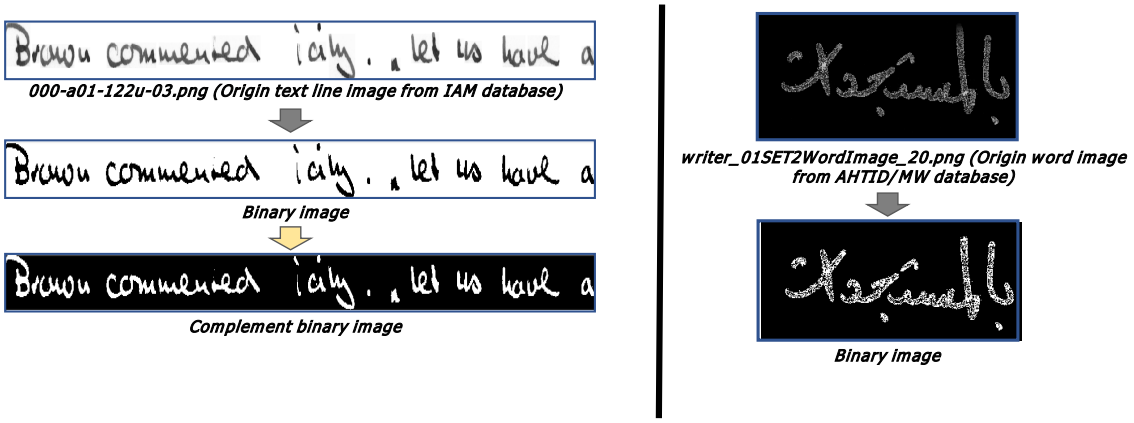


Figure 3.2: Example of binarization of handwritten samples from IAM (Marti and Bunke (2002)), and AHTID/MW (Mezghani et al. (2012)) databases.

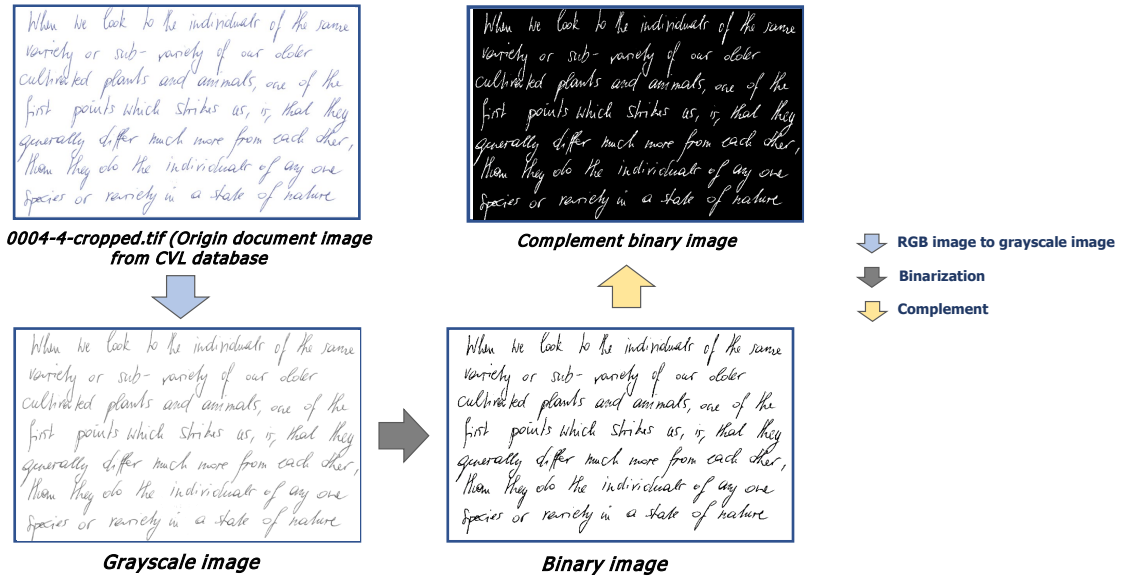


Figure 3.3: Example of binarization of handwritten sample from CVL (Kleber et al. (2013)) database.

Algorithm 1: Pre-processing and segmentation step

Input : S // handwriting sample.

Output: $[C^i]$ list, $i = 1, 2, \cdots, n$, where each element is an integer. // connected component sub-images.

/* Segment the input sample into connected component sub-images. */

$\mathbf{S_b} = \mathbf{binarizefunction(S)}$; // binarize the S image into binary image S_b.

$\mathbf{cord} = \mathbf{cordinatefunction(S_b)}$; // get the coordinates $cord(i) = [x_{min}(i),\ y_{min}(i),\ width(i),\ height(i)]$, of all connected components C from the binary image S_b; The lower left corner of the detected bonding box of the connected component i is the the pixel with spatial coordinates $(x_{min}(i), y_{min}(i))$; $width(i)$ and $height(i)$ are the width and height of the bounding box i.

/* Extract connected component sub-images from the binary input image S_b, $S_b = \cup_{i=1}^{card(cord)} C^i$ */

for $i \leftarrow 1$ **to** N_{cord} **do**

  // N_{cord} is the number of extracted bounding boxes (corresponding to connected components) in the binary image S_b.

  $\mathbf{C^i} = \mathbf{cropfunction(S_b, cord(i))}$;

  $\mathbf{C^i} = \mathbf{S_b(x_{min}(i) : (x_{min}(i) + width(i)),\ y_{min}(i) : (y_{min}(i) + height(i)))}$;

  if $size(C^i) >= 50 \times 50$ **then**

    save$(\mathbf{C^i})$;

  else

    skip; // ignore accidental writing traces and diacritics.

  end if

end for

return $[C^i]$ // output. n is the number of segmented connected components.

3.2.2/ FEATURE EXTRACTION METHODS

Feature extraction is a fundamental step in writer identification, using various techniques to capture local variations in writing style. Extracting such extreme variations increases the quality of the identification task. At this stage of our system, we have a set of related components for each handwritten document. It is still a difficult problem to accurately identify the corresponding writer since only the related components are obtained, and there is no information estimation about their content yet, which is necessary for the identification process. Feature extraction consists of defining a synthetic description of the writing style to be characterized in two-dimensional space from scanned handwritten images. The goal is to extract, characterize and consequently index common features

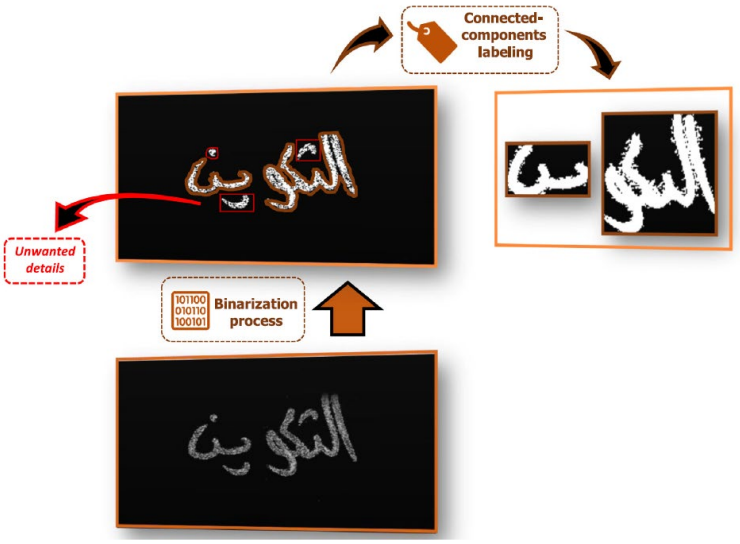


Figure 3.4: Pre-processing and segmentation of an Arabic word taken from the AHTID/MW (Mezghani et al. (2012)) database.

from handwritten images belonging to the same writer (class). If these features are inappropriately extracted from the writing, it will unfavorably affect the classifier used to identify unseen documents. For this reason, four texture-based methods are proposed to accurately characterize the writing style from the preprocessed connected component sub-images. Each method is described in detail below.

3.2.2.1/ BW-LBC DESCRIPTOR

Motivated by the effectiveness of small writing fragments and texture blocks in characterizing the writer's individuality (Bertolini et al. (2013); Hannad et al. (2016)), we propose a computationally efficient, high-quality, and conceptually simple descriptor called Block Wise Local Binary Count (BW-LBC) to characterize the connected components of handwriting images. To show its effectiveness, the BW-LBC model is compared with LBP, LTP, and LPQ descriptors, which are the most commonly used methods for writer identification (Hannad et al. (2016); Khan et al. (2016); Bertolini et al. (2013)). The analysis of this experiment is given in Chapter 5 (cf. Section 5.3.1). The BW-LBC operator characterizes the variability of the writing style within small blocks by computing the distribution of occurrences of pixels corresponding to writing in binary component images. For our system, these binary component images are the connected components previously obtained using the segmentation method in Section 3.2.1. Formally, the BW-LBC operator proceeds as follows. The preprocessed connected component sub-images are first resized into the same uniform window size $W \times W$. Then, each component C is scanned from top to bottom

and left to right and divided into N×N non-overlapping blocks $B_{i,i=1,..,N\times N}$ ($C=\bigcup_{i=1}^{N\times N} B_i$). For each block B_i ($i = 1, 2 \ldots N\times N$), the appearance frequency of the white pixels, i.e., the ink, is calculated. Finally, each block B_i is henceforth characterized by its feature $BWLBC(i)$, which can be computed as follows:

$$BWLBC(i) = \sum_{\mathbf{x}_p \in B_i} \delta(1, B_i(\mathbf{x}_p)),\ i = 1, 2 \ldots (N\times N) \tag{3.1}$$

where $B_i(\mathbf{x}_p)$ is the value of pixel $\mathbf{x}_p$ in the block B_i ($i = 1, 2 \ldots N\times N$) and δ being the Kronecker delta function, defined as follows:

$$\delta(\mathbf{x}, \mathbf{y}) = \begin{cases} 1\ , & if\ \ \mathbf{x} = \mathbf{y} \\ 0\ , & if\ \ \mathbf{x} \neq \mathbf{y} \end{cases} \tag{3.2}$$

The resulting normalized F_{BWLBC} feature vector is then calculated by the following forms:

$$\begin{aligned} F_{BWLBC} =& \frac{1}{W_B} \prod_{i=1}^{(N\times N)} BWLBC(i) \\ =& \frac{1}{W_B} \prod_{i=1}^{(N\times N)} \sum_{\mathbf{x}_p \in B_i} \delta(1, B_i(\mathbf{x}_p)),\ i = 1, 2 \ldots (N\times N) \end{aligned} \tag{3.3}$$

where $\prod$ is the concatenation operator and $W_B = \frac{\mathrm{W}\times\mathrm{W}}{N\times N}$ is the block size. Note that the normalized $F_{BWLBC}(i)$ represents the histogram bin number i in the final F_{BWLBC} feature vector. The whole procedure shown in Figure 3.5 to compute the $BWLBC$ code can be summarized as follows:

- Step 1: The input handwriting document with the writer label w_r is converted to a binary image D_{w_r}.

- Step 2: The obtained binarized image D_{w_r} is segmented into connected components ($D_{w_r} = \bigcup_{k=1\ \eta(C^k_{w_r})>\nu}^{N_{w_r}} C^k_{w_r}$) ; where $N_{w_r} = card(D_{w_r})$ is the number of connected components in the document D_{w_r} and $\eta(C^k_{w_r})$ is the number of pixels of the component $C^k_{w_r}$ and ν is a small threshold ($\nu << W \times W$). The connected components with very small proportions of writing pixels, i.e., with area less than ν pixels, such as diacritics and random writing traces, are considered as unwanted details and then discarded. At this stage, each preserved connected component is considered within its bounding box.

- Step 3: Connected components are resized into the same uniform window size $W \times W$.
- Step 4: Each component $C^k_{w_r}$ is scanned from top to bottom and from left to right and divided into $N \times N$ non-overlapping blocks $B^k_{(w_r,i),i=1,..,N\times N}(C^k_{w_r} = \bigcup_{i=1}^{N\times N} B^k_{(w_r,i)})$.
- Step 5: Each block $B^k_{(w_r,i),i=1,...,N\times N}$, is defined by its feature $BWLBC_{(k,w_r)}(i)$ using Eq. 3.1.
- Step 6: For each connected component $C^k_{w_r}$, the corresponding normalized feature vector $F^{(k,w_r)}_{BWLBC}$ is calculated using Eq. 3.3.
- Step 7: The input handwriting document with the writer label w_r is henceforth represented by a set of F_{BWLBC}-histograms $\{F^{(k,w_r)}_{BWLBC}, k = 1, ..., N_{w_r}\}$.

It is worth noting that when using the BW-LBC descriptor as feature extraction, the system performance strongly depends on the chosen number of blocks $(N \times N)$ and the window size $(W \times W)$. Indeed, different values of N and W would lead to a different encoding of the BW-LBC binary operator. The experiment to study the effects of these parameters on system performance is discussed later in Chapter 5 (Section 5.3.1.2).

Figure 3.5 shows an example of the procedure for computing the BW-LBC code. In this example, starting from the connected component within its bounding box of 187×81 pixels, the resized component (300×300) is divided into $N \times N$ $(= 7x7 = 49)$ blocks of W_B $(= \frac{W \times W}{N \times N} = 43 \times 43)$ pixels. The number of white pixels then represents each block. The

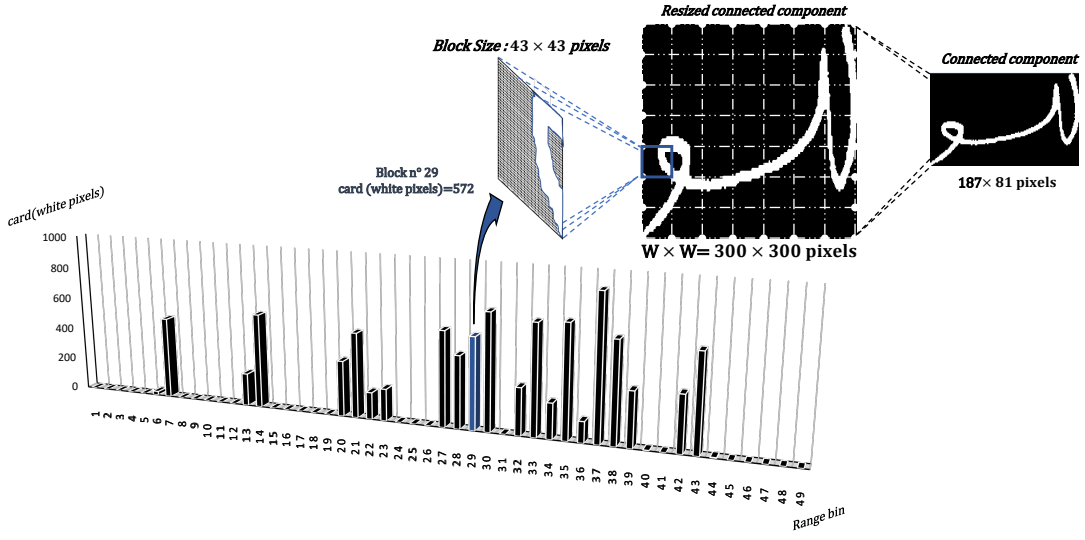


Figure 3.5: The local BW-LBC code computation process.

concatenation of the white pixel counts of all blocks forms the final feature vector. Note that the chosen number of blocks $(N \times N)$ in the binary sample represents the resulting dimensionality of the BW-LBC vector.

3.2.2.2/ ZONES-BASED HANDCRAFTED DESCRIPTOR

In this approach, three well-known state-of-the-art texture descriptors are used, including histograms of Local Binary Patterns (LBP), Local Ternary Patterns (LTP), and Local Phase Quantization (LPQ). These texture operators, which are commonly used in writer identification problems (Hannad et al. (2016); Khan et al. (2016); Bertolini et al. (2013)), are briefly described in the following.

Traditional local binary patterns (LBP). Ojala et al. (2002) introduced a robust descriptor for texture classification called a local binary pattern (LBP). It is defined as a gray-scale invariant operator that measures the local contrast for texture analysis and summarizes the gray-level structure into a local region. As shown in Figure 3.6, the traditional LBP operator labels the pixels of the grayscale image I in a 3×3 local neighborhood by thresholding the eight neighboring pixels with the central pixel. The mathematical formulation of LBP for a referenced pixel $\mathbf{x}_c$ is defined by the following equation:

$$LBP(\mathbf{x}_c) = \sum_{p=0}^{P-1} \phi(\mathcal{I}(\mathbf{x}_p) - \mathcal{I}(\mathbf{x}_c)) \times 2^p \tag{3.4}$$

where $\mathcal{I}(\mathbf{x}_c)$ is the central pixel gray-scale value, $\mathcal{I}(\mathbf{x}_p)$ is the value of its neighbors, P is the number of neighbors and $\phi(\cdot)$ is the Heaviside step function (cf. Eq. 3.5).

$$\phi(\mathbf{z}) = \begin{cases} 1\,, if\ \mathbf{z} \geq 0 \\ 0\,, if\ \mathbf{z} < 0 \end{cases} \tag{3.5}$$

LBP generates 256 (2^8) possible patterns (cf. Table 3.1). In this work, LBP features are computed in a 3×3 square neighborhood, i.e., the LPB radius and the number of neighbor pixels are set to 1 and 8, respectively.

Local ternary patterns (LTP). Tan and Triggs (2010) proposed an improved variant of LBP called Local Ternary Patterns (LTP), which is more robust to noise than LBP. LTP extracts 3-valued $(1, 0, -1)$ local grayscale codes by encoding the difference between the central pixel value $\mathcal{I}(\mathbf{x}_c)$ and adjacent values $\mathcal{I}(\mathbf{x}_p)$ using a threshold. The kernel function of the LTP code is defined as follows (cf. Eq. 3.6):

$$LTP(\mathbf{x}_c) = \sum_{p=0}^{P-1} \vartheta_\tau(\mathcal{I}(\mathbf{x}_c), \mathcal{I}(\mathbf{x}_p)) \times 2^p \tag{3.6}$$

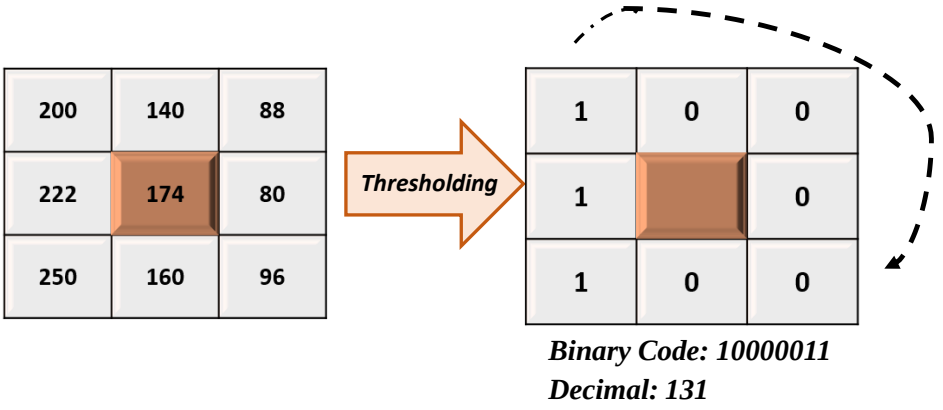


Figure 3.6: Example of LBP encoding scheme.

where τ is the user-specified threshold and $\vartheta_\tau(\cdot)$ is the Ternary Thresholding function defined as follows:

$$\vartheta_\tau(\mathbf{a},\mathbf{b}) = \begin{cases} 1, & \text{if } \mathbf{a} \geq \mathbf{b}+\tau \\ 0, & \text{if } \mathbf{b}-\tau \leq \mathbf{a} < \mathbf{b}+\tau \\ -1, & \text{otherwise} \end{cases} \tag{3.7}$$

To simplify the threshold function of the LTP descriptor, Tan and Triggs (2010) introduced a different representation based on the extraction of two binary patterns, splitting the ternary pattern into its positive and negative parts, as shown in Figure 3.7. LTP generates 512 (2^9) possible patterns (cf. Table 3.1).

Local phase quantization (LPQ). The local phase quantization (LPQ) operator, originally proposed in (Ojansivu and Heikkilä (2008)), is a texture descriptor based on the blur invariance property of the Fourier phase spectrum. It uses the phase information estimated in local M-by-M neighborhoods at each pixel position $\mathbf{x}$ of the image f. The local spectra are computed using a 2-D discrete Fourier transform (DFT) or, more precisely, a short-term Fourier transform (STFT) defined by:

$$F(\mathbf{u},\mathbf{x}) = \sum_{\mathbf{y}\in N_\mathbf{x}} f(\mathbf{x}-\mathbf{y})\, e^{-j2\pi\mathbf{u}^\mathbf{T}\mathbf{y}} = W_\mathbf{u}^\mathbf{T} f_\mathbf{x} \tag{3.8}$$

$W_\mathbf{u}$ being the basis vector of the 2-D DFT at frequency $\mathbf{u}$, $f_\mathbf{x}$ is the vector containing all M^2 image samples of the $(M\text{x}M)$ neighbourhood $(N_\mathbf{x})$ of $\mathbf{x}$. Local Fourier coefficients are computed at four frequency points $\mathbf{u}_1 = [\mathbf{a},0]^\mathbf{T}$, $\mathbf{u}_2 = [0,\mathbf{a}]^\mathbf{T}$, $\mathbf{u}_3 = [\mathbf{a},\mathbf{a}]^\mathbf{T}$ and $\mathbf{u}_4 = [\mathbf{a},-\mathbf{a}]^\mathbf{T}$; where $\mathbf{a}$ is a small scalar frequency. The resulting vector of each pixel position is given

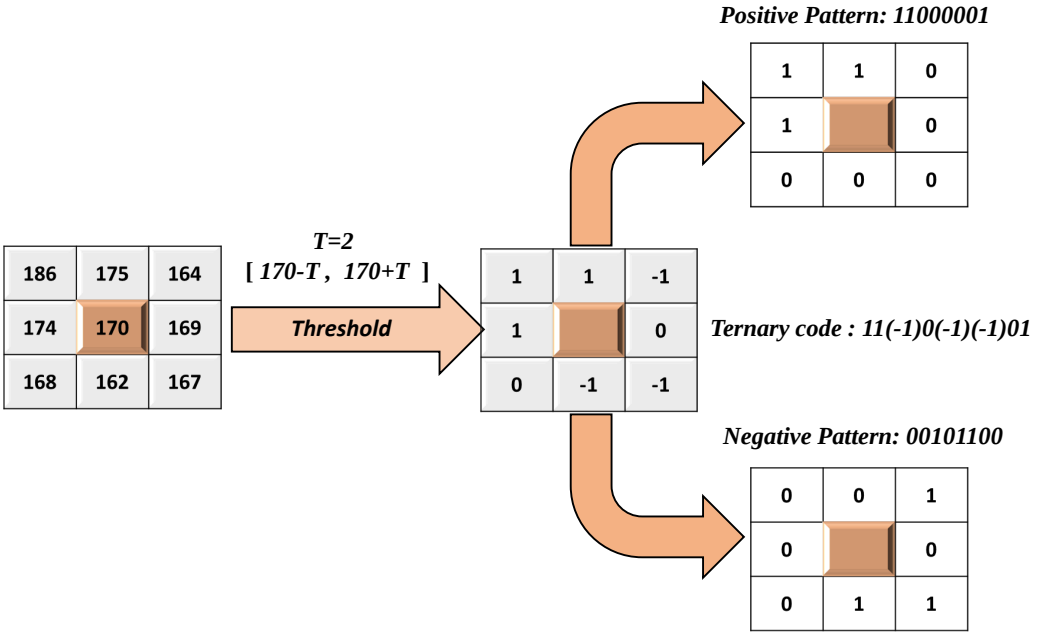


Figure 3.7: Example of LTP encoding scheme.

by:

$$F(\mathbf{x}) = [F(\mathbf{u}_1, \mathbf{x}), F(\mathbf{u}_2, \mathbf{x}), F(\mathbf{u}_3, \mathbf{x}), F(\mathbf{u}_4, \mathbf{x})] \tag{3.9}$$

The phase information in the Fourier coefficients is calculated by using a simple scalar quantization for each component in $F(\mathbf{x})$:

$$q_j = \begin{cases} 1\,, & if \quad g_i \geq 0 \\ 0\,, & \text{otherwise} \end{cases} \tag{3.10}$$

where g_i is the jth component of the vector G(·) (cf. Eq. 3.11):

$$G(\mathbf{x}) = [Re\{F(\mathbf{x})\}, Im\{F(\mathbf{x})\}] \tag{3.11}$$

The quantized coefficients are given as integer values between 0 and 255 using the following binary coding:

$$b = \sum_{j=0}^{7} q_j 2^j \tag{3.12}$$

Th LPQ method generates 256 (2^8) possible patterns (cf. Table 3.1).

Table 3.1: The tested texture descriptors

Texture operators	**Dimension (D_{im})**
Local Binary Pattern (LBP)	256
Local Ternary Pattern (LTP)	512
Local Phase Quantization (LPQ)	256

Proposed feature extraction process. The feature extraction process applied in the proposed system is shown in Figure 3.8. First, the previously obtained preprocessed component images for both the test and training sets are fitted into the same uniform window of 50×50 pixels. Then, each connected component image C is subjected to LBP, LTP, or LPQ descriptor coding to obtain the feature image FC. Since the original component image is in binary format, it can be observed that there are irrelevant features in the feature map FC, which may not be informative and could then be ignored. Therefore, we propose a simple dimensionality reduction technique to reduce the computational cost of the subsequent classification process. It normalizes the obtained feature image FC by a factor F, as shown in the example in Figure 3.8. To incorporate more spatial information in the final descriptor, the obtained normalized code map ($\frac{FC}{F}$) is scanned from top to bottom and from left to right and spatially divided into small spatial ($N_z \times N_z$), non-overlapping parts, which are called zones $Z_{i,i=1,..,N_z \times N_z}$ ($FC = \bigcup_{i=1}^{N_z \times N_z} Z_i$), and a histogram of LBP, LTP or LPQ codes is extracted from each zone. Thus, each zone Z_i ($i = 1, 2 \ldots N_z \times N_z$) in the feature image FC is characterized by a ($\frac{\mathbf{D_{im}}}{\mathbf{F}}$) bins histogram that eliminates the non-discriminatory bins. The feature histogram h_i characterizing each zone Z_i ($i = 1, 2 \ldots N_z \times N_z$) is calculated using Eq. 3.13.

$$\mathbf{h}_i(\lambda) = \sum_{\mathbf{x_p} \in \mathbf{Z_i}} \delta(\lambda, Z_i(\mathbf{x}_p))) \tag{3.13}$$

where $Z_i(\mathbf{x}_p)$ is the value of pixel $\mathbf{x}_p$ in the zone Z_i ($i = 1, 2 \ldots N_z \times N_z$), $\lambda \in [0, N_{bins}]$, $N_{bins} = \frac{\mathrm{D_{im}}}{F} - 1$ is the number of bins of the feature histogram h_i, D_{im} is the descriptor dimension given in Table 3.1, F is the dimensionality reduction factor, and $\delta(\cdot)$ is the Kronecker delta function defined as follows:

$$\delta(\mathbf{x}, \mathbf{y}) = \begin{cases} 1\,, & if \;\; \mathbf{x} = \mathbf{y} \\ 0\,, & if \;\; \mathbf{x} \neq \mathbf{y} \end{cases} \tag{3.14}$$

All these regional subhistograms $h_{i,i=1,..,N_z \times N_z}$ of dimensionality ($\frac{\mathbf{D_{im}}}{\mathbf{F}}$) are concatenated through Eq. 3.15 to form the holistic connected component representation H of dimen-

sionality $D_c = (\frac{\mathbf{D_{im}}}{\mathbf{F}}) \times N_z \times N_z$.

$$H = \prod_{i=1}^{(N_z \times N_z)} \mathbf{h}_i \tag{3.15}$$

Where $\prod$ is the concatenation operator. The histogram sequence concatenation on the normalized feature image ($\frac{FC}{F}$) allows a complete generalization of the writing intensity distributions into local regions (zones) that strongly characterize the writing style and discriminate the large variability between handwriting. The computational time of the classification process increases progressively with the number of zones $N_z \times N_z$, which is quite natural since the classification is performed by comparing the distances between corresponding component histograms. Indeed, the dimensionality D_c of the feature histogram H increases with the increase of the number of zones (D_c=($N_z \times N_z$)$\times$($\frac{\mathrm{D_{im}}}{F}$)). However, the dimensionality reduction factor F, introduced to ignore irrelevant features (bins), allows for a reduction in computation time since D_c decreases as F increases.

The feature extraction process to compute the final feature histogram H can be summarized in the following steps (see Figure 3.8):

- Step 1: The input handwriting sample (document or set of word/text line images) with writer label w_m is converted to a binary image S_{w_m}.
- Step 2: The obtained binarized image S_{w_m} is segmented into labeled regions (i.e., connected components) ($S_{w_m} = \bigcup_{j=1 \ \zeta(C^j_{w_m})>\rho}^{N_{w_m}} C^j_{w_m}$) ; where $N_{w_m} = card(S_{w_m})$ is the number of connected components in the sample S_{w_m} and $\zeta(C^j_{w_m})$ is the number of writing pixels in the component $C^j_{w_m}$ and ρ is a small threshold ($\rho << 50 \times 50$). Accidental writing traces and diacritics are considered as noise and then removed (labeled regions with very small proportions of writing pixels, i.e., with a surface less than ρ pixels). The threshold ρ is experimentally set to $\rho = 12 \times 12$ pixels for the IAM database and $\rho = 10 \times 10$ pixels for the IFN/ENIT, CVL and AHTID/MW databases.
- Step 3: Component images are resized to the same fixed window size of 50×50 pixels.
- Step 4: For each component $C^j_{w_m}$, the corresponding feature image $FC^j_{w_m}$ is computed using the texture descriptor encoding and then normalized by the dimensionality reduction factor F.
- Step 5: The normalized feature image ($\frac{\mathbf{FC^j_{wm}}}{\mathbf{F}}$) is scanned from top to bottom and from left to right and divided into $N_z \times N_z$ zones $Z^{(j,w_m)}_{i,i=1,..,N_z \times N_z}$ ($C^j_{w_m}$= $\bigcup_{i=1}^{N_z \times N_z} Z^{(j,w_m)}_i$).

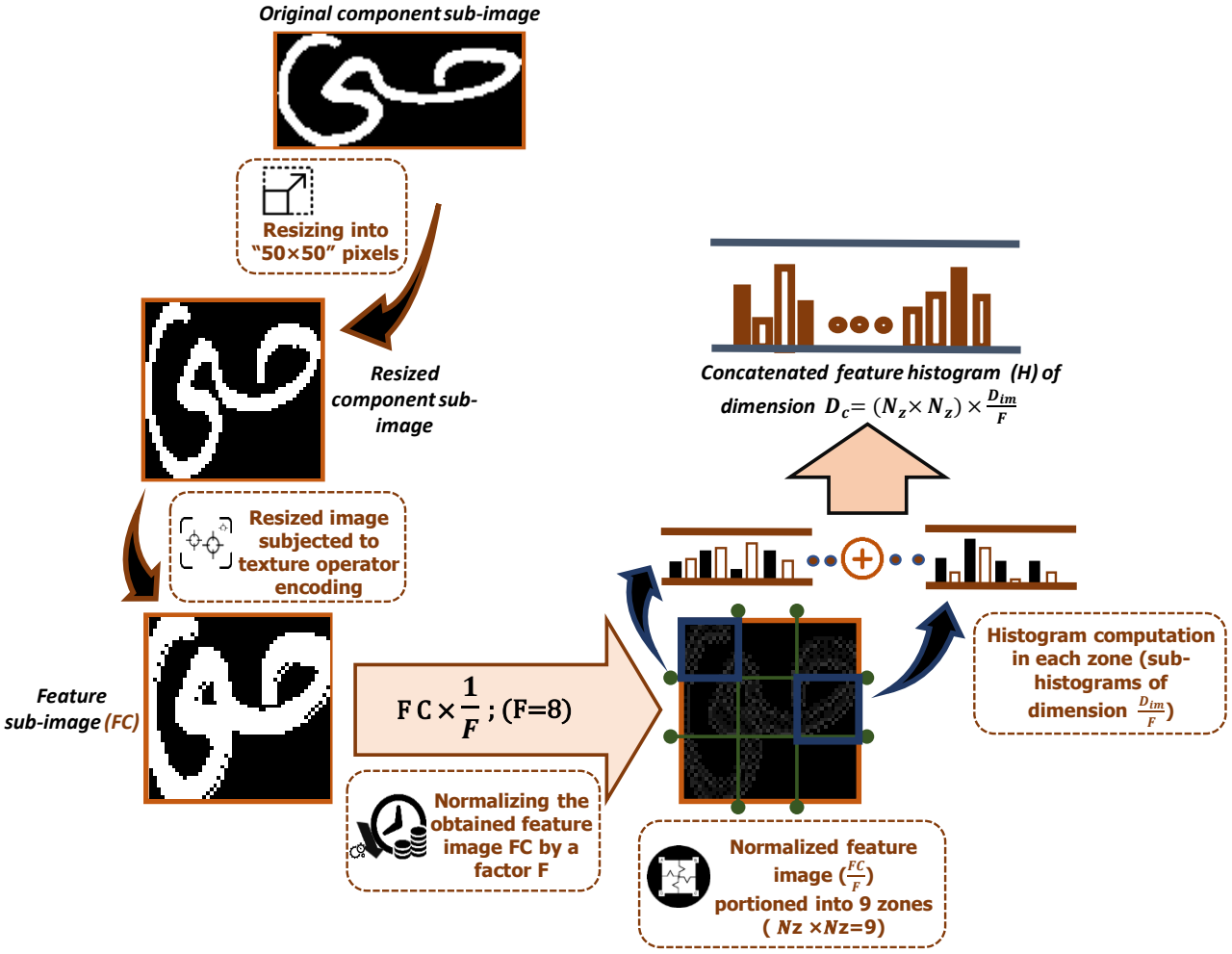


Figure 3.8: The proposed feature extraction process.

- Step 6: Each zone $Z^{(j,w_m)}_{i,i=1,..,N_z \times N_z}$ is represented by its feature histogram $h_i^{(j,w_m)}$ using Eq. 3.13.
- Step 7: Each connected component $C^j_{w_m}$ is characterized by its concatenated histogram $H^{(j,w_m)}$ using Eq. 3.15.
- Step 8: Finally, the set of H-histograms $\{H^{(j,w_m)}, j = 1, ..., N_{w_m}\}$ represents the input handwriting sample of the classifier with the writer label w_m.

Note that the performance of the system depends substantially on both the dimensionality reduction factor F and the number of zones $N_z \times N_z$. Indeed, different values of N_z and F would result in a different representation of the LBP, LPQ, and LTP histograms (H). A comprehensive experiment to investigate the system performance as a function of these parameters is addressed in Chapter 5 (Section 5.3.2.1).

In summary, given LBP, LPQ, and LTP texture descriptors, the writing sample S_{w_m} (docu-

ment or set of word/text line images) with writer label w_m is characterized by a set $H_{S_{w_m}}$ of feature histograms computed from all connected components extracted from it:

$$H_{S_{w_m}} = \{H^{(j,w_m)}, 1 \leq j \leq N_{w_m}\} \tag{3.16}$$

where $N_{w_m} = card(S_{w_m})$ is the number of connected components in the sample S_{w_m} and $H^{(j,w_m)}$ is the feature histogram corresponding to the jth component in the writing sample S_{w_m}.

3.2.2.3/ CROSS MULTI-SCALE LOCALLY ENCODED GRADIENT PATTERNS DESCRIPTOR

We build an effective feature descriptor referred to as Cross multi-scale Locally encoded Gradient Patterns (CLGP). The proposed CLGP feature descriptor captures the texture information of the writing image using transformation feature construction and encodes the obtained texture codes in multiple scales via the Histograms of Oriented Gradients (HOG) operator within non-overlapping blocks. The distribution of local intensity gradients within these non-overlapping blocks forms the final CLGP feature histogram. CLGP is insensitive to noise by using low-order Gaussian derivative filters and a global averaging operator in the scalar quantization step.

As shown in Figure 3.9, the feature extraction procedure for computing the final CLGP feature histogram H is outlined in the following steps:

Step 1: Extremum responses computation (spatial filtering). The input connected components (gray-scale sub-images) are first resized to the same uniform window size of 80×80 pixels. Note that we use the same segmentation step in Section 3.2.1 with a slight change in the overall algorithm to extract gray-scale connected components. In Algorithm 1 (line 4), we only change the binary sample S_b with the original sample S to be segmented into connected component sub-images C^i in gray-scale format. Each resized component sub-image is convolved with a family of Gaussian derivative filters (Freeman et al. (1991)) (up to second order) to compute extremum value responses (maximum and minimum) at multiple scales. The main goal of this step is to capture useful information contained in the first and second-order differential structures at a range of scales. Based on the two-dimensional circularly symmetric Gaussian function defined in Eq. 3.17, we compute the first and second Gaussian derivatives at an arbitrary orientation θ, as given in Eqs. 3.18 and 3.19.

$$G(x,\ y;\ \sigma) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x^2\ +\ y^2}{2\sigma^2}\right) \tag{3.17}$$

$$G_1{}^{\theta} = cos(\theta)G_x \ + \ sin(\theta)G_y \tag{3.18}$$

$$G_2{}^{\theta} = cos^2(\theta)\,G_{xx} - sin(2\theta)\,G_{xy} + sin^2(\theta)\,G_{yy} \tag{3.19}$$

where σ is the scale or standard deviation. G_x and G_{xx} are respectively the scale-normalized first and second derivatives of G along the x-axis, and analogously for G_y , G_{xy} and G_{yy}. For each connected sub-image I, the first and second order image derivatives are defined by: $L_x = G_x * I$, $L_y = G_y * I$,$L_{xx} = G_{xx} * I$, $L_{xy} = G_{xy} * I$, $L_{yy} = G_{yy} * I$, where $*$ is the convolution operator. Formally, the responses of the first and second Gaussian derivative filters at orientation θ (Freeman et al. (1991); Zhang et al. (2013)) are given as follows:

$$\begin{aligned} I_1^{\theta} = G_1^{\theta} * I &= cos(\theta)\ L_x + sin(\theta)L_y \\ &= \sqrt{L_x^2 + L_y^2}\,sin(\theta + \phi) \end{aligned} \tag{3.20}$$

where $\phi = arctan\left(\frac{L_x}{L_y}\right)$ and

$$\begin{aligned} I_2^{\theta} = G_2^{\theta} * I &= cos^2(\theta)\ L_{xx} - sin(2\theta)L_{xy} + sin^2(\theta)L_{yy} \\ &= \frac{1}{2}\left(L_{xx} + L_{yy} + \sqrt{\left(L_{xx} - L_{yy}\right)^2 + 4L_{xy}^2}cos(2\theta - \psi)\right) \end{aligned} \tag{3.21}$$

where $\psi = arctan\left(\frac{2L_{xy}}{L_{yy} - L_{xx}}\right)$. The extremum response values of I_1^{θ} and I_2^{θ} over all θ are computed as follows:

$$I_{1max}^{\theta} = \sqrt{L_x^2 + L_y^2} \tag{3.22}$$

$$I_{2max}^{\theta} = \frac{1}{2}\left(L_{xx} + \ L_{yy} + \sqrt{\left(L_{xx} - L_{yy}\right)^2 + 4L_{xy}^2}\right) \tag{3.23}$$

$$I_{2min}^{\theta} = \frac{1}{2}\left(L_{xx} + \ L_{yy} - \sqrt{\left(L_{xx} - L_{yy}\right)^2 + 4L_{xy}^2}\right) \tag{3.24}$$

Extreme value responses are computed on N_σ scales. As in (Zhang et al. (2013); Song et al. (2015)), the number of scales N_σ is experimentally set to $N_\sigma = 3$: $\sigma_1 = 1$, $\sigma_2 = 2$,

and $\sigma_3 = 4$.

Step 2: Transorm feature construction. Linear and nonlinear operators are applied to the previously obtained extreme value responses I^{θ}_{1max}, I^{θ}_{2max}, and I^{θ}_{2min} to construct a compact, rotation-invariant, yet discriminative set of transformation features, denoted as $F = \{g, d, s, r\}$ ($\{g, d\}$ is constructed with linear combinations of the extreme value responses, while $\{s, r\}$ is constructed with nonlinear ones). The transformation feature g, referred to as the gradient magnitude, is the maximum response of the first directional Gaussian derivative filter, i.e., $g = I^{\theta}_{1max} = \sqrt{L_x^2 + L_y^2}$. The second transformation feature d, i.e., the extrema difference of the maximum and minimum responses of the second directional Gaussian derivative filter is calculated by:

$$d = I^{\theta}_{2max} - I^{\theta}_{2min} = \sqrt{\left(L_{xx} - L_{yy}\right)^2 + 4L_{xy}^2} \tag{3.25}$$

The feature set F includes other quantitative measures of the second-order differential structure defined by the shape index s:

$$\begin{aligned} s &= \frac{1}{2} - \frac{1}{\pi} arctan\left(-\frac{I^{\theta}_{2max} + I^{\theta}_{2min}}{I^{\theta}_{2max} - I^{\theta}_{2min}}\right) \\ &= \frac{1}{2} - \frac{1}{\pi} arctan\left(\frac{-L_{xx} - L_{yy}}{\sqrt{\left(L_{xx} - L_{yy}\right)^2 + 4L_{xy}^2}}\right) \end{aligned} \tag{3.26}$$

The correlation information of first and second order differential structures is characterized by means of the mixed extremal ratio r, which is defined as follows:

$$\begin{aligned} r &= \frac{2}{\pi} arctan\left(\frac{d}{g}\right) = \frac{2}{\pi} arctan\left(\frac{I^{\theta}_{2max} - I^{\theta}_{2min}}{I^{\theta}_{1max}}\right) \\ &= \frac{2}{\pi} arctan\left(\sqrt{\frac{\left(L_{xx} - L_{yy}\right)^2 + 4L_{xy}^2}{L_x^2 + L_y^2}}\right) \end{aligned} \tag{3.27}$$

Step 3: Quantization and cross-scale joint coding. With the obtained transformation features, the scalar quantization step aims to design a discriminative and computationally efficient quantizer to quantize the feature set F into discrete texture codes. To this end, two types of scalar quantization by simple binary or multilevel thresholding are designed. For the feature subset $\{g, d\}$, we perform a mean-based binary ratio quantizer $Q_1(.)$:

$$y = Q_1(x) = \begin{cases} 0, & if \ \frac{x}{m_x} > k \\ 1, & otherwise \end{cases} \tag{3.28}$$

where $x \in \{g, d\}$, m_x is the mean value of the transform feature map of x and k is a tuning parameter. The transform feature values of $\{s, r\}$ are in the range of $[0, 1]$ (cf. Eqs. 3.26 and 3.27). Therefore, for the feature subset $\{s, r\}$, a simple uniform quantization $Q_2(.)$ is used (cf. Eq. 3.29):

$$y = Q_2(x) = \begin{cases} 0, & x \in [0,\ \Delta] \\ 1, & x \in [\Delta,\ 2\Delta] \\ \quad \cdots & \\ (L\_x) - 1, & x \in [((L\_x) - 1)\Delta,\ 1] \end{cases} \tag{3.29}$$

where $x \in \{s, r\}$, L_x is the quantization level (L_s and L_r for the transform features s and r, respectively), and $\Delta = 1/L\_x$ is the quantization step. In our experiments, we empirically set the parameters related to scalar quantization as $k = 2$, $L\_s = 3$, and $L\_r = 5$, as proposed in (Song et al. (2017)).

In the next step, the cross-scale joint coding is performed to aggregate the generated discrete pixel-wise codes obtained by scalar quantization into a compact image feature representation. This is performed by jointly encoding the texture codes across all scales, i.e., constructing multiple feature code maps across multiple scales (cf. Figure 3.9). The first feature code map, referred to as Adjacent-Scale Coding (ASC), is obtained by jointly encoding the quantized texture codes of the transform feature subset $\{g, d, s\}$ across two adjacent scales (e.g., $(\sigma_1,\ \sigma_2)$, $(\sigma_2,\ \sigma_3)$, etc). For the adjacent scale pair $(\sigma_i,\ \sigma_{i+1})$ $(i = 1, 2, \ldots,\ N_\sigma - 1)$, the feature code map ASC value of the pixel (x, y) in the connected sub-image I is composed as follows:

$$\begin{aligned} c_i(x, y) = & \sum_{j=1}^{2} (L\_s)^{j-1} y_s\left(x, y;\ \sigma_{i+j-1}\right) + \\ & (L\_s)^2 \left[\sum_{j=1}^{2} (L\_d)^{j-1} y_d\left(x, y;\ \sigma_{i+j-1}\right)\right] + \\ & (L\_s)^2 (L\_d)^2 \left[\sum_{j=1}^{2} (L\_g)^{j-1} y_g\left(x, y;\ \sigma_{i+j-1}\right)\right] \end{aligned} \tag{3.30}$$

$y_s\left(x, y;\ \sigma_{i+j-1}\right)$, $y_d\left(x, y;\ \sigma_{i+j-1}\right)$, and $y_g\left(x, y;\ \sigma_{i+j-1}\right)$ are the quantized texture codes of the transform features s, d, and g at scale σ_i. L_s, L_d and L_g are the quantization levels for features s, d and g, respectively. We empirically set $L\_s = 3$ and $L\_d = L\_g = 2$. Based on the ASC encoding, two feature code maps are generated: c_1 (i.e., $ASC1$) at two adjacent scales $(\sigma_1,\ \sigma_2) = (1, 2)$, and c_2 (i.e., $ASC2$) at $(\sigma_2,\ \sigma_3) = (2, 4)$ (cf. Figure 3.9). Next, we perform another joint scale encoding, referred to as Full-Scale Coding (FSC), where the transform feature subset $\{r\}$ is jointly encoded over all N_σ scales $(\sigma_1, ..., \sigma_{N_\sigma})$. The feature code map FSC value of the pixel (x, y) in the connected sub-image I is calculated

as follows:

$$c_{N_\sigma}(x,y) = \sum_{j=1}^{N_\sigma} (L\_r)^{j-1} y_r\left(x, y;\ \sigma_j\right) \tag{3.31}$$

$y_r\left(x, y;\ \sigma_j\right)$ is the quantized texture codes of the transform features $\{r\}$ at the scale σ_j. L_r is the quantization level for the features $\{r\}$. The third feature code map c_3 (i.e., FSC) is computed at three adjacent scales $(\sigma_1, \sigma_2, \sigma_3) = (1, 2, 4)$.

Step 4: Code maps encoding via HOG operator. As shown in Figure 3.9, the proposed technique is to portion each obtained feature code map into small spatial regions (blocks) and encode each of them via the HOG operator (Dalal and Triggs (2005)) to generate the corresponding HOG histogram. Concatenating the histograms of all the regions forms the normalized feature vector related to a particular feature code map. After that, we further concatenate all the obtained feature vectors of all feature code maps to obtain the final CLGP (Cross multi-scale Locally encoded Gradient Patterns) histogram-based feature representation.

In summary, three feature code maps are obtained, i.e., $ASC1$, $ASC2$, and FSC, each of which is scanned from top to bottom and left to right and spatially divided into small uniform N_b non-overlapping blocks. The histogram HOG is extracted from each block $B_{t,t=1,\ldots,N_b}$ (cf. Figure 3.9). Given a feature code map $c_{m,m=1,\ldots,N_\sigma}$ partitioned into N_b blocks, we compute their respective HOG feature histograms using gradient detectors. Formally, each pixel of each block B_t^m is convolved with the simple convolution kernel defined as follows:

$$Gr_x = B_t^m(x+1, y) - B_t^m(x-1, y) \tag{3.32}$$

$$Gr_y = B_t^m(x, y+1) - B_t^m(x, y-1) \tag{3.33}$$

Gr_x and Gr_y are the horizontal and vertical components of the gradients, respectively. HOG descriptor is computed from the occurrence of oriented gradients, i.e., magnitude and direction, within rectangular non-overlapping cells (R- HOG) of the feature block B_t^m. The gradient orientation θ and magnitude M are calculated as follows:

$$M(x,y) = \sqrt{Gr_x^2 + Gr_y^2} \tag{3.34}$$

$$\theta(x,y) = tan^{-1}\frac{Gr_y}{Gr_x} \tag{3.35}$$

Each block $B^m_{t,t=1,...,N_b}$ in the feature code map $c_{m,m=1,...,N_\sigma}$ contains N_{cell} non-overlapping cells and 9 bin histograms per cell. These histograms are then concatenated to construct a $(N_{cell} \times 9)$-dimensional feature vector $V^m_{t,t=1,..,N_b}$, which is then normalized by applying $L2$ block normalization (Lee et al. (2013)) as follows:

$$h^m_t = \frac{V^m_t}{\sqrt{\left\| V^m_t \right\|^2 + \varepsilon}} \tag{3.36}$$

h^m_t is the normalized $(N_{cell} \times 9)$-dimensional feature vector and ε is a small value close to zero.

Subsequently, each feature code map c_m is henceforth characterized by its concatenated histogram H_m using Eq. 3.37:

$$H_m = \prod_{t=1}^{N_b} h^m_t \tag{3.37}$$

where $\prod$ is the concatenation operator. The three $(N_\sigma = 3)$ obtained normalized feature vectors $H_{m,m=1,..,(N_\sigma=3)}$ representing the three feature code maps $c_{m,m=1,...,(N_\sigma=3)}$ are further concatenated to obtain the final CLGP histogram-based feature representation $H = [H_1, H_2, H_3]$, which is a $(N_{cell} \times 9 \times N_b \times (N_\sigma = 3))$-dimensional image feature descriptor.

Note that the performance of the system depends significantly on the number of blocks and cells (N_b and N_{cell}, respectively). Indeed, the optimal settings of these parameters for each tested database are determined by extensive experiments discussed in Chapter 5 (cf. Section 5.3.3.2).

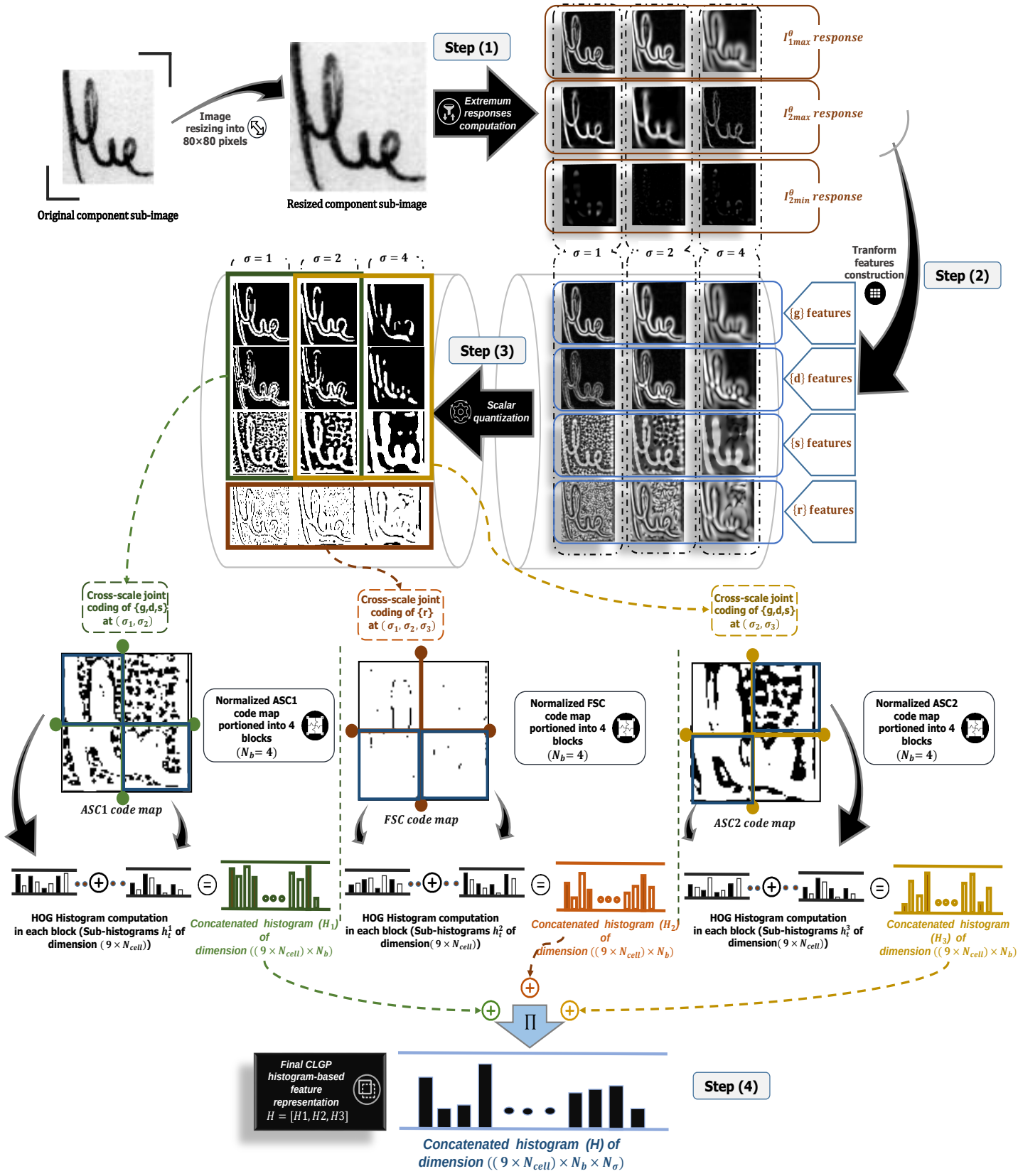


Figure 3.9: The pipeline of CLGP feature extraction method.

3.2.2.4/ LOCAL GRADIENT FULL-SCALE TRANSFORM PATTERNS DESCRIPTOR

An effective feature descriptor for writing characterization called Local gradient full- Scale Transform Patterns (LSTP) is proposed. Inspired by our previous CLGP descriptor, especially the part that uses extreme value filtering and linear/nonlinear transformations, our LSTP feature descriptor explicitly captures and encodes the writing pattern over non-overlapping blocks. Indeed, it characterizes the writing information using a set of transformation features across multiple scale-spaces and encodes the resulting transformation feature map (obtained by intersecting all transformation features) in small observation blocks via the Histogram of Oriented Gradients operator (Dalal and Triggs (2005)). The local intensity gradients computed within these blocks form the final LSTP feature histogram. The use of low-order Gaussian derivative filters makes LSTP insensitive to noise.

As shown in Figure 3.10, the steps of the LSTP feature descriptor are explained in detail in the following:

Spatial Gaussian derivative filtering. Input connected components (gray-scale sub-images) are first subjected to directional Gaussian derivative filtering (DGD) (Freeman et al. (1991)). Using the circularly symmetric 2-D Gaussian function defined in Eq. 3.38, the first and second DGD filters are computed as follows (cf. Eqs. 3.39 and 3.40):

$$G(x,\ y;\ \sigma) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x^2\ +\ y^2}{2\sigma^2}\right) \tag{3.38}$$

$$G_1^{\ \theta} = cos(\theta)G_x\ +\ sin(\theta)G_y \tag{3.39}$$

$$G_2^{\ \theta} = cos^2(\theta)\, G_{xx} - sin(2\theta)\, G_{xy} + sin^2(\theta)\, G_{yy} \tag{3.40}$$

$G(x,\ y;\ \sigma)$ is defined and written in Cartesian coordinates x and y. θ is an arbitrary orientation. σ is the standard deviation. In the (x)- axis, G_x (first) and G_{xx} (second) are the scale derivatives of G, and G_y, G_{yy} and G_{xy} over the (y)- axis. For each connected component sub-image I, the image derivatives are defined by: $L_x = G_x * I$, $L_y = G_y * I$, $L_{xx} = G_{xx} * I$, $L_{xy} = G_{xy} * I$, $L_{yy} = G_{yy} * I$, where $*$ is the convolution operator. The first and second Gaussian responses at orientation θ (Freeman et al. (1991); Zhang et al. (2013)) are formally defined as follows:

$$\begin{aligned} I_1^{\theta} = G_1^{\theta} * I &= cos(\theta)\ L_x + sin(\theta)L_y \\ &= \sqrt{L_x^2 + L_y^2}\, sin(\theta + \phi) \end{aligned} \tag{3.41}$$

where $\phi = arctan\left(\frac{L_x}{L_y}\right)$ and

$$\begin{aligned} I_2^\theta &= G_2^\theta * I = cos^2(\theta)\, L_{xx} - sin(2\theta)L_{xy} + sin^2(\theta)L_{yy} \\ &= \frac{1}{2}\left(L_{xx} + L_{yy} + \sqrt{\left(L_{xx} - L_{yy}\right)^2 + 4L_{xy}^2}cos\,(2\theta - \psi)\right) \end{aligned} \tag{3.42}$$

where $\psi = arctan\left(\frac{2L_{xy}}{L_{yy}-L_{xx}}\right)$. Extremum (maximum and minimum) response values of I_1^θ and I_2^θ over all θ are computed at N_σ scales (cf. Eqs 3.43, 3.44, and 3.45):

$$I_{1max}^\theta = \sqrt{L_x^2 + L_y^2} \tag{3.43}$$

$$I_{2max}^\theta = \frac{1}{2}\left(L_{xx} + L_{yy} + \sqrt{\left(L_{xx} - L_{yy}\right)^2 + 4L_{xy}^2}\right) \tag{3.44}$$

$$I_{2min}^\theta = \frac{1}{2}\left(L_{xx} + L_{yy} - \sqrt{\left(L_{xx} - L_{yy}\right)^2 + 4L_{xy}^2}\right) \tag{3.45}$$

Following (Varma and Zisserman (2009); Crosier and Griffin (2010); Zhang et al. (2013); Song et al. (2015)), the number of scales N_σ is experimentally set to $N_\sigma = 3$: $\sigma_1 = 1$, $\sigma_2 = 2$, and $\sigma_3 = 4$.

Cross-scale features. I_{1max}^θ, I_{2max}^θ and I_{2min}^θ are convolved with linear and nonlinear operators to construct a discriminative and compact set of transform features $F = \{g, d, s, r\}$. This step captures the local texture structures of the writing, and computes their correlation in the input component sub-image. The gradient magnitude g is the first transform feature, simply defined by: $g = I_{1max}^\theta = \sqrt{L_x^2 + L_y^2}$. The extrema difference feature d is calculated as:

$$d = I_{2max}^\theta - I_{2min}^\theta = \sqrt{\left(L_{xx} - L_{yy}\right)^2 + 4L_{xy}^2} \tag{3.46}$$

The feature s (the shape index) is given by:

$$\begin{aligned} s &= \frac{1}{2} - \frac{1}{\pi}arctan\left(-\frac{I_{2max}^\theta + I_{2min}^\theta}{I_{2max}^\theta - I_{2min}^\theta}\right) \\ &= \frac{1}{2} - \frac{1}{\pi}arctan\left(\frac{-L_{xx} - L_{yy}}{\sqrt{\left(L_{xx} - L_{yy}\right)^2 + 4L_{xy}^2}}\right) \end{aligned} \tag{3.47}$$

The fourth transform feature r (mixed extrema ratio) is computed as given in Eq 3.48:

$$\begin{aligned} r &= \frac{2}{\pi} arctan\left(\frac{d}{g}\right) = \frac{2}{\pi} arctan\left(\frac{I^{\theta}_{2max} - I^{\theta}_{2min}}{I^{\theta}_{1max}}\right) \\ &= \frac{2}{\pi} arctan\left(\sqrt{\frac{\left(L_{xx} - L_{yy}\right)^2 + 4L^2_{xy}}{L^2_x + L^2_y}}\right) \end{aligned} \tag{3.48}$$

Full-scale map encoding via HOG operator. This encoding step aims to construct a compact feature map over multiple scales. It performs a quantification of the generated transform features $F = \{g, d, s, r\}$ into a full-scale code map, i.e., a quantitative feature code map at $(\sigma_1, \sigma_2, \sigma_3) = (1, 2, 4)$. As shown in Figure 3.10, each transformation subset, viz. $\{g\}$, $\{d\}$, $\{s\}$ or $\{r\}$, comprises three feature sub-images at three scales σ_1, σ_2 and σ_3, respectively: $g = \{g_{\sigma_1}, g_{\sigma_2}, g_{\sigma_3}\}$, $d = \{d_{\sigma_1}, d_{\sigma_2}, d_{\sigma_3}\}$, $s = \{s_{\sigma_1}, s_{\sigma_2}, s_{\sigma_3}\}$, and $r = \{r_{\sigma_1}, r_{\sigma_2}, r_{\sigma_3}\}$. For each transform feature subset (g, d, s and r), the proposed technique is to jointly merge the three feature sub-images (computed at three different scales) into a holistic discriminative feature sub-image. Formally, using Eqs. 3.49, 3.50, 3.51, and 3.52, the Hadamard product is performed for each transformed feature subset, which takes three feature sub-images (matrices) of the same dimension and generates a different feature matrix (i.e., a holistic discriminative feature sub-image) where each element is the product of the elements of the original three feature matrices. This increases the ability of the algorithm to accurately characterise the writing content within each related component sub-image.

$$G = \left(g_{\sigma_1} \circ g_{\sigma_2} \circ g_{\sigma_3}\right)_{ij} = (g_{\sigma_1})_{ij} \bullet (g_{\sigma_2})_{ij} \bullet (g_{\sigma_3})_{ij} \tag{3.49}$$

$$D = (d_{\sigma_1} \circ d_{\sigma_2} \circ d_{\sigma_3})_{ij} = (d_{\sigma_1})_{ij} \bullet (d_{\sigma_2})_{ij} \bullet (d_{\sigma_3})_{ij} \tag{3.50}$$

$$S = (s_{\sigma_1} \circ s_{\sigma_2} \circ s_{\sigma_3})_{ij} = (s_{\sigma_1})_{ij} \bullet (s_{\sigma_2})_{ij} \bullet (s_{\sigma_3})_{ij} \tag{3.51}$$

$$R = (r_{\sigma_1} \circ r_{\sigma_2} \circ r_{\sigma_3})_{ij} = (r_{\sigma_1})_{ij} \bullet (r_{\sigma_2})_{ij} \bullet (r_{\sigma_3})_{ij} \tag{3.52}$$

Next, the four holistic feature matrices G, D, S and R are concatenated horizontally/vertically to form the full-scale code map FC (cf. Figure 3.10). Then, FC is partitioned into N_{bk} blocks $Bk_{t,t=1,..,N_{bk}}$ (without overlap) encoded via the HOG operator to generate their respective HOG histograms:

$$Grad_x = Bk_t(x+1, y) - Bk_t(x-1, y) \tag{3.53}$$

$$Grad_y = Bk_t(x, y+1) - Bk_t(x, y-1) \tag{3.54}$$

$Grad_x$ (horizontal) and $Grad_y$ (vertical) are the gradients. Oriented gradient occurrences, i.e., magnitude and direction computed within rectangular non-overlapping cells (R-HOG), define the HOG descriptor:

$$Mg(x, y) = \sqrt{Grad_x^2 + Grad_y^2} \tag{3.55}$$

$$\theta g(x, y) = tan^{-1} \frac{Grad_y}{Grad_x} \tag{3.56}$$

Mg and θg are the gradient magnitude and orientation. In our experimental study, N_c non-overlapping cells and 9 bin histograms per cell are set for each block $Bk_{t,t=1,...,N_{bk}}$ in the full-scale code map FC. The concatenation of these histograms constructs a $(N_c \times 9)$-dimensional feature histogram $h_{t,t=1,..,N_{bk}}$. The feature histogram $h_{t,t=1,..,N_{bk}}$ is normalized via the $L2$ block normalization function (Lee et al. (2013)):

$$H_t = \frac{h_t}{\sqrt{||h_t||^2 + \xi}}, t = 1, .., N_{bk} \tag{3.57}$$

H_t is the normalized $(N_c \times 9)$-dimensional feature histogram and ξ is a small value close to zero. Normalized histograms $H_{t,t=1,...,N_{bk}}$ of all blocks in the full-scale code map FC are further concatenated to form the final LSTP histogram-based feature representation, which is a $(N_c \times 9 \times N_{bk})$-dimensional image feature descriptor:

$$H_{LSTP} = \prod_{t=1}^{N_{bk}} H_t \tag{3.58}$$

$\prod$ is the concatenation operator. The optimal values for the number of blocks and cells (N_{bk} and N_c, respectively) are empirically determined for each tested handwritten database. The proposed LSTP algorithm for feature extraction is given in Algorithm 2.

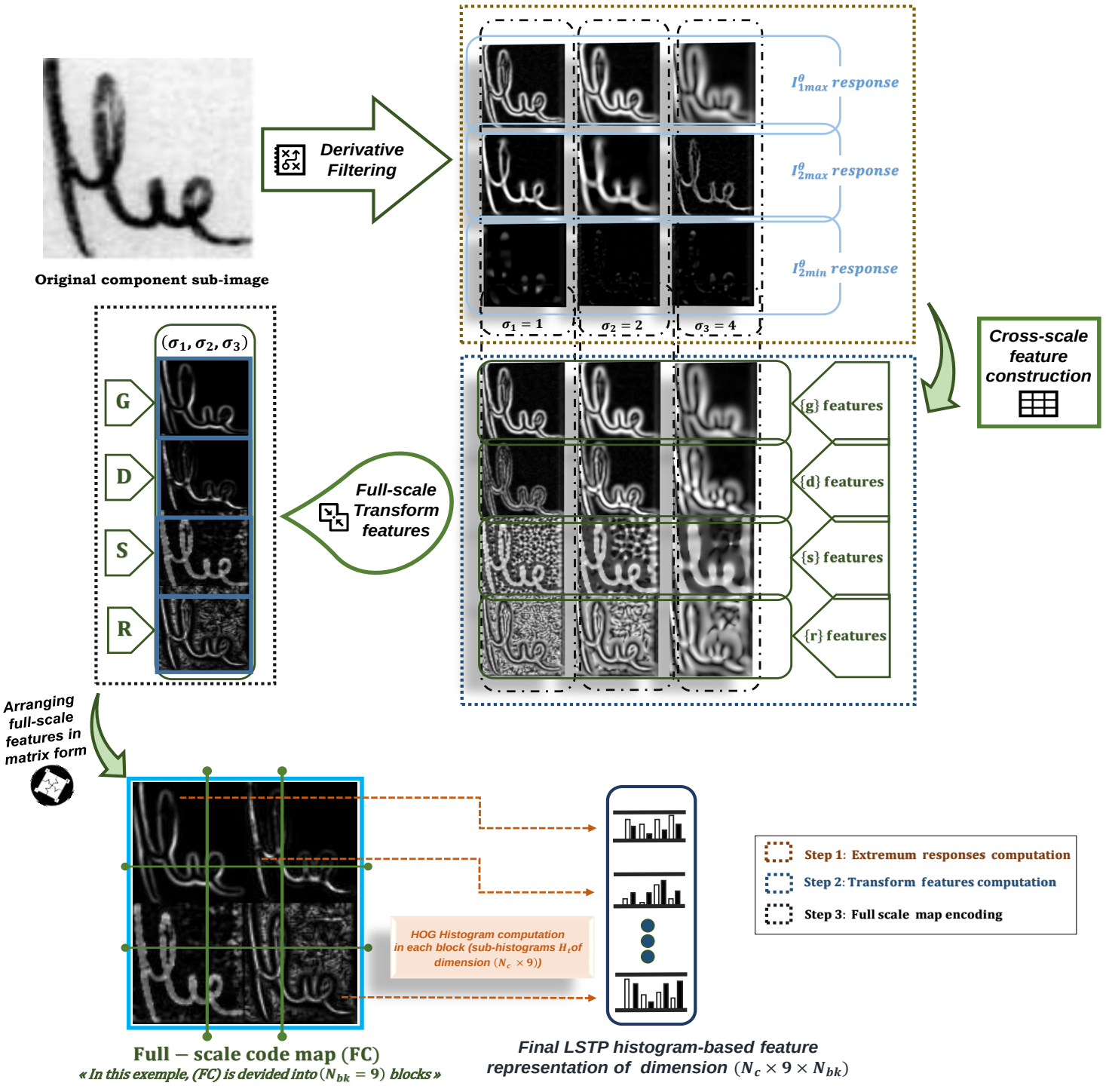


Figure 3.10: The proposed LSTP feature extraction method.

Algorithm 2: Proposed LSTP for feature extraction

Input : C^i, $i = 1, 2, \cdots, n$, where each element is an integer. // connected component sub-image number i.

Output: H_{LSTP} // LSTP feature vector histogram of the connected component number i.

$\mathbf{I} \leftarrow \mathbf{C^i}$;

/* Compute extremum (maximum and minimum) response values of I_1^θ and I_2^θ over all θ at three different scales $N_\sigma = 3$: $\sigma_1 = 1$, $\sigma_2 = 2$, and $\sigma_3 = 4$. */

$\mathbf{G}(\mathbf{x}, \mathbf{y}, \sigma)$; // Eq. 3.38; define the 2-D circularly symmetric Gaussian function.

$\mathbf{G_1^\theta}$; $\mathbf{G_2^\theta}$; // Eqs. 3.39 and 3.40; compute the first and second Gaussian derivatives G_1^θ and G_2^θ at orientation θ.

$\mathbf{I_1^\theta} = \mathbf{G_1^\theta} * \mathbf{I}$; $\mathbf{I_2^\theta} = \mathbf{G_2^\theta} * \mathbf{I}$; // Eqs. 3.41 and 3.42; compute the first and second Gaussian responses I_1^θ and I_2^θ at orientation θ.

$\mathbf{I_{1max}^\theta}$; $\mathbf{I_{2max}^\theta}$; $\mathbf{I_{2min}^\theta}$; // Eqs. 3.43, 3.44, and 3.45; get the maximum and minimum response values of I_1^θ and I_2^θ over all θ at three different scales $N_\sigma = 3$: $\sigma_1 = 1$, $\sigma_2 = 2$, and $\sigma_3 = 4$.

/* Construct cross-scale transform features denoted as $F = \{g, d, s, r\}$. */

$\mathbf{g} = \mathbf{I_{1max}^\theta}$; $\mathbf{d} = \mathbf{I_{2max}^\theta} - \mathbf{I_{2min}^\theta}$; $\mathbf{s}$; $\mathbf{r} = \frac{2}{\pi}\mathbf{arctan}\left(\frac{\mathbf{d}}{\mathbf{g}}\right)$; // Eqs. 3.46, 3.47, and 3.48; compute the gradient magnitude g, extrema difference d, shape index s, and mixed extrema ration r.

/* Compute the full-scale tranform features G, D, S, and R. */

$\mathbf{G} = \mathbf{g}_{\sigma_1} \circ \mathbf{g}_{\sigma_2} \circ \mathbf{g}_{\sigma_3}$; $\mathbf{D} = \mathbf{d}_{\sigma_1} \circ \mathbf{d}_{\sigma_2} \circ \mathbf{d}_{\sigma_3}$; $\mathbf{S} = \mathbf{s}_{\sigma_1} \circ \mathbf{s}_{\sigma_2} \circ \mathbf{s}_{\sigma_3}$; $\mathbf{R} = \mathbf{r}_{\sigma_1} \circ \mathbf{r}_{\sigma_2} \circ \mathbf{r}_{\sigma_3}$; // Eqs. 3.49, 3.50, 3.51 and 3.52;

/* Concatenate G, D, S, and R to form the full-scale code map FC. */

$\mathbf{c1} = \mathbf{concatenatefunction}(\mathbf{G}, \mathbf{D}, \mathbf{axis} = \mathbf{2})$; // horizental concatenation of G and D. $\mathbf{c2} = \mathbf{concatenatefunction}(\mathbf{S}, \mathbf{R}, \mathbf{axis} = \mathbf{2})$; // horizental concatenation of S and R. $\mathbf{FC} = \mathbf{concatenatefunction}(\mathbf{c1}, \mathbf{c2}, \mathbf{axis} = \mathbf{1})$; // vertical concatenation of $c1$ and $c2$.

/* Compute the final LSTP feature histogram. */

$\mathbf{Bk} = \mathbf{getblocksfunction}(\mathbf{FC}, \mathbf{N_{bk}})$; // devide the full-scale code map FC into N_{bk} non-overlapping blocks $FC = \cup_{t=1}^{N_{bk}} Bk_t$ (Bk_t is the block number t).

$\mathbf{H_{LSTP}} = [\ ]$;

for $t \leftarrow 1$ **to** N_{bk} **do**

  $\mathbf{H_t} = \mathbf{HOGfunction}(\mathbf{Bk_t})$; // Eq. 3.57; compute the HOG histogram of the block Bk_t.

  $\mathbf{H_{LSTP}} = \mathbf{concatenatefunction}(\mathbf{H_{LSTP}}, \mathbf{H_t}, \mathbf{axis} = \mathbf{2})$; // Eq. 3.58; output; HOG histograms of all blocks in FC are concatentated to form the feature vector histogram H_{LSTP}.

end for

return H_{LSTP}

3.2.3/ CLASSIFICATION PROCESS

After extracting feature histograms from all the connected components in the writing, we perform the classification process, i.e., identifying the writer using the Nearest Neighbor classifier. Keeping the same classification process, we empirically evaluated various distance metrics to study the system performance. These include Euclidean, correlation, Bhattacharyya, cosine, and Hamming distance metrics. The experiments have shown that Hamming distance, which is commonly used and considered as an efficient distance in writer identification (Hannad et al. (2016); Schomaker and Bulacu (2004), etc.), performs better than the other metrics.

To perform the identification process, we set up a comparison mechanism between handwritten samples (cf. Figure 3.11). It consists of comparing the Hamming distances of the respective feature histograms of one sample from the test set and the other from the training set: (1) test sample (unseen) to be identified, denoted as S_{w_x}, where w_x is the label of the unknown writer in the test set; (2) training sample, denoted as $S_{w_{rf}}$, produced by a known writer labeled by w_{rf} in the reference base B_{rf}. A set of connected sub-images $C_{w_x}^{j\ (j=1,.,N_{w_x})}$ is extracted from the test sample S_{w_x} characterized by their corresponding feature histograms $H_{S_{w_x}} = \{H^{C_{w_x}^j}_{\zeta(C_{w_x}^j)>\rho}, 1 \leq j \leq N_{w_x}\}$. $\zeta(C_{w_x}^j)$ is the number of pixels in the component $C_{w_x}^j$. ρ is a small threshold used to remove certain unwanted diacritics and accidental writing traces that represent a very small fraction of writing pixels with an area less than ρ pixels ($\rho << 50 \times 50$). $N_{w_x} = card(S_{w_x})$ is the number of connected component sub-images cropped from the test sample S_{w_x}. In the same way, the connected component sub-images $C_{w_{rf}}^{j\ (j=1,.,N_{w_{rf}})}$, which form the training sample $S_{w_{rf}}$, are represented by their feature histograms $H_{S_{w_{rf}}} = \{H^{C_{w_{rf}}^j}_{\zeta(C_{w_{rf}}^j)>\rho}, 1 \leq j \leq N_{w_{rf}}\}$. $N_{w_{rf}} = card(S_{w_{rf}})$ is the number of connected component sub-images in the training sample $S_{w_{rf}}$.

To summarise, the classification process (cf. Algorithm 3) for identifying the query sample S_{w_x} (based on distance comparison with a training sample $S_{w_{rf}}$) is briefly outlined as follows: (i) Hamming distances are computed between each connected component $C_{w_x}^j$ in the query sample S_{w_x} and all components $C_{w_{rf}}^{k\ (k=1,.,N_{w_{rf}})}$ in the training set $S_{w_{rf}}$, (ii) the component in the training set that has the smallest metric value is the one that matches the test component $C_{w_x}^j$ in the query sample S_{w_x}, and (iii) the final dissimilarity measure $DIS(S_{w_x}, S_{w_{rf}})$ between the unknown sample S_{w_x} and the (known) training sample $S_{w_{rf}}$ is defined as follows:

$$DIS(S_{w_x}, S_{w_{rf}}) = \frac{1}{N_{w_x}} \sum_{j=1}^{N_{w_x}} min\{\eta\left(C_{w_x}^j, C_{w_{rf}}^1\right), \eta\left(C_{w_x}^j, C_{w_{rf}}^2\right), \ldots, \eta\left(C_{w_x}^j, C_{w_{rf}}^{N_{w_{rf}}}\right)\} \tag{3.59}$$

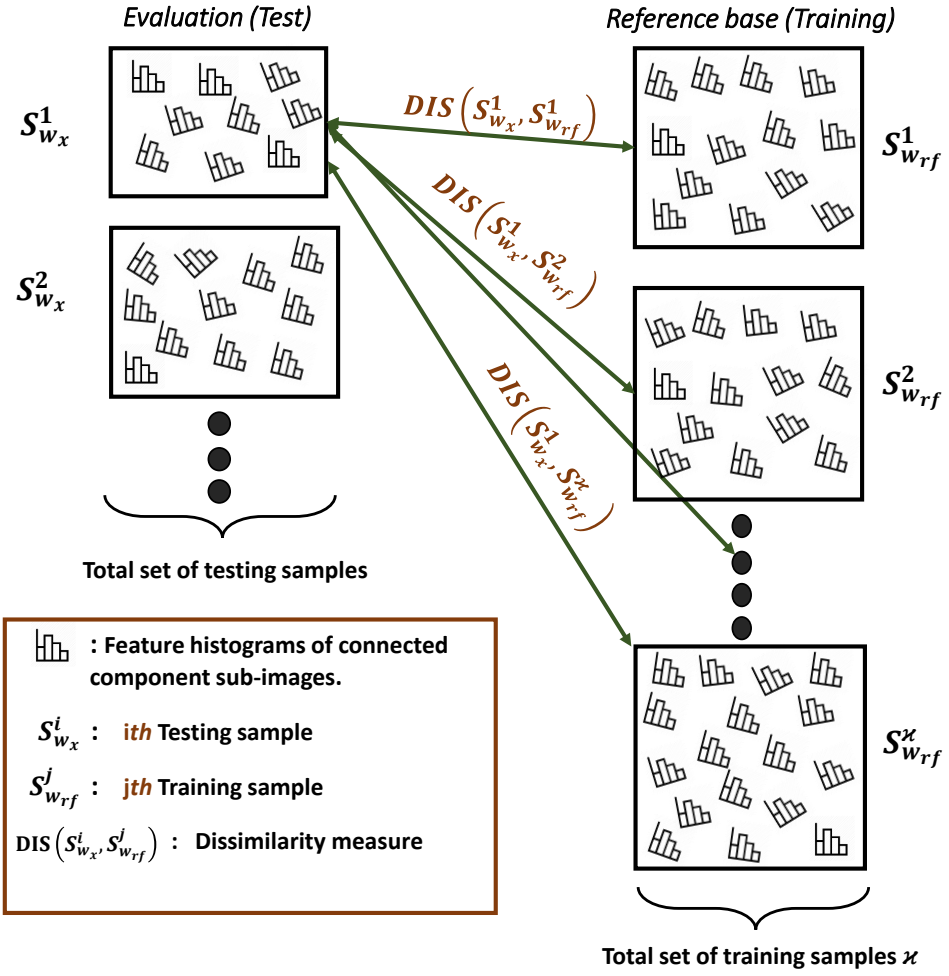


Figure 3.11: An overview of the classification process.

where $\eta\left(C^j_{w_x}, C^k_{w_{rf}}\right)$ is the Hamming distance between the connected component number j (i.e., $C^j_{w_x}$) in the test document S_{w_x} and the connected component number k (i.e., $C^k_{w_{rf}}$) in the training document $S_{w_{rf}}$. The Hamming distance function $\eta(\cdot,\cdot)$ is given in the following:

$$\eta\left(C^j_{w_x}, C^k_{w_{rf}}\right) = \sum_{n=1}^{D_{im}} |H^{C^j_{w_x}}(n) - H^{C^j_{w_{rf}}}(n)| \tag{3.60}$$

$H^{C^j_{w_x}}$ is the feature histogram of the connected component $C^j_{w_x}$, $H^{C^j_{w_{rf}}}$ is the feature histogram of the connected component $C^k_{w_{rf}}$, and D_{im} is the feature histogram dimension. All dissimilarities between the unseen sample S_{w_x} and all the training ones $S_{w_{rf}}, rf = 1, ..., \varkappa$ are arranged in a hit list ($\varkappa$ is the number of handwritten samples in the training base B_{rf}). As final classification decision, the writer of the query sample S_{w_x} is then identified as the writer of the sample in training set B_{rf}, which reports the minimum dissimilarity:

$$Writer\,(S_{w_x}) = argmin\{DIS\,(S_{w_x}, S_{w_1}), \ldots, \\ , \ldots, DIS\,(S_{w_x}, S_{w_\varkappa})\} \tag{3.61}$$

Algorithm 3: The classification process

Input : S_{w_x}, $[S_{w_{rf}}]$ // S_{w_x} is the query handwriting sample to be identfied; $[S_{w_{rf}}]$ list is the training samples, where $rf = 1, ..., \varkappa$ ($\varkappa$ is the number of handwritten samples in the training base B_{rf}).

Output: $Writer(S_{w_x})$ // writerID of the query sample S_{w_x}.

/* Parameters definition. */

// $N_{w_x} = card(S_{w_x})$ is the number of connected components of S_{w_x}; $N_{w_{rf}} = card(S_{w_{rf}})$ is the number of connected components of $S_{w_{rf}}$; $C^i_{w_x}$ is the connected component number i charecterized by its feature histogram $H^{C^i_{w_x}}$ in the query sample S_{w_x}; $C^j_{w_{rf}}$ is the connected component number j charecterized by its feature histogram $H^{C^j_{w_{rf}}}$ in the training sample $S_{w_{rf}}$;

/* Compute the dissimilarity between the query sample S_{w_x} and the training sample $S_{w_{rf}}$. */

1 **for** $i \leftarrow 1$ **to** N_{w_x} **do**

2 **for** $j \leftarrow 1$ **to** $N_{w_{rf}}$ **do**

3 **tmpdis(j)** = **hammingdisfunction**($\mathbf{C^i_{w_x}}, \mathbf{C^j_{w_{rf}}}$); // Eq.3.60; compute Hamming distance between $H^{C^i_{w_x}}$ and $H^{C^j_{w_{rf}}}$.

4 **end for**

5 **mindistances(i)** = **minfunction(tmpdis)**; // compute the minimum Hamming distance between $C^i_{w_x}$ and all $[C^j_{w_{rf}}]$ of the training sample $S_{w_{rf}}$.

6 **end for**

7 **diss**($\mathbf{S_{w_x}}, \mathbf{S_{w_{rf}}}$) = $\frac{1}{\mathbf{N_{w_x}}} \sum_{\mathbf{i=1}}^{\mathbf{N_{w_x}}}$ **mindistances(i)**; // Eq. 3.59; dissimilarity measure between S_{w_x} and $S_{w_{rf}}$.

/* Identify the writer of the query sample S_{w_x}. */

8 **for** $rf \leftarrow 1$ **to** $\varkappa$ **do**

9 **dissimilarities(rf)** = **diss**($\mathbf{S_{w_x}}, \mathbf{S_{w_{rf}}}$); // compute all dissimilarities between the query sample S_{w_x} and all the training ones $[S_{w_{rf}}]$.

10 **end for**

11 **Writer**($\mathbf{S_{w_x}}$) = **argminfunction(dissimilarities)** // Eq. 3.61;

12 **return** $Writer(S_{w_x})$

3.3/ CONCLUSION

In this chapter, we presented four texture-based systems for text-independent offline writer identification. The following pipeline was considered for the proposed systems: image preprocessing, feature extraction, and classification. We proposed and used a common step for all four texture-based systems for image preprocessing, namely, removing background noise and diacritics from the writing and segmenting the document image into connected component sub-images. These connected components are the input to the feature extraction step in all the proposed approaches.

The first system is based on the BWLBC method. It characterizes the variability of writing style in small binary blocks (of connected components) by computing the distribution of white pixels corresponding to the ink. A second contribution is to use LBP, LTP, LPQ handcrafted descriptors in a new and effective learning framework to capture texture features of small regions (zones) of interest in writing, followed by dimensionality reduction. The feature vector is then a set of sub-histograms computed sequentially within each zone of the connected component. This chapter also introduced two feature extraction methods, namely CLGP and LSTP. The CLGP model computes the distribution of local intensity gradients within small connected regions called cells (in connected components) across multiple feature code maps. The LSTP method is an extended variant of CLGP to extract more relevant texture information using the HOG operator. In the classification step, features are matched and compared using the nearest neighbor with an efficient strategy to identify unseen documents.

Deep learning methods are also explored in this thesis. In the next chapter, new and effective techniques based on Convolutional Neural Network (CNN) are presented to improve the writer identification task.

4

Deep Learning for Writer Identification

4.1/ Introduction

In deep learning, Convolutional Neural Network (CNN or ConvNet) is a class of deep neural networks that have recently emerged as a modern tool for large-scale pattern recognition problems and has found wide application in computer vision. With enough training data and good optimization, CNNs can provide an accurate solution for identifying writers from handwriting images. They also provide an easier way to obtain the desired features for characterizing individual writing style information. These features can be automatically learned by CNNs thanks to appropriate deep modeling and learning. To this end, we exploit the power of deep convolutional neural networks (CNNs) to improve the task of offline text-independent writer identification. Our first contribution consists of an effective deep learning-based framework called $WriterINet$. It includes three main steps: (1) image preprocessing to segment handwritten documents into word and connected component images; (2) deep feature extraction step. Since it is challenging to model the within-writer and between-writers variability, characterizing such features requires a well-designed and effective feature method using suitable classifiers. If these features are inappropriately extracted from the writing, it will affect the system in performing correct identification. Therefore, we introduce a multi-path deep CNN consisting of ResNet50 and DenseNet201 networks. Both models are trained and fine-tuned with different input data, i.e., words and connected components; (3) writer classification process using a conceptually simple and effective artificial neural network (1D- ANN). We define a custom 1D artificial neural network to classify CNN features (1D- ANN is initially used in signal processing for 1D data);

To further investigate the performance of writer identification using deep learning, we design an efficient CNN model referred to as $DeepWINet$. The proposed model is used and evaluated in two different ways. In the first scenario, the CNN activation features of

$DeepWINet$ computed from the connected components of the writing are fed to a nearest-neighbor classifier for writer identification. In the second scenario, $DeepWINet$ is evaluated as an end-to-end CNN network trained on connected components of the writing. Then, a score-averaging component-decision combiner is performed to average the predicted scores for writer identification.

The present work also contributes to the solution of historical document retrieval based on writer identification. We participated in the ICFHR2020 competition on image retrieval for historical handwritten fragments. The competition's goal is to retrieve all similar fragment images that belong to the same writer ID and find all fragment images segmented from the same page ID. Our submitted approach consists of two CNN streams trained and fine-tuned with different input data (fragment images from the ICFHR2020 training set). The average pooling layer of the two CNN branches is used to extract learned deep features of the test fragment images. As a similarity measure, the distance computation is performed using the Ch-square metric. In this competition, four approaches are presented by participants from different universities, including University Bourgogne Franche-Comte (UBFC), University of Groningen (RUG), and Tebessa University (ULT). Our system won first place in Task 2 (Page Retrieval) and second place in Task 1 (Writer Retrieval).

This chapter explains our different CNN-based approaches for writer identification, introduces the ICFHR2020 competition and the tested database and presents our proposed method along with a brief description of the other submitted methods.

4.2/ WRITERINET: A MULTI-PATH DEEP CNN FOR WRITER IDENTIFICATION

As shown in Figure 4.1, our proposed deep learning-based approach, called $WriterINet$, consists of three steps. First, the writing documents are fed to the preprocessing and segmentation phase, where each writing sample is decomposed into word images and connected component sub-images. Second, a powerful deep multipath CNN model is built for feature extraction, which consists of ResNet-50 (He et al. (2016a)) and DenseNet-201 (Huang et al. (2017)) models (pre-trained on the training set of IAM database). The model extracts discriminative features of the input word and its related component images. Third, a simple artificial neural network model (ANN) is developed to classify the obtained local features (corresponding to the words and connected components). In this way, for each test document (unseen sample), the trained model predicts all the similarity scores of its local deep feature vectors, based on which the writer identification is performed. Each step is described in detail in the following sections.

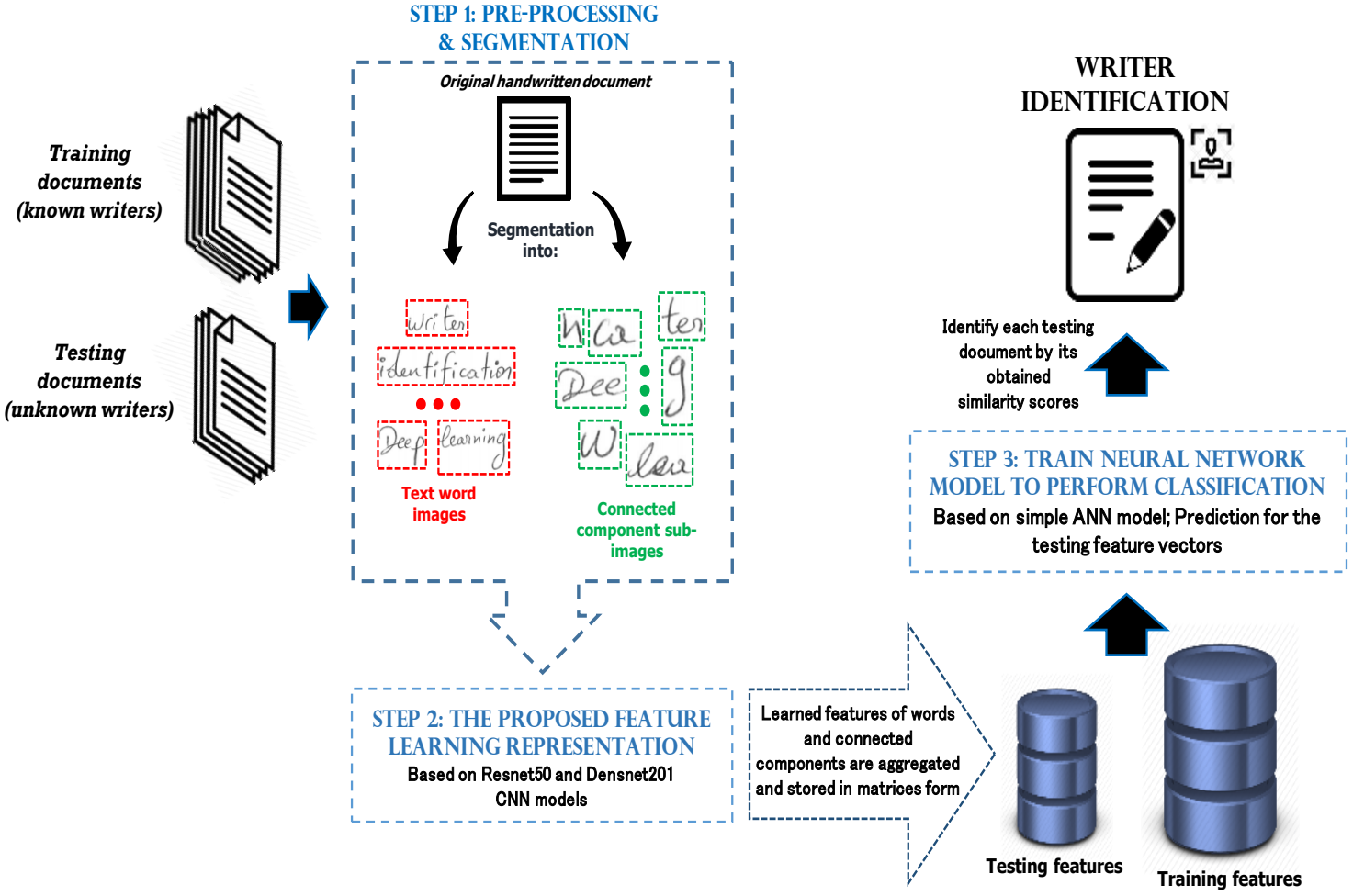


Figure 4.1: Design and structure of the proposed *WriterINet* approach.

4.2.1/ IMAGE SEGMENTATION METHOD

Image preprocessing and segmentation is an essential step in improving writer identification performance. As shown in Figure 4.1, each handwritten input sample is segmented: into word images and connected component sub-images. This step is critical because we need to isolate each word from the others and separate the individual connected components in writing. For word segmentation, we employ the same scale-space technique as in (Manmatha and Srimal (1999)), which is based on the use of blob analysis and Gaussian filters. This segmentation algorithm is fast, easy to implement [1] and, more importantly, ensures that one gets satisfactory segmentation results. As for the second segmentation, different regions of interest are extracted from the segmented word images, called connected components. This process is performed using a labeling function that assigns

[1] code: https://github.com/githubharald/WordSegmentation

a small bounding box around each connected component, i.e., determines all connected neighbor pixels that form a connected writing trace. Each truncated connected component is labeled with its specific class (writer). To quantify the writing content within each component, all connected neighbor pixels with non-significant information such as diacritics and random writing traces are removed. To discard these unwanted details, which are either generated during the image segmentation step or filled with the original background, a size-based thresholding procedure is applied to the connected components using a small threshold σ ($<< 50 \times 50$). An example of preprocessing and segmentation Chinese-English sample is shown in Figure 4.2.

4.2.2/ FEATURE LEARNING

Feature extraction in writer identification is considered as an important milestone that allows characterizing the repeated patterns of the writing style. To extract such relevant features, a well-designed and effective feature extraction method is required. If these features are poorly extracted from the writing, the system will not perform correct writer identification even with a good classifier. At this stage, there is a need to extract explicit writing style features and find descriptive details that reflect individual writing habits. This

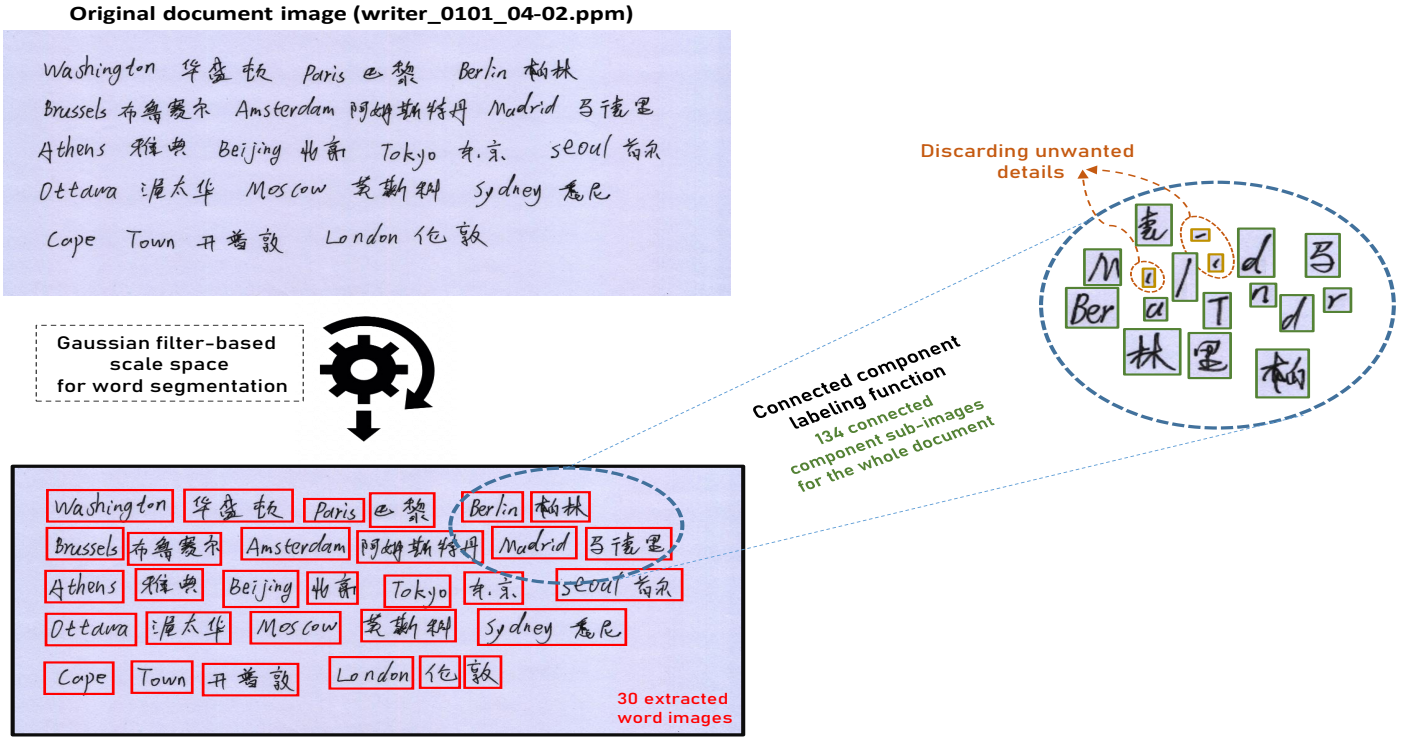


Figure 4.2: Pre-processing of English-Chinese handwritten document from CERUG-MIXED (He and Schomaker (2017b)) database.

makes the task more difficult for SOTA feature extraction methods, which typically deal with texture and local writing structure. In this step, deep convolutional neural networks (CNNs) are used as a powerful approach to handle the writing variability for robust and reliable feature extraction. Thanks to a suitable deep modeling and learning strategy, the writing style can be automatically learned to obtain the desired and more representative features for writer identification.

We propose an effective deep learning-based approach called $WriterINet$, which consists of ResNet-50 (He et al. (2016a)) and DenseNet-201 (Huang et al. (2017)) CNN models combined for feature extraction. These two networks are used simultaneously over two paths to enhance the identification performance. As shown in Figure 4.1, $WriterINet$ segments the input writing sample into word images and connected component sub-images. To enrich the learning process, the word images are the input of DenseNet-201, and the connected component sub-images are the input of ResNet-50. First, the DenseNet-201 and ResNet-50 models are pre-trained using word images and connected component sub-images from the IAM database, respectively. These pre-trained models are then used as a starting point for feature learning. Next, as shown in Figure 4.3, we fine-tune the two CNN models on each tested database by transfer learning. This is performed by replacing the last layers with new layers adapted to the tested database. Since both the ResNet-50 and DenseNet-201 models require input images of size 224-by- 224-by-3, we automatically resize them before inputting them to the two networks. After fine-tuning, deep features are learned from *Net1* and *Net2* and used to train an artificial neural network for classification, as shown in Figure 4.1.

As illustrated in Figure 4.4, the *Net1* and *Net2* models construct a hierarchical representation of the input document image where higher-level features are extracted from deeper layers. Global average pooling layer activations of *Net1* and *Net2* are used to capture features of the handwritten input document. As a result, the learned features are pooled over all spatial locations, yielding 2048 and 1920 bins in the feature dimension for each connected component sub-image by *Net1* and for each word image through *Net2*, respectively. In other words, Given a handwritten input document, each extracted word image is characterized by its *Net2* learned feature (dimension of 1920), and each extracted connected component sub-image is characterized by its *Net1* learned feature (dimension of 2048). *Net1* and *Net2* deep features of the same input document are arranged separately in matrix form. The zero-padding method is applied to the features learned from *Net2* to obtain the same feature matrix size as that derived from *Net1*. In this way, we can easily concatenate the two feature matrices of *Net1* and *Net2* vertically to form a holistic feature representation of the input writing document.

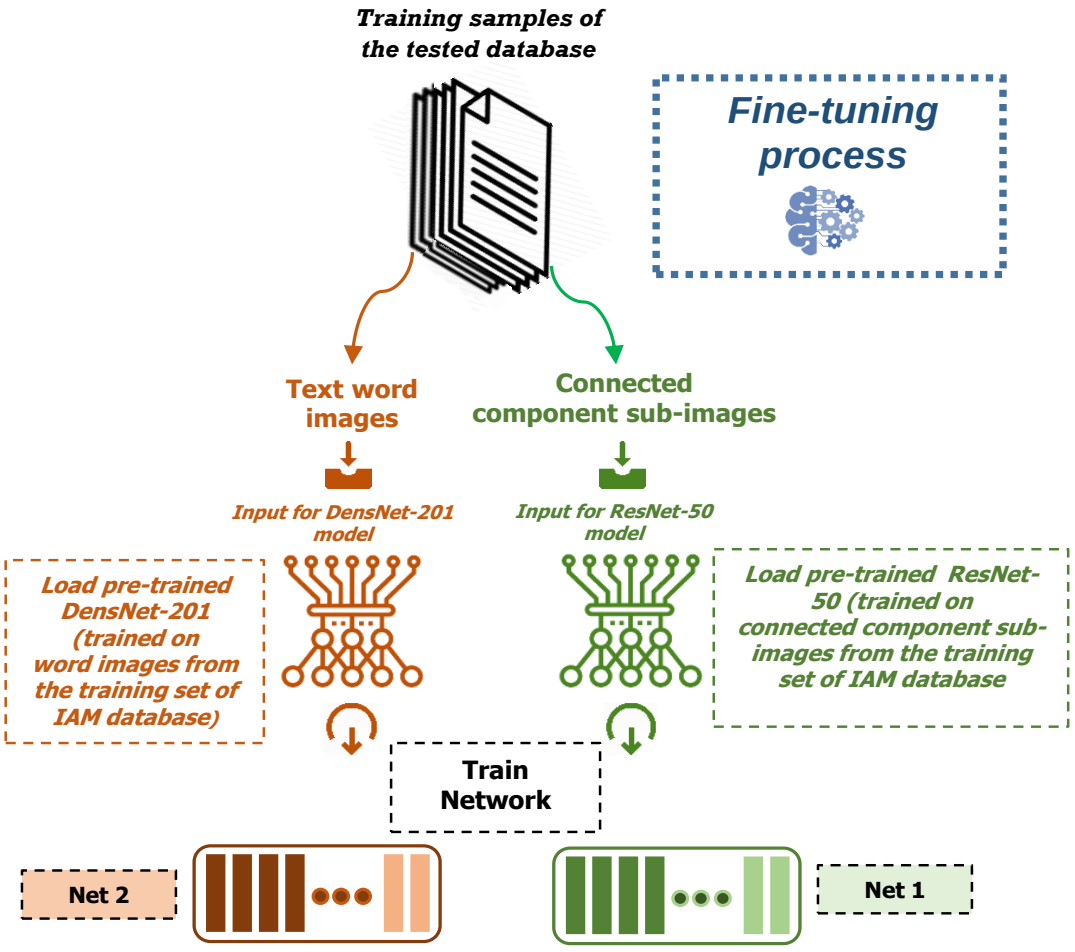


Figure 4.3: Overview of the fine-tuning process.

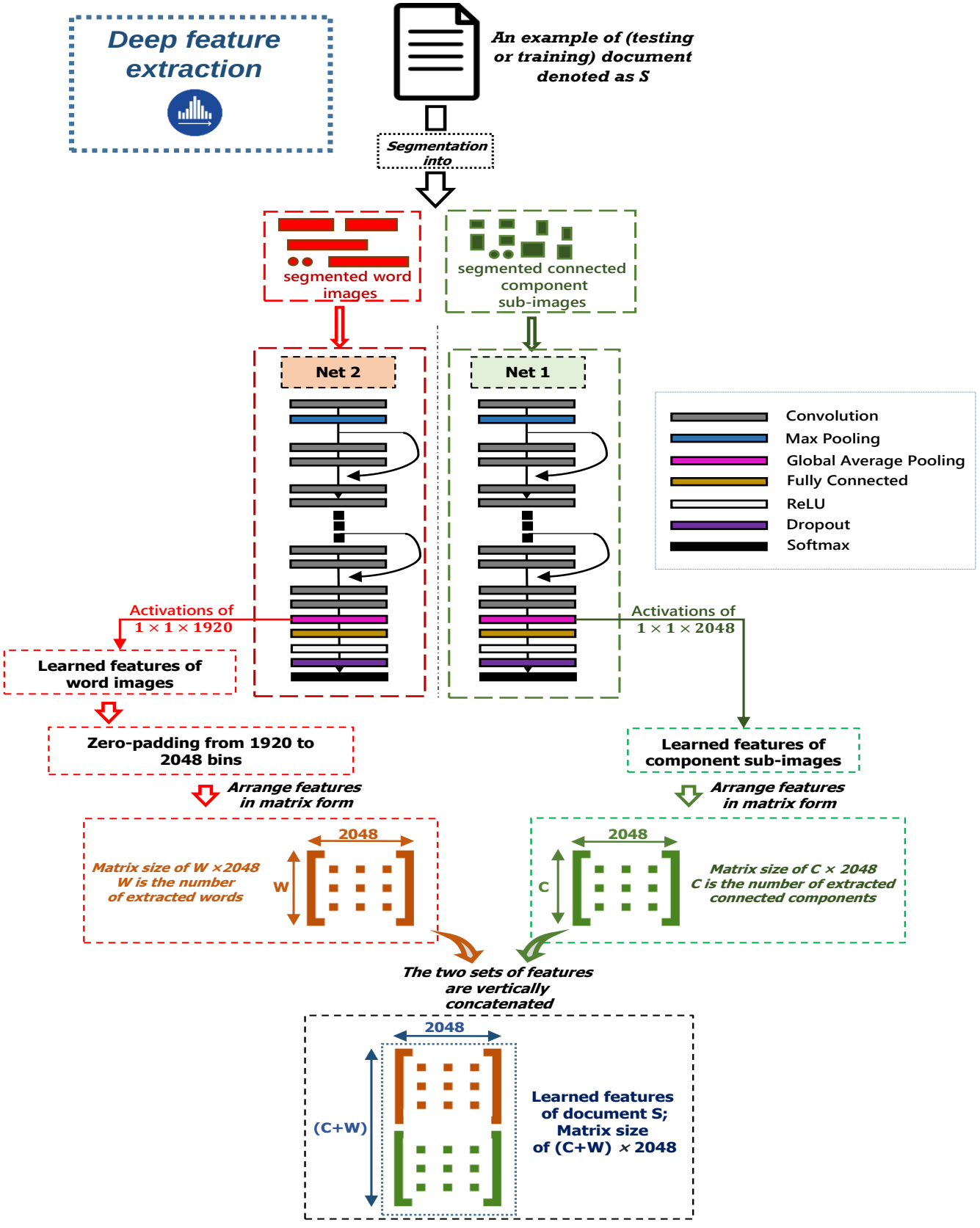


Figure 4.4: Pipeline of deep feature extraction.

4.2.3/ WRITER IDENTIFICATION PROCESS

At this level of our framework, we dispose of learned features that characterize all extracted connected components and words from each writing sample. Note that the number of extracted component sub-images and word images corresponds to the number of feature vectors representing its relevant document sample. To perform the identification (classification) process, a simple end-to-end artificial neural network (ANN) is trained on the learned features, as shown in Figure 4.5. The ANN architecture consists of a 1D input, two fully connected layers, a softmax layer, and a classification layer. Given a unseen writing sample, the testing process is performed as follows (see Figure 4.5): (i) according to the methodology in Section 4.2.2, we compute the learned features (represented by the feature matrix) corresponding to the segmented connected component sub-images and the word images; (ii) the ith feature vector, i.e., ith row of the feature matrix, is fed into the trained ANN classifier to compute its similarity score vector Si, where $i = 1, ..., F$, F ($C + W$, the number of rows of the feature matrix, as shown in Figure 4.4) denotes the number of segmented images (words and components). Si contains similarity scores related to the classes (writers), and its dimension size is the number of writers (classes); (iii) computing the final score of $j - th$ class by accumulating its scores obtained for all F feature vectors: $SCj = \frac{1}{F}\sum_{i=1}^{F} Sij$, $j = 1, ..., N$, N is the number of classes (writers); (iv) return the writer ($j - th$ class) with the highest similarity score SCj^*. Note that the similarity score vector returned by the trained classifier ANN for each feature vector (row) of the feature matrix is considered as a probability distribution over all writers. Therefore, the similarity score vectors of all observations (segmented images, i.e., words and connected components represented by the rows of the feature matrix) are averaged to produce the final prediction of the input writing sample. The testing pipeline is shown in Figure 4.5.

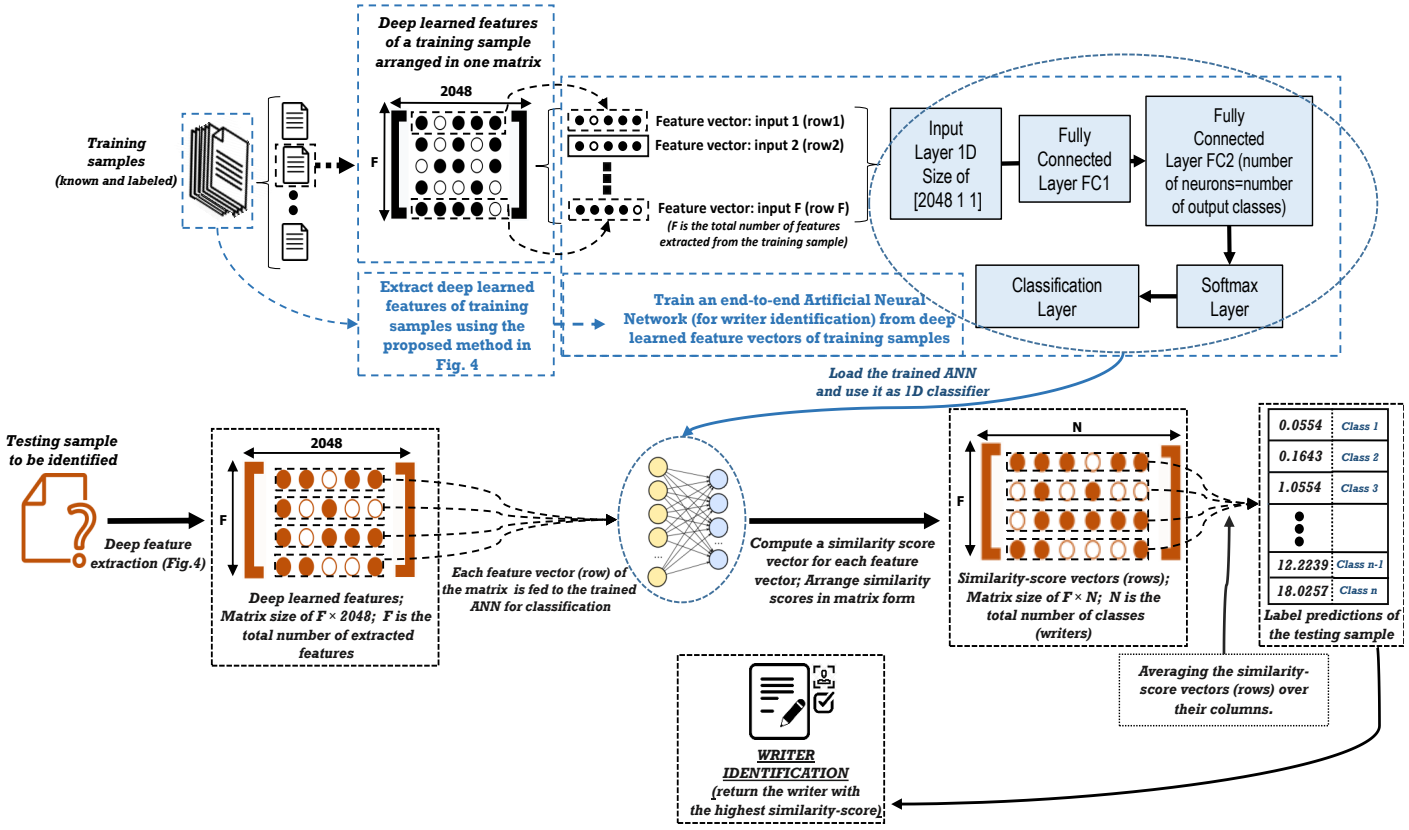


Figure 4.5: Classification process.

4.3/ AN EFFECTIVE DEEPWINET CNN MODEL FOR WRITER IDENTIFICATION

The proposed system consists of a three-stage writer identification model: (i) image preprocessing and segmentation to extract all connected components contained in each writing page, (ii) an effective end-to-end CNN model: $DeepWINet$ to characterize and classify each single connected component sub-image, and (iii) a classification process that includes the two following scenarios to identify unseen documents: (1) the $DeepWINet$ model is used as feature extraction, and the deep features (test and training features) are passed to a nearest-neighbor classifier for writer identification; (2) $DeepWINet$ is evaluated as an end-to-end CNN network to compute predicted scores. Then, we perform a score-averaging component-decision combiner to identify unseen documents. Figure 4.6 illustrates the concept of the proposed system. Each step is described in detail below.

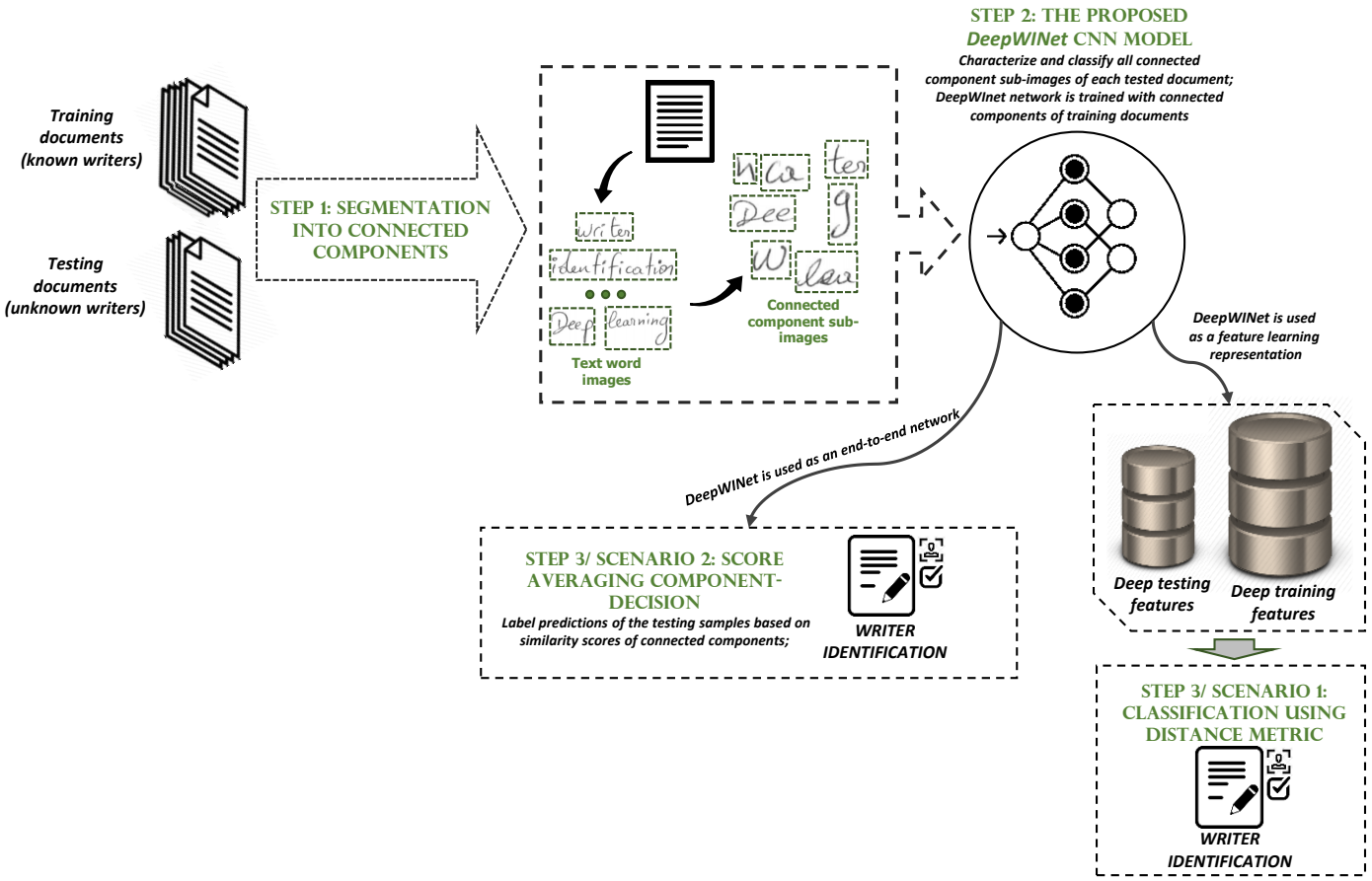


Figure 4.6: Architecture of the proposed system.

4.3.1/ COMPONENT SEGMENTATION

Each handwritten document is segmented into words and then from word images into connected component sub-images. In this step, we need to separate each word from the others. Therefore, we use blob analysis and Gaussian filters as reported in (Manmatha and Srimal (1999)) to perform scale-space coding for word segmentation. This technique gives better results which is conceptually simple [2] yet fast in processing time. The segmented words are then fed to a labeling approach to extract different regions of interest called connected components, which are the main input of our $DeepWINet$ CNN network. This labeling algorithm detects all connected neighbor pixels that form a connected writing trace, and then assigns a bounding box to each of them. Each connected component sub-image is labeled to specify its respective writer (class). Non-significant components such as diacritics and accidental writing traces are discarded using a size-based threshold with a small threshold of $\sigma\ (50{\times}50)$. An example of component segmentation is shown

[2] code: https://github.com/githubharald/WordSegmentation

in Figure 4.7.

4.3.2/ DEEPWINET CNN NETWORK

At this level, the objective is to find and capture descriptive details contained in the segmented components that reflect the writing habits of each writer. For this, we take advantage of the power of deep convolutional neural networks (CNNs) to characterize individual writing style information. With proper network modeling, handwriting can be deeply learned for writer identification. As mentioned before, the previously obtained connected components will be the main input of our CNN model. We propose an effective and reliable deep CNN architecture called *DeepWINet*. As shown in Figure 4.8, our *DeepWINet* model, inspired by VGG networks (Simonyan and Zisserman (2014)), is mainly designed based on the following two layout settings: (i) *DeepWINet* layers have the same number of filters when the feature map size remains unchanged. (ii) across the *DeepWINet* pipeline, when the feature map size is changed, we double the number of filters to balance the time complexity per layer. CNN down-sampling is performed using max-pooling layers with a kernel size of 3×3 and a stride step of 2.

This directly divides the input feature map into rectangular pooling regions and computes

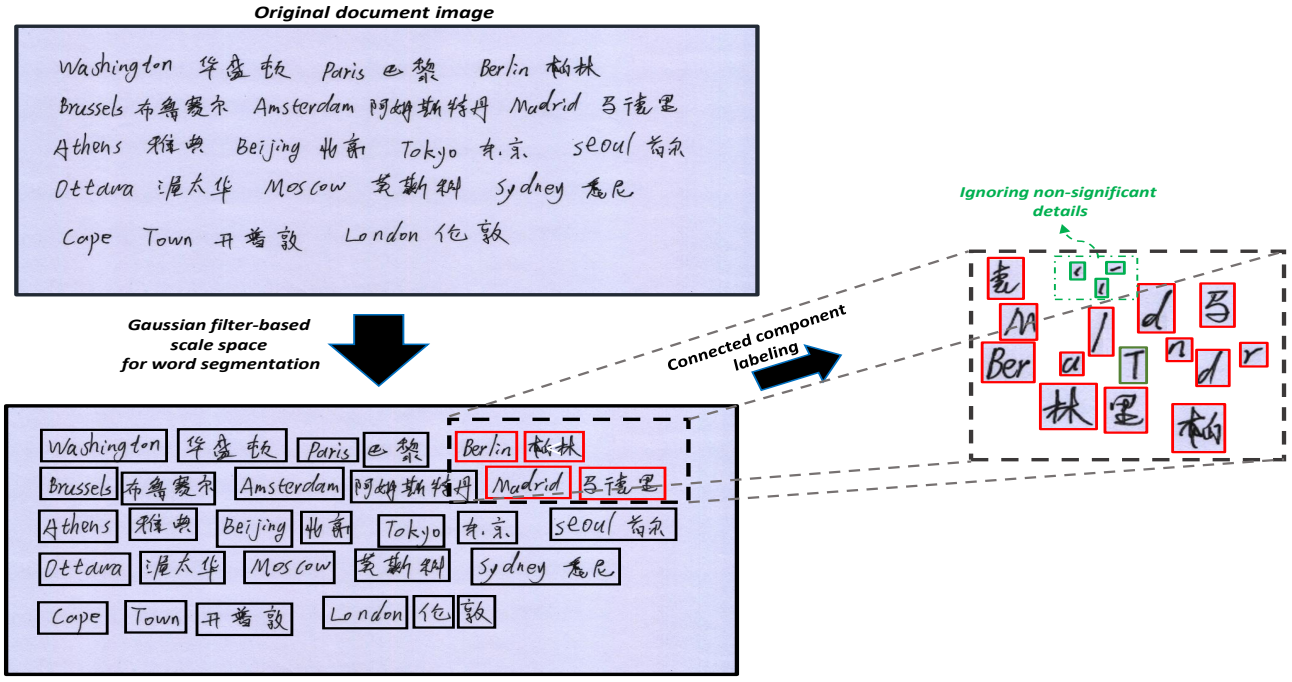


Figure 4.7: Component segmentation of English-Chinese document taken from CERUG-MIXED (He and Schomaker (2017b)) database.

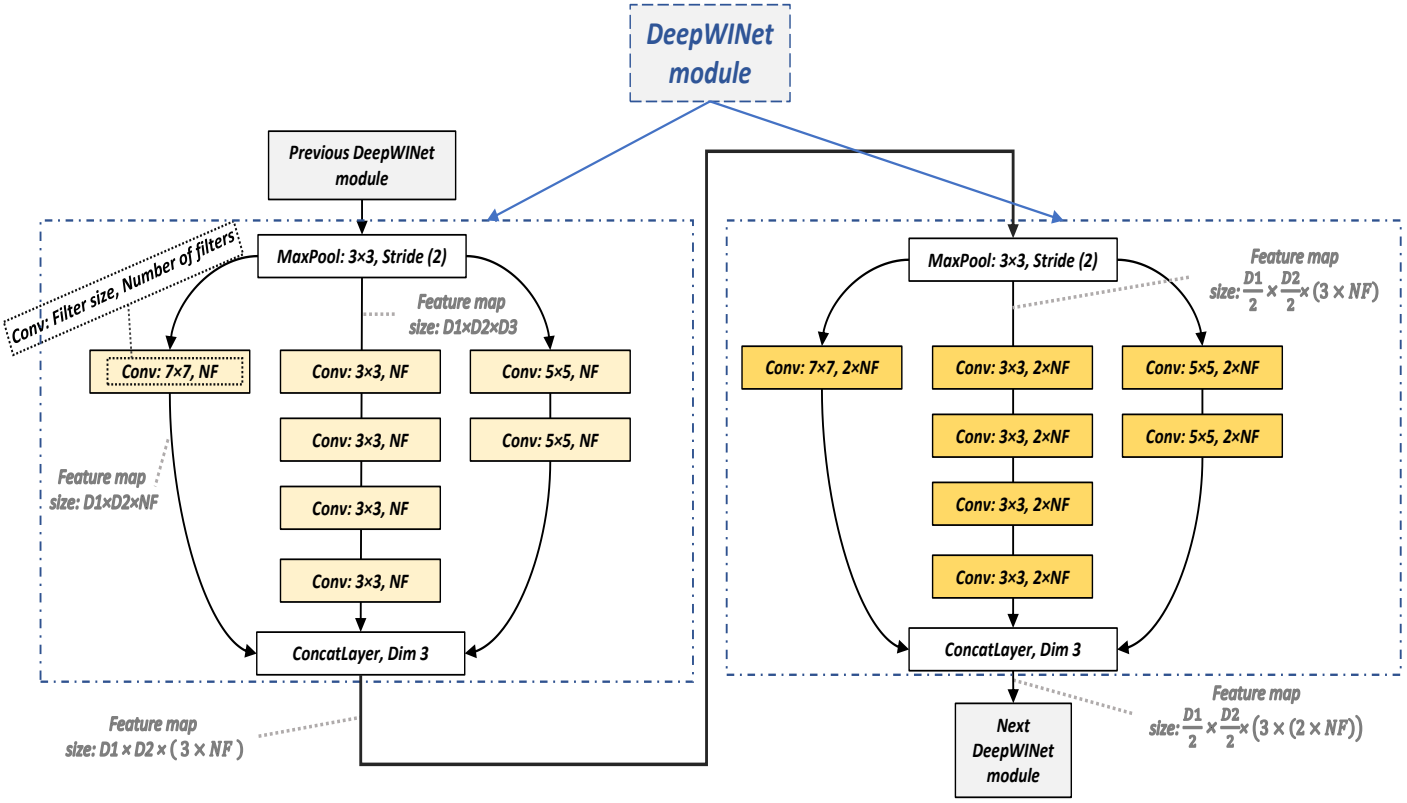


Figure 4.8: Structure and design of the proposed $DeepWINet$ module.

the maximum of each region. Batch normalization followed by layers of rectified linear units are applied directly after each convolution layer. These layers are run over a mini-batch to speed up the training of $DeepWINet$.

As shown in Figure 4.8, $DeepWINet$ branches into three paths for each feature map size: (1) The middle CNN baseline consists of four 2-D convolutional layers with filters of size 3×3. This applies sliding convolutional filters to the input map to compute the dot product of the weights and the input; (2) right CNN baseline with two 2-D convolutional layers and 5×5 in kernel size; (3) left CNN baseline has one 2-D convolutional layer with a filter size of 7×7. An increase in the filter size accompanies the reduction in the number of 2-D convolutional layers. This is related to the fact that a larger filter size is likely to capture more detail than smaller ones. Therefore, fewer CNN layers are used when filter size is important to preserve per-layer time complexity and reduce sensitivity to network initialization. Throughout the $DeepWINet$ pipeline, the outputs of the three paths are merged using the concatenation layer to form a holistic deep feature map with more discriminative details. The $DeepWINet$ pipeline ends with average-down sampling, i.e., the mean of the height and width dimensions of the last feature map is computed, using a global average-pooling layer. The last layers are fully connected (the number of neurons

is equal to the number of classes), followed by soft-max and classification layers. This calculates the cross-entropy loss for the multi-class writer identification problem and can then be used to return the scores and labels of the sub-images of the input connected components.

As shown in Figure. 4.9, we design the full and light versions of $DeepWINet$. Since the input image size in the light version of $DeepWINet$ is 80-by-80-by-3, we remove the first 2-D convolutional layer with the filter size of 7×7. This balances the feature map size and preserves more handwriting details to be processed by the next convolutional block. In total, $DeepWINet$ consists of 30-weighted layers (full version) and 24-weighted layers (light version). In terms of network complexity, our $DeepWINet$ model (light) has lower complexity compared to VGG models with 22 million parameters, which is the only 15% of the VGG-19 model (144 million parameters). The full version of $DeepWINet$ has 65 million parameters.

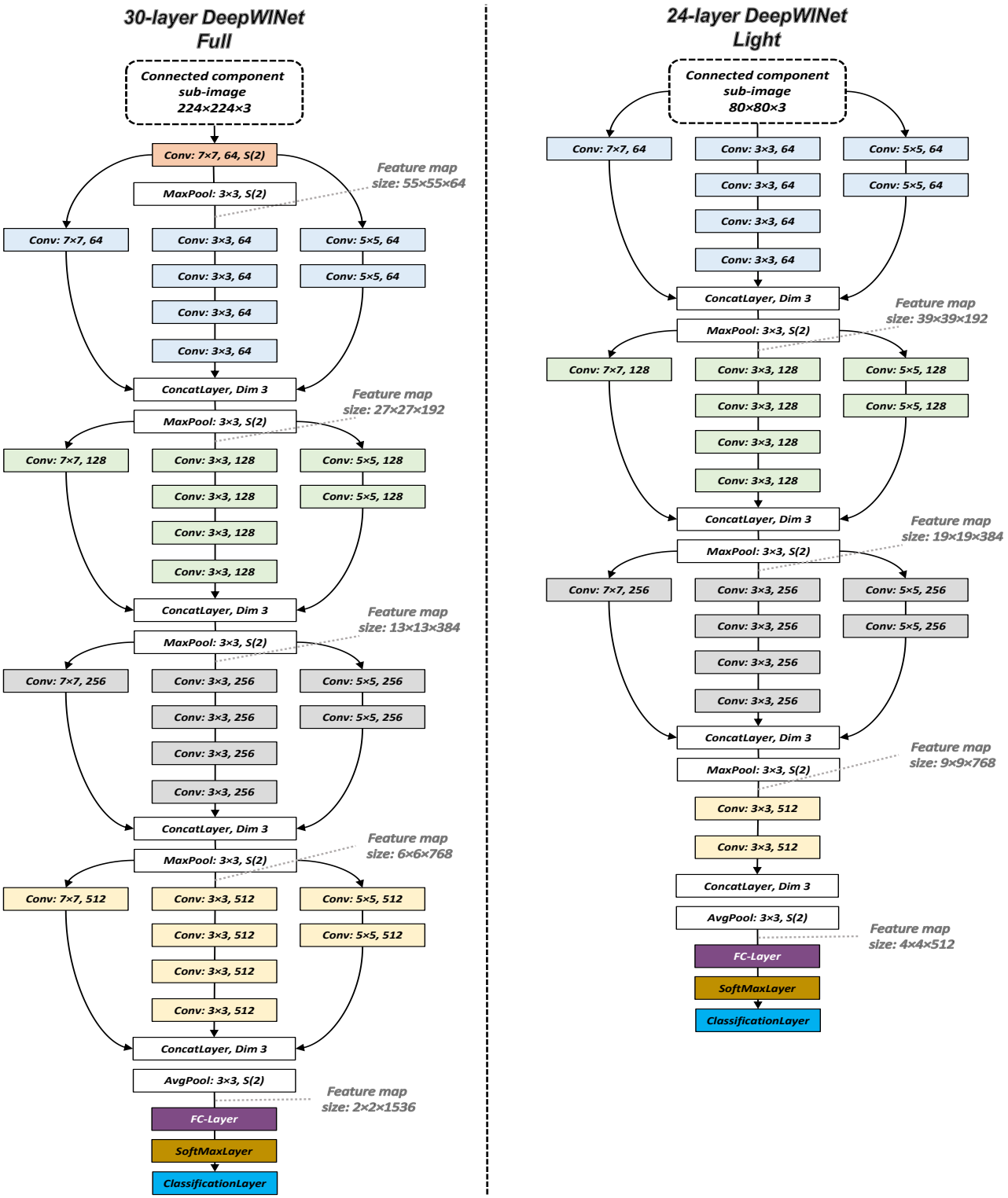


Figure 4.9: Architecture of $DeepWINet$ CNN Network with two versions (Full and Light).

4.3.3/ WRITER IDENTIFICATION

Scenario 1: Writer classification using a distance metric. For reliable feature extraction, we exploit the ability and effectiveness the $DeepWINet$ CNN model as a feature learning representation for writer identification. First, we pre-trained the $DeepWINet$ network with connected components of the training set of the IAM database (Marti and Bunke (2002)) to use the model as a starting point for feature learning. Next, $DeepWINet$ is fine-tuned through transfer learning by replacing the last three layers with new layers adapted to the tested dataset. $DeepWINet$ forms a hierarchical CNN representation where higher-level features are captured from deeper layers. Therefore, the global average pooling layer is used to extract feature activations related to each sub-image of the input connected components. This procedure performs feature pooling over all spatial locations, resulting in a feature dimension of 6144 (full $DeepWINet$) and 8192 (light $DeepWINet$). For each handwriting document, the number of learned feature vectors is equal to the number of extracted sub-images. The learned features are then fed to the nearest neighbor classifier to perform the writer identification. For this, we experimentally tested several distance metrics to explore the system performance. These include Correlation, Hamming, Chi-Square, Bhattacharyya, Cosine, and Euclidean distance metrics. The highest performance is obtained using the Chi-Square metric, which is considered as efficient distance in pattern recognition problems. The identification process used to classify a query document Q is summarized as follows: (1) Chi-Square distances between each connected component CP_Q^j in the query sample Q and all components $CP_R^{k\ (k=1,..,N_R)}$ in the training sample R are calculated. $N_R = card(R)$ is the number of connected component sub-images in the training document R; (2) the training component reporting the smallest Chi-Square distance is considered to be the one that matches the test component CP_Q^j in the query document Q; and (3) we compute the final dissimilarity metric $DISM(Q,R)$ between the query document Q (unseen to be identified) and the training document (known) R:

$$DISM(Q,R) = \frac{1}{N_Q}\sum_{j=1}^{N_Q} min\{dist\left(CP_Q^j, CP_R^1\right), dist\left(CP_Q^j, CP_R^2\right), \ldots, dist\left(CP_Q^j, CP_R^{N_R}\right)\} \tag{4.1}$$

where $N_Q = card(Q)$ is the number of connected component sub-images cropped from test document Q, and $dist\left(CP_Q^j, CP_R^k\right)$ is the Chi-Square distance between component number j (i.e., CP_Q^j) in the query document Q and the component number k (i.e., CP_R^k) in the training document R. Chi-Square distance $dist(\cdot,\cdot)$ is defined as follows:

$$dist\left(CP_Q^j, CP_R^k\right) = \sum_{t=1}^{Dims} \frac{(V^{CP_Q^j}(t) - V^{CP_R^k}(t))^2}{V^{CP_Q^j}(t) + V^{CP_R^k}(t)} \tag{4.2}$$

where $Dims$ is the feature vector dimensionality, $V^{CP_Q^j}$ is the extracted deep feature vector of the connected component CP_Q^j, and $V^{CP_R^k}$ is the deep feature vector of the connected component CP_R^k. Next, we arrange in a hit-list all dissimilarity measures between the query document Q and all the training ones $R_n, n = 1, ..., \eta$. η is the total number of training documents. The authenticity of the query document Q is then recognized as the writer of the training document that records the minimum dissimilarity:

$$\begin{aligned} WriterID(Q) = & argmin\{DISM(Q, R_1), \ldots, \\ & , \ldots, DISM\left(Q, R_\eta\right)\} \end{aligned} \tag{4.3}$$

Scenario 2: scores averaging using the proposed $DeepWINet$ ***(deployed as an end-to-end CNN network)***. In this scenario, the writer identification is performed based on prediction scores of the tested connected components, obtained from $DeepWINet$ CNN model (cf. Figure 4.6). Given a query document, the classification procedure is as follows: (1) the computation of the predicted scores corresponding to the input connected components using $DeepWINet$. In doing so, we inputted the ith connected component into the trained $DeepWINet$ to return its similarity score vector SC_i, where $i = 1, ..., N$, N denotes the number of segmented components. The prediction scores in SC_i, i.e., the probability distribution over all classes, indicate the similarity to the writers (classes). The size of SC_i is equal to the number of classes; (2) the final score of the jth writer (for each unseen document) is calculated by averaging its scores obtained for all N input connected components: $W_{sc}^j = \frac{1}{N} \sum_{i=1}^{N} SC_{ij}$, $j = 1, ..., W$, W is the number of writers (classes); (3) as the final prediction decision, the writer of the query document is the jth class with the highest similarity score W_{sc}^{j*}. The score averaging-connected component process is shown in Figure 4.10.

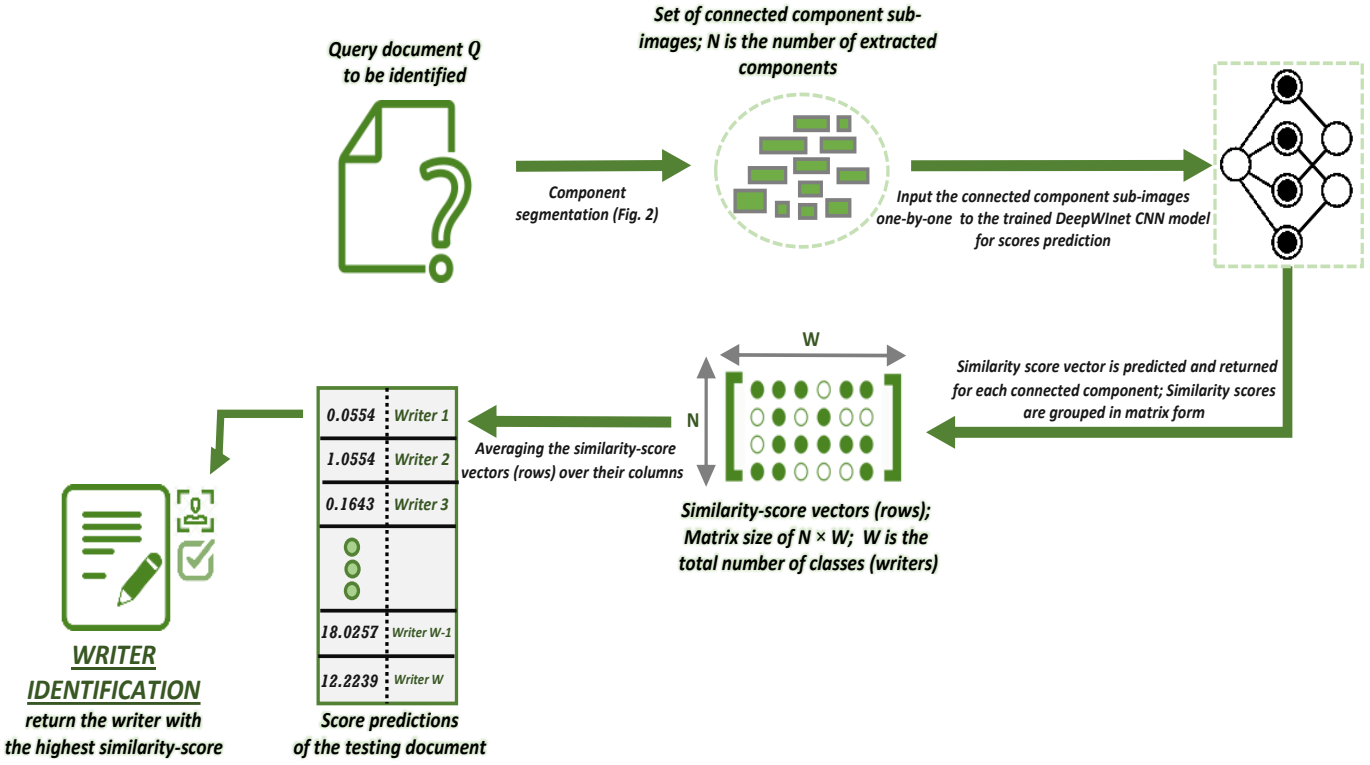


Figure 4.10: Writer identification using scores averaging.

4.4/ Image Retrieval for Historical Handwritten Fragments

4.4.1/ ICFHR2020 competition

Analyzing historical documents is a difficult task that is usually accomplished by trained humanists. However, there are still challenges in identifying historical writers when documents have particularly complex handwriting styles. The ICFHR2020 competition (Seuret et al. (2020)) investigates the performance of large-scale retrieval of historical document fragments based on document identification, i.e., identifying both the page and writer IDs. To simulate fragments, Seuret et al. (2020) extracted random snippets of text from historical document images called historical fragments. The authors used a semi-automatic technique to generate a large database consisting of more than 120 000 historical fragments written by 9800 writers from about 20 000 document images.

These fragments with random shapes and non-straight edges are generated using a free,

open-source fragmentation technique[3] developed by Seuret et al. (2020) based on the use of the diamond-square algorithm originally proposed in (Fournier et al. (1982)) and mainly used to create height maps in video games. In this way, they generated two types of fragments: (i) fragments with holes and completely random shapes, with no constraint other than being in one piece. (ii) fragments generated by cutting the historical documents using horizontal and vertical non-linear polygonal chains, resulting in a more rectangular shape, with the constraint that the chains must always move forward along an axis. Figure 4.11 shows some examples of historical fragments used in the competition. The database is publicly available in (Seuret et al. (2020)).

The historical documents are collected by many institutions and made available as manuscripts, letters, and charters. Some historical documents are collected from the publicly available Historical-IR19 test set in (Christlein et al. (2019)). Some other historical samples include a corpus of books written in the European Middle Ages, mainly from the 9th to 15th centuries CE. The letters were provided by the University Library Basel [4] where the IDs of the writers are given in a metadata.

The two main tasks of the ICFHR2020 contest are: (1) finding all similar fragment patches belonging to the same writer ID based on writing style, and (2) retrieving all fragment images segmented from the same page ID. As for the compilation of the ICFHR2020 competition database, Seuret et al. (2020) provided 101 706 fragments as the training set collected from 8717 writers from 17 222 historical documents. The test set consists of 20 019 fragments from 1152 writers generated from 2732 historical documents. Note that the IDs of the writers in the training set are different from those in the test set, and there is no overlap between the two sets. Seuret et al. (2020) found that the number of fragments provided is sufficient to train and test deep neural networks for the competition tasks.

[3] https://github.com/seuretm/diamond-square-fragmentation
[4] https://www.unibas.ch/de

Figure 4.11: Examples of generated historical document fragments. The two examples on the left have rectangular shapes. The two examples on the right have completely random shapes.

4.4.2/ PARTICIPANTS OF THE ICFHR2020 COMPETITION

Five methods, including our proposed approach (with two variants), from different universities, are evaluated in the ICFHR2020 competition. Since the test set includes 20019 historical fragments, participants are asked to submit a CSV file containing 20019×20019 distance matrix. The leave-one-image-out-cross-validation rule is used as the evaluation strategy for participants' results. In other words, each of the 20019 fragment images in the test set is used as a query sample, while the other samples are ranked in a hit list according to their distance similarities (the smaller, the more similar). The metrics are then averaged over all the queries. The error metrics and experimental results are discussed in detail in Chapter 5 (cf. Section 5.6.1). The approaches presented are briefly explained in the following.

4.4.2.1/ CONTEST BASELINE

The competition baseline is the system presented in (Nicolaou et al. (2015)). They used an image descriptor called Sparse Radial Sampling Local Binary Patterns (SRS-LBP), a variant of Local Binary Patterns (LBP) originally proposed for document image analysis. The SRS-LBP operator transforms each image pixel into multiple SRS-LBP codes representing the relationship between the central pixel and its surrounding pixels. This encodes the input image as multiple 8-bit images, with a histogram of SRS-LBP codes computed for each radius (for radii 1 to 12). A global SRS-LBP pooling is performed to form a normalized histogram with 256 bins for each radius, and then all histograms are concatenated into a 3072-dimensional block-normalized descriptor. Using the Principal Component Analysis (PCA) projection, the 3072-dimensional vector is mapped to the first 200 principal components. The Hellinger kernel is applied to the 200-projected vector (representing each historical fragment image), followed by l^2 normalization. Manhattan similarity was used as a metric to calculate the distances.

4.4.2.2/ UNIVERSITY OF GRONINGEN

Sheng He and Lambert Schomaker from the University of Groningen used a CNN network called FragNet, which was previously proposed for writer identification in (He and Schomaker (2020)). The FragNet architecture is structured into two streams: a feature pyramid stream used to extract feature maps and a CNN fragment stream (fragments as input) trained to identify the writer based on the predicted scores of fragment images and feature maps (computed from the feature pyramid). Note that only the cross-entropy loss is used for the FragNet model instead of the triplets loss. Horizontal/vertical projections remove some black edges created during historical fragment generation. Since the CNN

FragNet model requires input images of size 64×128, each fragment is partitioned into non-overlapping blocks of dimension 64×128. For each historical fragment, the FragNet features of all extracted blocks are averaged into a 512-dimensional feature vector. The Euclidean metric is used as the distance measure for the competition tasks.

4.4.2.3/ UNIVERSITY OF BOURGOGNE FRANCHE-COMTE

Michel Chammas, Abdallah Makhoul, and Jacques Demerjian from the Femto- ST Research Institute at the University of Bourgogne Franche-Comte have proposed a CNN-based approach. The overall system is based on the approach presented in (Christlein et al. (2017b)) with some improvements. First, scale-invariant keypoint descriptors (SIFT) are computed from the fragment images. Principal component analysis (PCA) is used to reduce the dimension of SIFT descriptors from 128 to 32 and then clustered using the K-means algorithm originally proposed in (MacQueen et al. (1967)). A deep residual network (ResNet20) is trained with the cluster IDs (targets) using patches extracted from each SIFT location (from the same SIFT keypoints). The deep features are extracted from the penultimate CNN activation layer, encoded, and normalized to produce a global feature vector. The authors used the multi-VLAD approach (Vector of Locally Aggregated Descriptors) to aggregate all features into a holistic global descriptor. An incremental PCA algorithm (Goel (2019)) with whitening was used to reduce the dimension of the global descriptor. The distance matrix is calculated using the Cosine similarity.

4.4.2.4/ UNIVERSITY OF TEBESSA

Abdeljalil Gattal and Chawki Djeddi from the University of Tebessa used the oriented Basic Image Features (oBIFs) column histograms as in their earlier work in (Gattal et al. (2016)) and (Abdeljalil et al. (2018)). Two different configurations of oBIFs, i.e. σ= 2, 4 and σ= 1, 8, are extracted directly from the historical fragments and concatenated into a feature descriptor. The oBIF parameter ϵ is set to 0.01, and the Correlation distance is calculated to obtain the results as similarity measures.

4.4.2.5/ PROPOSED DEEP LEARNING-BASED APPROACH

For robust and reliable feature extraction, we use deep convolutional neural networks (CNNs) to characterize the writing style of historical handwritten fragments. In our method, the CNN activations of the ResNet-50 model (He et al. (2016a)) are used as deep features. In particular, two ResNet-50 models are trained and fine-tuned with different input data (fragment images of the ICFHR2020 training set). As shown in Figure 4.12, each fragment image is fed into two CNN path-ways: *Net1* uses the entire fragment

image, while for *Net2* the input image is split into four sub-blocks, i.e., a vertical split and a horizontal split. Therefore, *Net2* uses a total of about 400 000 training samples.

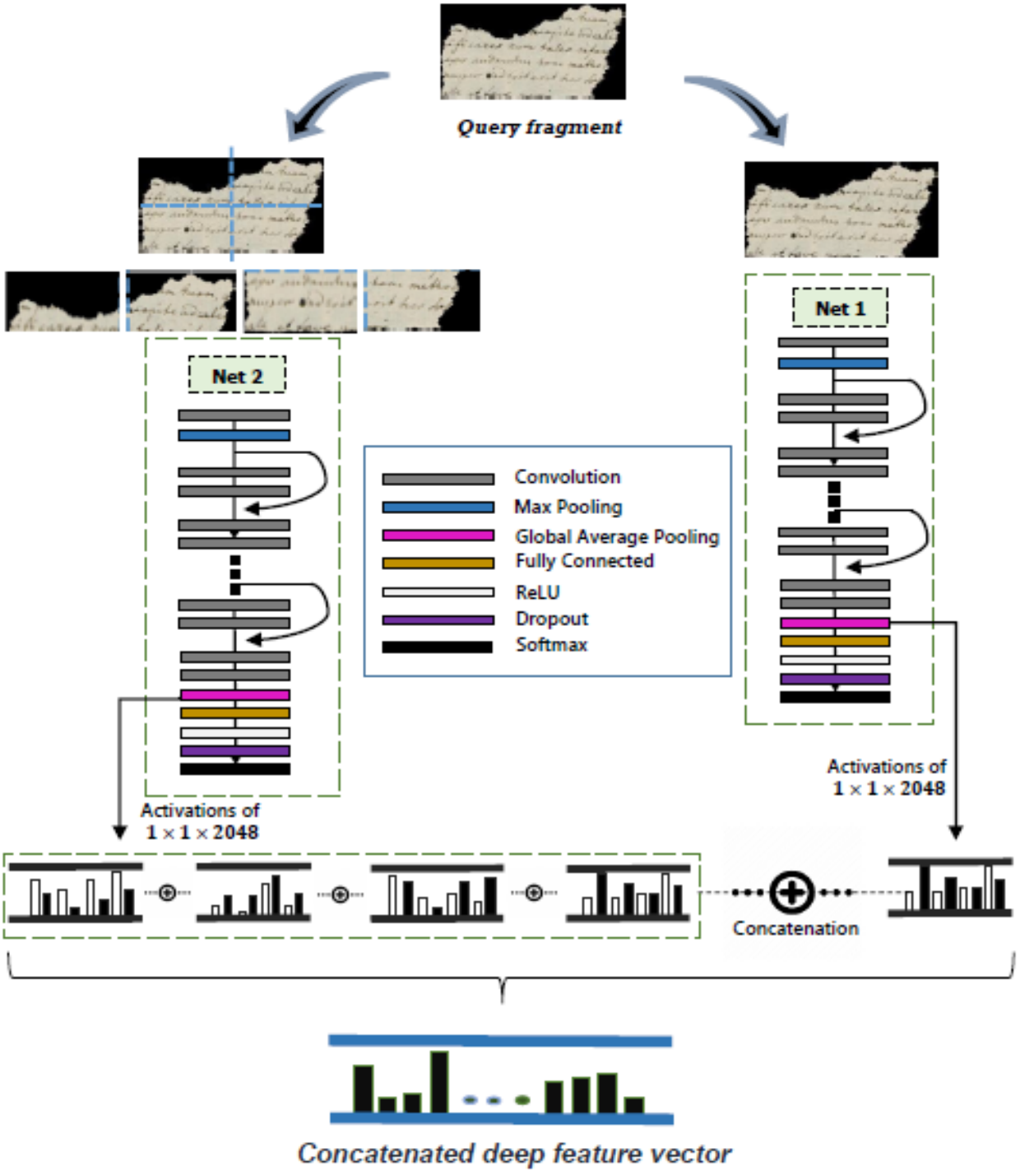


Figure 4.12: Overview of the proposed approach.

For training, the Adam optimizer proposed in (Kingma and Ba (2014)) is used with a learning rate of 0.0001, 10 epochs, and a mini-batch size of 40. In the CNN architecture, higher-level features are captured by deeper layers. The average pooling layer is activated to extract learned deep features of the test fragment images. More specifically, the activations of the average pooling layer, which consists of a 2048-deep feature vector of each network (*Net1* and *Net2*), are concatenated, resulting in a 4096-global feature representation for each fragment image. Finally, the distance computation is performed using the χ^2 distance metric. Note that we experimentally evaluated several distance metrics, including correlation, Hamming, Manhattan, Chi-Square, Bhattacharyya, cosine, and Euclidean. The correct performance (on the training set) is obtained with the Chi-Square metric, which is considered as an efficient distance in pattern recognition problems.

We have submitted two variants of our approach. The first model ($TwoPath_{writer}$) is trained with writer labels, and the second one ($TwoPath_{page}$) is trained with page labels. Both models achieved excellent results, and our $TwoPath_{writer}$ won the first place in Task 2 (Image Retrieval) of the ICFHR2020 competition. The experimental results and discussions are reported in Chapter 5 (Section 5.6.2).

4.5/ CONCLUSION

To further improve the task of offline text-independent writer identification, we exploited the effectiveness of Convolutional Neural Network as a powerful deep learning tool to characterize writing variability. In this chapter, two different CNN-based approaches, namely $WriterINet$ and $DeepWINet$ are presented. The former approach segments handwritten documents into word and connected component images, from which deep features are computed. To obtain a discriminative feature representation, we proposed a multi-path deep network consisting of two CNN streams trained and fine-tuned with different input data. A conceptually simple and effective artificial neural network (1D- ANN) was designed to classify CNN features for writer identification in the classification phase. The second proposed approach involves a reliable and efficient deep CNN architecture, called $DeepWINet$, which has 30-weighted layers for the full version and 24-weighted layers for the light version. Compared to the VGG network structure (144 million parameters), our $DeepWINet$ (light version) has lower network complexity with only 22 million parameters.

The $DeepWINet$ model was used as a first scenario to compute CNN activation features from the connected components of the writing. Then, the features are fed into a nearest-neighbor classifier using Chi-square similarity as a distance metric to perform writer identification. The second scenario aims to train $DeepWINet$ as an end-to-end CNN network, where the predicted scores are averaged using a new and efficient strategy, the score-

averaging component-decision combiner. The next chapter is devoted to validating the effectiveness and superiority of our methods by comparing their performance with the literature.

This chapter also presented our proposal for retrieving historical documents based on writer identification. In this context, a competition was announced at the ICFHR2020 conference to award the best performing learning system in image retrieval for historical handwritten fragments. The goal is to find all similar fragment images belonging to the same writer ID (Task 1) and find all fragment images segmented from the same page ID (Task 2). We contributed with an effective approach based on two CNN networks sharing the same feature learning process, each trained with different fragment patches. The Ch-square distance metric was used to compute the similarities between fragments. The overall approach achieved higher retrieval scores and won first place in Task 2 (Page Retrieval) and second place in Task 1 (Writer Retrieval). These results are discussed in the next chapter.

5

EXPERIMENTS AND DISCUSSIONS

5.1/ INTRODUCTION

In order to evaluate our proposals in Chapters 3 and 4 and rank them in the literature, this chapter is devoted to the study of experimental results obtained with our different approaches to text-independent offline writer identification. A detailed description of the benchmarks used with the standard protocol setup are provided for the evaluations. To the best of our knowledge, we were one of the first to perform extensive experiments on 10 challenging handwritten databases with different languages (English, Arabic, Dutch, Chinese, French, German, and Greek). For each proposed texture-based system (reported in Chapter 3), a series of experiments are performed to investigate and evaluate the stability of the system performance under different configurations. This includes analyzing the key parameters of each method and studying the system sensitivity as a function of the number of writers and the amount of data for each benchmark studied. For the CNN-based approaches discussed in Chapter 4, we list comprehensive evaluations to validate their effectiveness in better characterizing the writing variability. The implementation details and training options of the CNN models are also presented. In a separate section, we compare and discuss our achieved performance (complete results of our proposals) with the current state-of-the-art.

This chapter also presents the evaluation protocol and error metrics used in the ICFH2020 competition for image retrieval for historical handwritten fragments and provides the retrieval results with a performance comparison with the participants' approaches.

5.2/ BENCHMARK AND EXPERIMENTAL SETUP

Availability of datasets is one of the basic requirements for development and evaluation in any research area. For writer identification, there are various handwriting databases reported in the literature with diverse script languages. This section provides a detailed

description of well-known benchmarks used to evaluate the performance and effectiveness of our proposed approaches along with the current state- of- the- art. Extensive experiments are conducted on 10 popular publicly handwritten databases: (1) English IAM (Marti and Bunke (2002)), (2) Arabic IFN/ENIT (Pechwitz et al. (2002)), (3) English ICDAR2013 (Louloudis et al. (2013)), (4) Dutch Firemaker (Schomaker and Vuurpijl (2000)), (5) English CVL (Kleber et al. (2013)), (6) English CERUG-EN (He and Schomaker (2017b)), (7) Chinese CERUG-CN (He and Schomaker (2017b)), (8) Mixed Chinese and English CERUG-MIXED (He and Schomaker (2017b)), (9) Arabic AHTID/ MW (Mezghani et al. (2012)), and (10) hybrid language ICDAR2011 (Louloudis et al. (2011)). The description and evaluation setup for each tested database are presented in the following:

IAM. The IAM database (Marti and Bunke (2002)) is one of the most widely used English handwriting databases in the literature for both writer identification/verification and handwriting recognition. It contains 1539 forms with 13353 isolated and labeled handwritten English text lines with variable content. Each form was scanned at 300 dpi and provided as a PNG image with 256 gray levels. An example of handwritten document is given in Figure 5.1. In total, 657 writers contributed one to 59 pages of their handwriting. In the experiments, we set the following two default settings: # Document Setup: Two documents are used per writer: 1 document for training and 1 document for testing. For writers who created only one document, the page image is split into two sub-pages. # Text Line Setup: A maximum of 14 text lines are used for each writer. 60% of the data is used for training and the remaining 40% is used for testing.

IFN/ENIT. The IFN/ENIT database (Pechwitz et al. (2002)), the most popular Arabic handwriting database, was developed primarily for training and evaluation of Arabic handwriting recognition systems and has been widely used for the problem of writer identification. It contains 2200 forms scanned at a resolution of 300 dpi. The forms (cf. Figure 5.2), which are also in binary image format, include 26000 handwritten Tunisian city/village names written by 411 writers. A maximum of 50-word binary images is used per writer. 60% of the word images are selected for the training set, and 40% are used for testing.

ICDAR2013. The ICDAR2013 (Louloudis et al. (2013)) database consists of 1000 Greek and English handwritten document samples collected from 250 scribes (2 documents in English and 2 in Greek per writer). In the experiments, we use two documents for testing and the others for training. An example of English and Greek handwriting produced by the same writer is shown in Figure 5.3.

Firemaker. The Firemaker database (Schomaker and Vuurpijl (2000)) contains Dutch document samples of handwritten text of variable content, scanned at 300 dpi and grayscale, collected from 250 writers, mainly students. Each writer was asked to complete four different A4 pages of handwritten text (cf. example in Figure 5.4). On page 1,

five short paragraphs are written in normal handwriting, i.e., lowercase letters with some capital letters at the beginning of sentences and names. Page 2 contains another handwritten text of two paragraphs with only capital letters. On page 3, writers were asked to

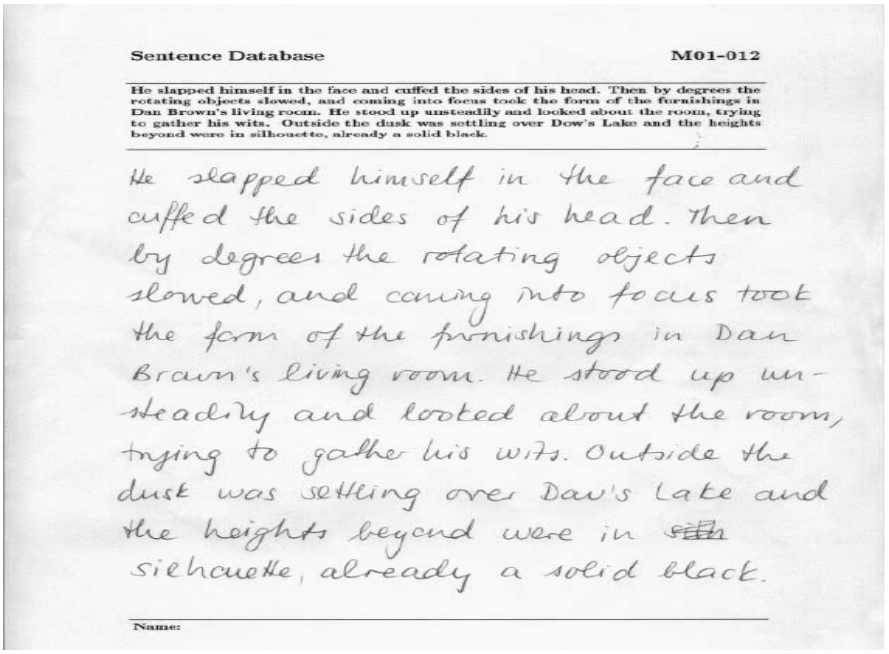


Figure 5.1: An example of a filled form taken from IAM database

Figure 5.2: Filled form and words from IFN/ENIT database

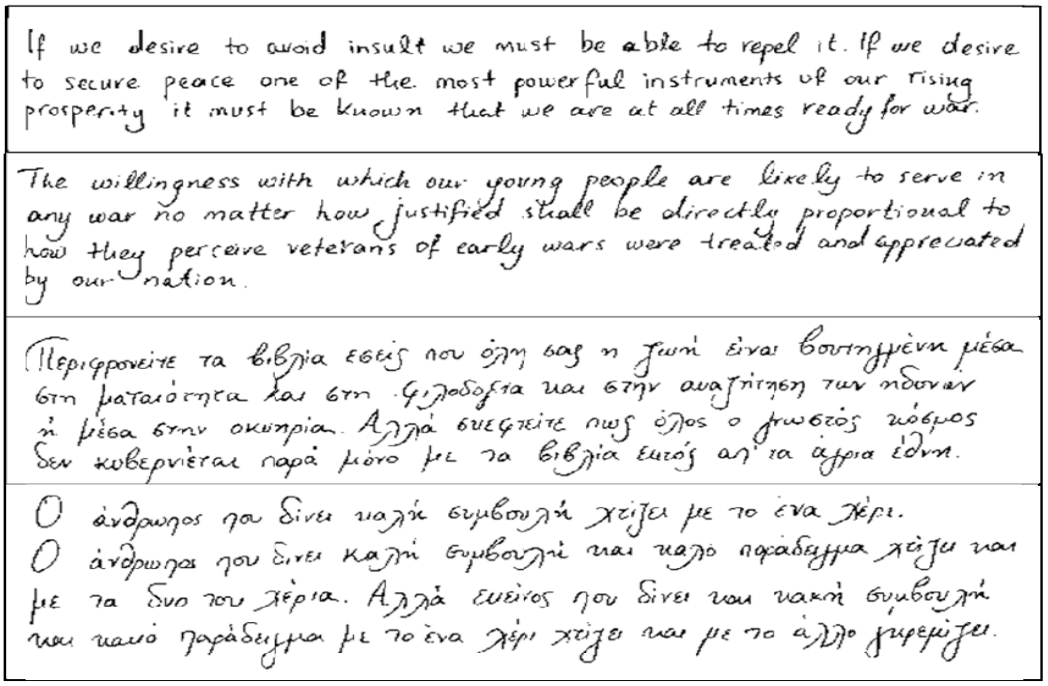


Figure 5.3: Image samples from the same writer written in English and Greek language from ICDAR2013 database

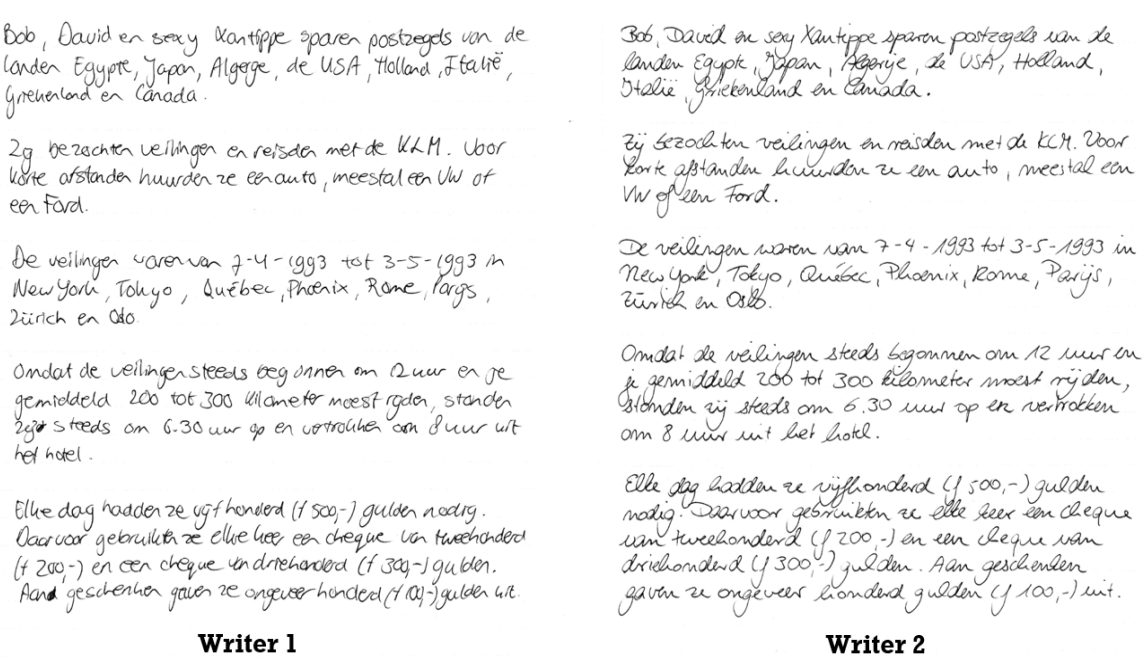


Figure 5.4: Image samples with the same content written by two different writers from Firemaker database

produce a forged text, while on page 4, they were asked to write and describe the content of a given cartoon in their own words. Only two pages are selected in the experiments

(half-half setup): Page 4 is used for testing, while Page 1 was used for training.

CVL. The CVL database (Kleber et al. (2013)) is a publicly available standard database of handwritten documents for writer identification and word spotting. It contains English and German handwritten texts written by 310 scribes, of whom 283 wrote five written documents (four in English and one in German) and 27 wrote seven documents (six in English and one in German). The scanned text images are in RGB color format with a resolution of 300 dpi. We set two standard protocols: # Setup-1: only English documents are used (one document for testing, while the last three are used for training). # Setup-2: only the first five documents are used in our experiments to ensure an evenly distributed dataset. The first two documents are used as a test set, and the other three are used as the training set.

CERUG. This dataset disposes of Chinese-English documents scanned at 300 dpi, 8 bits/pixel, grayscale, and produced by 105 writers. Each writer is asked to fill four different A4 documents: two pages in Chinese, one page in English, and the fourth page is mixed. The English page is divided into two sub-pages, each containing one paragraph. According to this arrangement, the English CERUG-EN subset is collected. Similarly, the English-Chinese pages are divided into two subsets to form the CERUG-MIXED dataset. The third subset is called CERUG-CN and contains only the Chinese pages. A half-half setting is applied to the three subsets, where one sub-page in the testing and the other for training. An example of image samples from the same writer is shown in Figure 5.5.

AHTID/MW. The Arabic Handwritten Text Images Database of Multiple Writers (AHTID/MW) (Mezghani et al. (2012)) is an open-access database for researchers working on Arabic handwritten text recognition worldwide. It involves 53 individuals from different educational levels and ages with free choice of pen. Each person was asked to write 70 lines of text, collecting a total of 3710 lines of text, where each line contains an average of 6.17 words and each word has 5.53 characters. Then, 22896 word images are collected for all writers. The handwritten words are scanned and saved in PNG format as grayscale images with a resolution of 300 dpi. The database was divided into 4 sets of word images, where 3 sets are used for training and the last set is used for testing.

ICDAR2011. The ICDAR2011 (Louloudis et al. (2011)) database contains English, French, Greek, and German handwritten texts written by 26 writers. Two full pages of handwritten text for each language, for a total of 8 pages per writer. We used a subset of the ICDAR2011 database, the so-called CICDAR2011 dataset, where only the first two lines of text are truncated from each handwritten page (cf. Figure 5.6). The first five pages are used as the training dataset, and the other three pages are used as the test dataset.

Standard protocols used in our experimental study are summarized in Table 5.1.

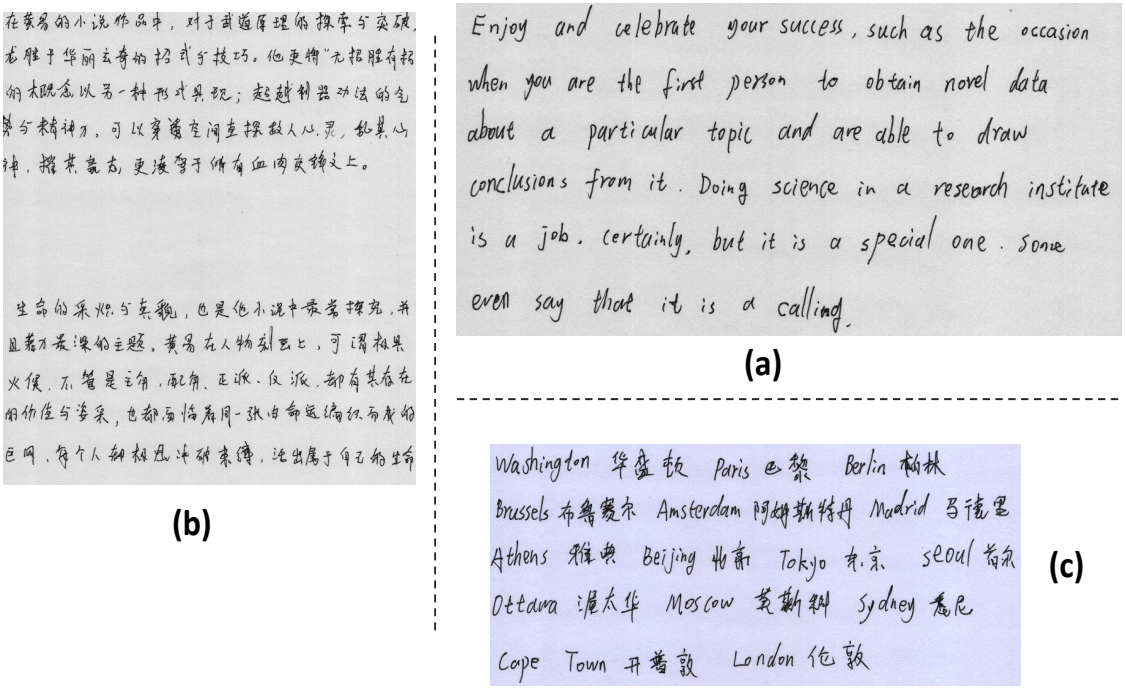


Figure 5.5: Image samples from the same writer: (a) English sample from CERUG-EN dataset; (b) Chinese sample from CERUG-CN dataset; (c) English-Chinese sample from CERUG-MIXED dataset

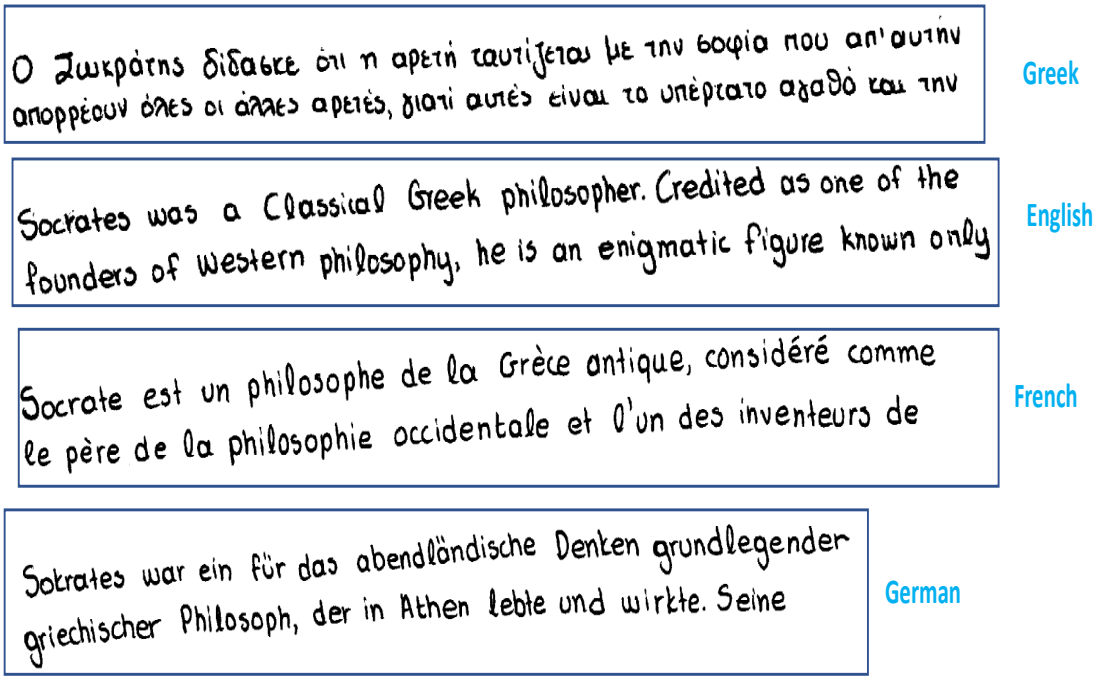


Figure 5.6: English, French, Greek, and German samples from ICDAR2011 database

Table 5.1: Experimental setup. The use of 2 or 3 documents in testing is more challenging than using 1 document.

Dataset	Script	Classes (Writers)	Year	Query sample	Total number of query documents
CERUG-CN	Chinese	105	2017 He and Schomaker (2017b)	1 document	105
CERUG-EN	English	105	2017 He and Schomaker (2017b)	1 document	105
CERUG-MIXED	Chinese/English	105	2017 He and Schomaker (2017b)	1 document	105
CVL (setup-1)	English	310	2013 Kleber et al. (2013)	1 document	310
CVL (setup-2)	English/German	310	2013 Kleber et al. (2013)	2 documents	620
IFN/ENIT	Arabic	411	2002 Pechwitz et al. (2002)	20 words at most	411
IAM (document setup)	English	657	2002 Marti and Bunke (2002)	1 document	657
IAM (text-line setup)	English	657	2002 Marti and Bunke (2002)	about 6 text lines	657
Firemaker	Dutch	250	2000 Schomaker and Vuurpijl (2000)	1 document	250
ICDAR2013	Greek/English	250	2013 Louloudis et al. (2013)	2 documents	500
ICDAR2011	Hybrid	26	2011 Louloudis et al. (2011)	3 documents	78

5.3/ EXPERIMENTAL RESULTS: TEXTURE FEATURES-BASED SYSTEMS

Our proposed texture-based approaches, explained earlier in Chapter 3, are performed in a text-independent manner, i.e., each writer has completely different handwriting samples in the training and testing sets. Moreover, each handwritten document (or set of words/lines of text) is considered as a scanned image characterized by a set of feature vectors computed from all its segmented connected components. In the proposed framework (cf. Chapter 3), the classification process is performed over the Nearest Neighbor -classifier (1-NN), which is applied to the normalized feature vectors using the Hamming distance metric. The system performance is given by % (Writer Identification Rate). For all writers, the performance score is the number of testing samples, which are correctly classified divided by the total number of test samples (unseen documents).

5.3.1/ PERFORMANCE OF THE BWLBC-BASED APPROACH

5.3.1.1/ CLASSIFICATION RESULTS

The BWLBC-based approach is evaluated on the IFN/ENIT, IAM (text-line setup), CVL (setup-1), and AHTID/MW databases. Unlike most existing works in the literature (Abdi and Khemakhem (2015); Hannad et al. (2016); Khan et al. (2017)) where the standard evaluation protocol is used, i.e., only one subdivision into training and testing sets is used, the proposed system is evaluated using 10 different split permutations randomly generated for each writer to assess its performance stability over the IFN/ENIT and IAM databases. Thus, the identification procedure is repeated 10 times, each time with a different split between training and test sets, and the average accuracy is reported as the final result. In the CVL and AHTID/MW databases, a quad validation strategy (cross setting) is specified in the setup of the CVL benchmark. This experimentation strategy thoroughly evaluates the performance of our system. It generates and tests all possible

splits since we used one sample in testing and three samples in training. The identification process is iterated four times with different split configurations. The final result is the average rate over the four sets. The performance of BW-LBC (Block Wise Local Binary Count) descriptor is compared with LBP (Local Binary Patterns), LTP (Local Ternary Patterns), and LPQ (Local Phase Quantization) texture descriptors, which are the most commonly used descriptors in writer identification (Hannad et al. (2016); Khan et al. (2016); Bertolini et al. (2013)).

Tables 5.2 and 5.3 report the average top-1 identification rates achieved across all splits tested, as well as the rates recorded across each split separately on IFN/ENIT and IAM databases (cf. Table 5.2) and CVL and AHTID/MW (cf. Table 5.3), respectively. From these tables, it is clear that the BW-LBC operator is significantly and consistently the best descriptor in terms of overall writer identification rate on all tested databases compared to the evaluated methods. Moreover, it is easy to see that the BW-LBC operator shows significant performance stability for all subdivisions (in training and test sets) across the four tested datasets, where the identification rates tend to converge to the average accuracy. As for the optimal parameters of the BW-LBC method, it is noteworthy that they are empirically determined for all tested databases through the extensive experiment reported in Section 5.3.1.2.

The window size $W \times W$ of 300×300 pixels of the connected components with the number of blocks $N \times N = 49$ enable to achieve the best top-1 average accuracies of 96.47% and 88.99% on the IFN/ENIT and IAM databases, respectively, while the top-1 average of 98.38% is recorded for the CVL database with the window size $W \times W$ of 400×400 pixels and the number of blocks $N \times N = 49$. For the AHTID/MW database, an identification rate of 99.53% is recorded as the top-1 average accuracy with the number of blocks $N \times N = 16$ and 300×300 pixels in the window size. As for the evaluated state-of-the-art descriptors, none of them performs well in the tested datasets. For example, considering the LTP descriptor, which is considered the $2nd$ best descriptor in the IFN/ENIT database,

Table 5.2: Identification rates on the IFN/ENIT (411 writers) and IAM (657 writers) databases

Descriptor	Database	Split										Average accuracy	Dimension
		Sp.1	*Sp.2*	*Sp.3*	*Sp.4*	*Sp.5*	*Sp.6*	*Sp.7*	*Sp.8*	*Sp.9*	*Sp.10*		
BW-LBC	IFN/ENIT	95.37	96.83	96.59	97.08	**97.56**	96.35	96.83	96.59	94.89	96.59	96.47%	49
	IAM	**90.11**	88.58	88.89	**90.11**	88.89	88.13	88.43	88.74	89.49	88.58	88.99%	
LPQ	IFN/ENIT	73.24	74.69	73.48	75.42	74.94	73.72	72.75	72.99	74.28	75.18	74.07%	256
	IAM	74.12	72.75	73.66	73.21	73.82	75.49	72.45	72.3	73.51	72.60	73.39%	
LBP	IFN/ENIT	69.83	70.32	70.07	71.29	71.05	68.86	67.4	67.64	70.07	71.05	69.76%	256
	IAM	68.49	68.19	63.01	63.47	68.04	66.97	66.51	65.45	65.90	63.77	65.98%	
LTP	IFN/ENIT	82.72	82.97	82.48	83.45	82.97	82.48	80.29	80.53	82.24	83.21	82.33%	512
	IAM	73.06	69.25	70.17	70.01	72.60	72.91	73.51	72.75	70.93	68.49	71.37%	

Table 5.3: Identification rates on the CVL (310 writers) and AHTID/MW (53 writers) databases

Descriptor	Database	Split				Average accuracy	Dimension
		Sp.1	*Sp.2*	*Sp.3*	*Sp.4*		
BW-LBC	CVL	98.7	**99.03**	97.41	98.38	98.38%	49
	AHTID/MW	**100**	**100**	**100**	98.11	99.53%	16
LPQ	CVL	83.82	78.64	69.9	78.32	77.67%	256
	AHTID/MW	69.81	58.49	69.81	73.58	67.92%	
LBP	CVL	75.4	71.2	65.69	72.49	71.19%	256
	AHTID/MW	64.15	50.94	66.04	69.81	62.73%	
LTP	CVL	85.44	79.29	74.11	82.52	80.34%	512
	AHTID/MW	66.04	52.82	67.92	71.70	64.62%	

it achieves an identification rate of 82.33%, which is low compared to the one obtained with BW-LBC descriptor (96.47%).

Figure 5.7 shows the processing time (in seconds), including the time for feature extraction, distance calculation, and classification process by BW-LBC, LBP, LTP, and LPQ descriptors to identify a class over the four databases tested. It can be seen that the BW-LBC operator is very efficient in terms of feature extraction and classification execution time (CVL: 7.8s, IAM: 4.9s, IFN/ENIT: 4.1s, AHTID/MW: 2.9s), which is faster than the traditional LBP for all the tested databases (about 12 and 5.5 times faster than LBP for AHTID/MW and CVL databases, respectively). This performance is achieved thanks to the small BWLBC feature size (7x7=49 different patterns for the IFN/ENIT, CVL, and IAM databases and 4x4=16 for the AHTID/MW database). It is also evident from Figure 5.7 that the LTP descriptor is the most computationally expensive method compared to the BW-LBC, LBP, and LPQ methods, as it takes 105.7, 23.4, 14.8, and 79.15 seconds to identify the writer $n^{\circ}1$ from the CVL, IAM, IFN/ENIT and AHTID/MW databases, respectively. This limitation is mainly due to the dimensionality of the LTP features (512 possible patterns), which requires more time to compare the writing samples based on the Hamming distance of their respective feature histograms.

5.3.1.2/ IMPACT OF BW-LBC-PARAMETERS ON THE SYSTEM PERFORMANCE

The construction of the BW-LBC code requires that all extracted connected components be resized into the same uniform window size $W \times W$ and scanned from top to bottom and left to right to partition them into $N \times N$ non-overlapping blocks. Since the setting of these two parameters can significantly impact feature extraction performance, special attention should be paid when evaluating the BW-LBC operator. Therefore, it seems worthwhile to present the results of some key analytical experiments to quantify the impact of these two user-specific parameters (i.e., the window size $W \times W$ and the number of blocks $N \times N$) on

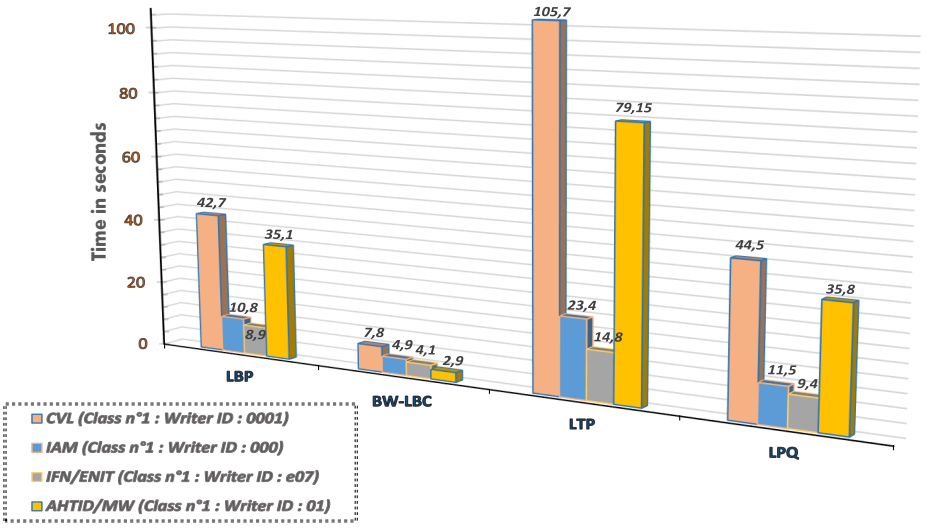


Figure 5.7: The processing time (in seconds) taken by the four tested descriptors to identify the writer $n°1$ (class $n°1$) from IAM (657 Writers), CVL (310 Writers), IFN/ENIT (411 Writers) and AHTID/MW (53 Writers) databases.

the performance of the proposed writer identification system. Note that all components are fitted into the same uniform window size $W \times W$ for both the training and test sets.

For each setting of the parameter $W \times W$, the number $N \times N$ is varied from 16 to 81 blocks, and for each pair of values ($W \times W$, $N \times N$), the average accuracy is recorded over ten random splits on the IAM and IFN/ENIT databases and over 4-fold cross-validation on the CVL and AHTID/MW databases. The effect of the parameter ($W \times W$, $N \times N$) on the system performance can be observed in Figures 5.8, 5.9, 5.10, and 5.11 for the IFN/ENIT, IAM, CVL, and AHTID/MW databases, respectively.

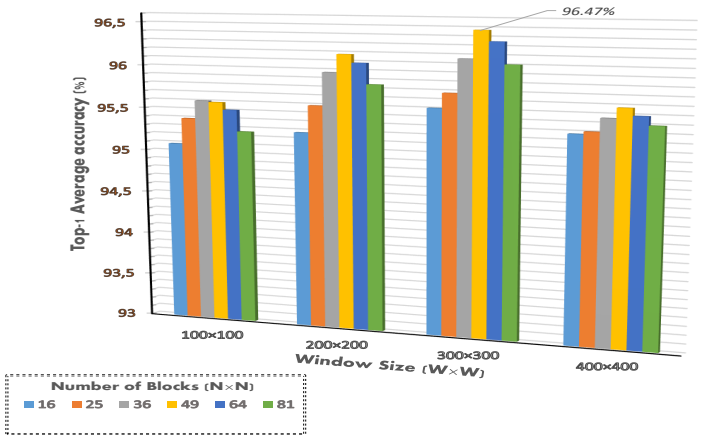


Figure 5.8: Performance stability as a function of BW-LBC-parameters on IFN/ENIT database.

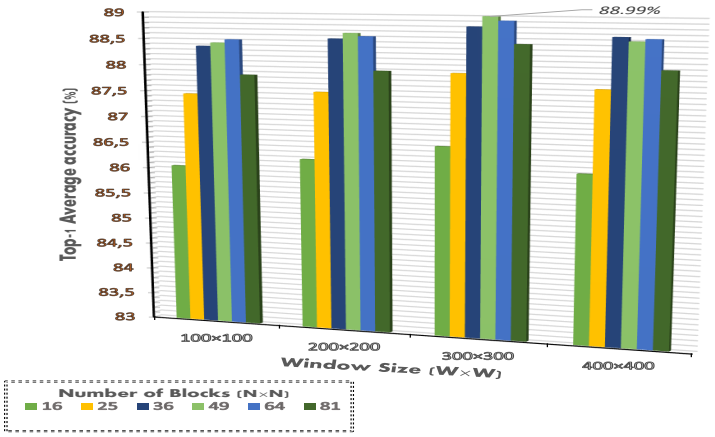


Figure 5.9: Performance stability as a function of BW-LBC-parameter on IAM databases.

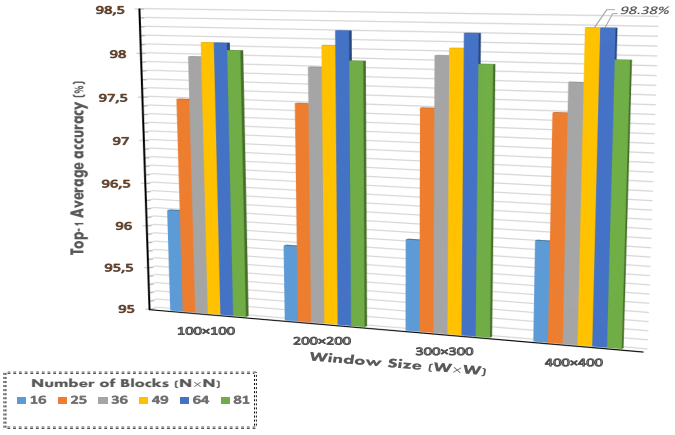


Figure 5.10: Performance stability as a function of BW-LBC-parameters on CVL database.

From the results shown in Figures 5.8 and 5.9, it can be seen that the pair of values (300×300 pixels, 7×7) is more suitable for characterizing the writing style for both the IFN/ENIT and IAM databases, as it allows recording the highest scores of 88.99% and 96.47%, respectively. The window size of 400×400 pixels with (N×N=49) and (N×N=64) prove to be the best configurations in the CVL database, as shown in Figure 5.10 since it gives an identification rate of 98.38%. On the AHTID/MW database, the highest identification rate (99.53%) is obtained with block numbers of 16 and 25, and this is independent of the window size, as shown in Figure 5.11. Moreover, it can be seen from Figure 5.11 that the window size W×W does not affect the identification rate since the BW-LBC descriptor seems to be matched to the word images available in the AHTID/MW database. The small size of the feature vector leads to a reduced computation time for the classification. Thus, since both the number of blocks (N×N=64) and (N×N=49) performed better

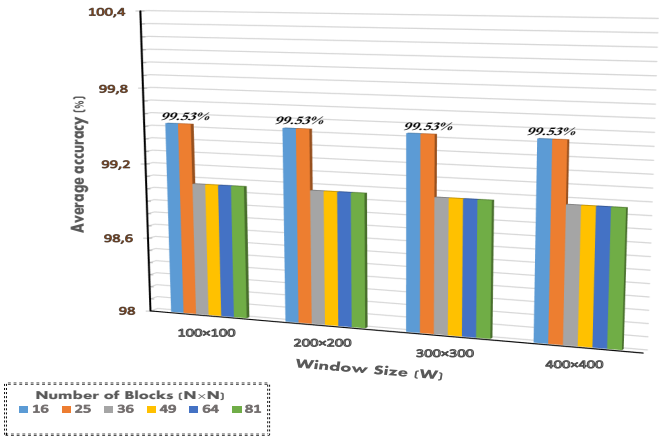


Figure 5.11: Performance stability as a function of BW-LBC-parameter on AHTID/MW database.

in the CVL database, the second value is set, which corresponds to a 49-dimensional feature vector. Similarly, the number of blocks of (N×N=16) is used as the optimal value in experiments for the AHTID/MW database. From the variation of the identification rate as a function of the number of blocks, it can be concluded that the block size should be sufficiently large to contain a significant amount of information about the writing style for better characterization of the writer to ensure acceptable identification performance.

5.3.1.3/ IMPACT OF THE NUMBER OF WRITERS ON THE SYSTEM PERFORMANCE

In this section, our motivation to perform a second set of experiments is to investigate the system performance's stability as a function of the number of writers. The idea is to analyze how the system behaves while varying the number of writers Nb from 10 to the complete set of writers in each database. It is worth noting that the average top-1 identification rate for each value of Nb and each descriptor is recorded over ten random subdivisions for the IAM and IFN-ENIT databases and after 4-fold cross-validation for the CVL and AHTID-MW databases. Figures 5.12, 5.13, 5.14 and 5.15 illustrate the obtained top-1 writer identification results. On the one hand, it can be observed that the higher the number of writers, the lower the system performance, especially when LBP, LTP, and LPQ are used as feature extraction methods. From the Figures 5.12, 5.13, 5.14 and 5.15, it can be seen that the performance of writer identification gradually decreases as the number of writers increases, which is quite normal since the classification is done by comparing the dissimilarity measure of each writer with those of a large number of writers (the complexity of identification between writers is gradually increased). The results obtained from these figures are consistent with the expected relative behavior of the proposed BW-LBC operator. It achieves the highest average accuracy on all tested databases and shows the

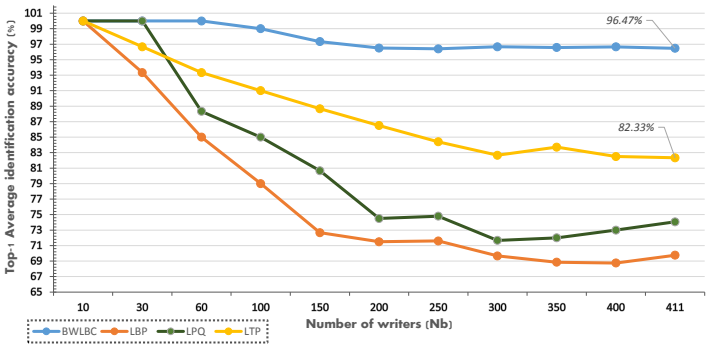


Figure 5.12: Writer identification rates under different numbers of writers on the IFN/ENIT database.

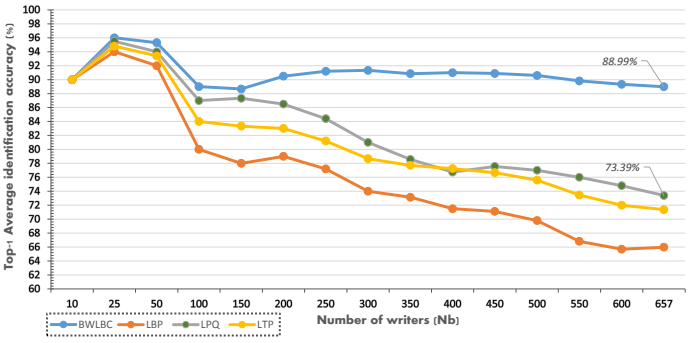


Figure 5.13: Writer identification rates under different numbers of writers on the IAM database.

consistent performance when the number of writers exceeds 150 and 200 writers on the IFN/ENIT and IAM databases, respectively. As for the CVL and AHTID/MW databases, the system performance shows significant performance stability for all Nb values over the evaluated LTP, LBP, and LPQ descriptors when using the BW-LBC model.

5.3.1.4/ Impact of the number of handwriting samples on the system performance

The main reason for performing this evaluation is to further assess the stability of system performance as a function of the amount of handwritten data used in the training and test sets. To this end, we conduct extensive experiments that record the accuracy of writer identification over a varying number of training images. On the IFN/ENIT and IAM databases, the writer identification process starts with at least 30% of the handwritten

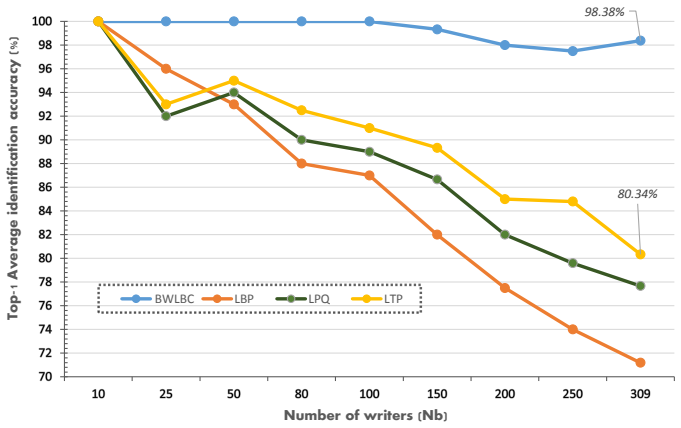


Figure 5.14: Writer identification rates under different numbers of writers on the CVL database.

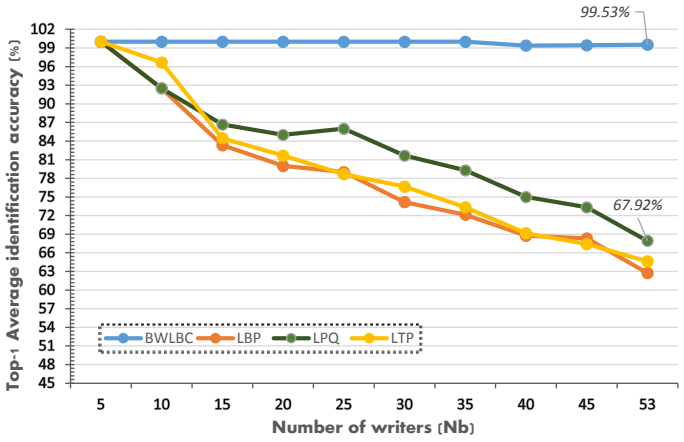


Figure 5.15: Writer identification rates under different numbers of writers on the AHTID/MW database.

images available per writer in the training set versus 70% in the evaluation set. We then gradually increase the training set until we reach a percentage of 70%. This setup cannot be applied to the remaining two databases (CVL and AHTID/MW) as they only offer three configuration possibilities. These include 75%/25% (i.e., 75% of handwritten images in the training set versus 25% in the test set) and 25%/75% configurations with average accuracy after 4-fold cross-validation. The final result in the case of the 50%/50% configuration (half-half setup) is recorded after 6-fold cross-validation (i.e., six permutations). This constraint is directly related to the number of samples used per writer where the AHTID/MW database is provisionally divided into four subsets, while the CVL database (setup-1) contains four documents for each writer. Tables 5.4 and 5.5 report the classification results obtained. It can be easily observed that the identification rates recorded

Table 5.4: Identification rates (in percentage) on the IFN/ENIT and IAM databases over a variable number of training and test handwritten images.

Descriptor	Database	Train setup (%) /Test setup (%)				
		30/70	***40/60***	***50/50***	***60/40***	***70/30***
BW-LBC	IAM	80.67	85.99	87.67	**89.49**	89.04
	IFN/ENIT	86.86	92.94	96.35	**96.59**	96.11
LPQ	IAM	64.53	70.01	71.54	73.51	74.43
	IFN/ENIT	64.96	70.07	73.96	75.18	73.72
LBP	IAM	56.01	65.29	64.99	65.9	64.84
	IFN/ENIT	57.18	65.45	70.08	71.05	68.61
LTP	IAM	62.71	69.86	70.01	70.93	72.3
	IFN/ENIT	70.8	77.13	82.24	83.21	81.99

Table 5.5: Identification rates (in percentage) on the CVL and AHTID/MW databases over a variable number of training and test handwritten images.

Descriptor	Database	Train setup (%) /Test setup (%)		
		25/75	***50/50***	***75/25***
BW-LBC	CVL	93.44	**98.76**	98.38
	AHTID/MW	95.75	**100**	99.53
LPQ	CVL	63.59	75.78	77.67
	AHTID/MW	54.24	70.44	67.92
LBP	CVL	61.41	70.76	71.19
	AHTID/MW	45.75	62.89	62.73
LTP	CVL	71.76	80.2	80.34
	AHTID/MW	49.05	69.49	64.62

for each descriptor decrease as the number of images in training set decreases. Obviously, the system needs more learning samples to train the classifier to perform the writer identification with high precision. In fact, the best results are obtained when there is an acceptable amount of handwritten samples in both training and test sets. On IAM and IFN/ENIT databases, the best performance (89.49 % for IAM and 96.59% for IFN/ENIT) is obtained when 60% of data available per writer is used in the training set, while the (50%/50%) setup allows to record the higher values of 98.76% and 100% for CVL and AHTID/MW databases, respectively.

Note that we assess all possible scenarios in each configuration of CVL and AHTID/MW databases (i.e., the 4- and 6-fold cross-validation) to demonstrate the effectiveness of our system on a variable number of samples.

5.3.2/ PERFORMANCE OF THE HANDCRAFTED-BASED APPROACH

5.3.2.1/ INFLUENCE OF THE NUMBER OF ZONES ($N_z \times N_z$) AND THE DIMENSIONALITY REDUCTION FACTOR (F) ON SYSTEM PERFORMANCE

The handcrafted-based approach is evaluated on IAM (text-line setup), IFN/ENIT, CVL (setup-1), and AHTID databases. A first analysis is conducted to investigate, through extensive experiments, the performance of the proposed identification system as a function of the feature extraction parameters. We defined, as indicated in Chapter 3 (cf. Section 3.2.2.2), the following two parameters: dimensionality reduction factor F and number of zones $N_z \times N_z$ of the normalized feature image ($\frac{\mathbf{FC}}{\mathbf{F}}$). The number $N_z \times N_z$ is set for each value of parameter F from 4 to 36 zones and the identification rate is recorded for each setting of (F, $N_z \times N_z$). Figures 5.16 and 5.17 show the impact of the feature extraction parameters on the overall system performance for the IFN/ENIT, IAM, CVL, and AHTID/MW databases. For CVL and AHTID/MW databases, the average accuracy after 4-fold cross-validation is plotted for each pair of values (F, $N_z \times N_z$) to assess the system performance.

Analyzing the classification rates shown in Figures 5.16 (a) and 5.17 (c), it is clear that the LPQ operator with the settings ($N_z \times N_z$=16, F=8), ($N_z \times N_z$=16, F=11) and ($N_z \times N_z$=16, F=14) has the highest scores of 97.81% and 98.62% on the IFN/ENIT and CVL databases, respectively. As summarized in Table 5.6, ($N_z \times N_z$=16, F=14) is set as the optimal parameter value for the IFN/ENIT and CVL databases, resulting in a reduced length of the final LPQ feature histogram (304 bins). Similarly, for LBP and LTP operators,

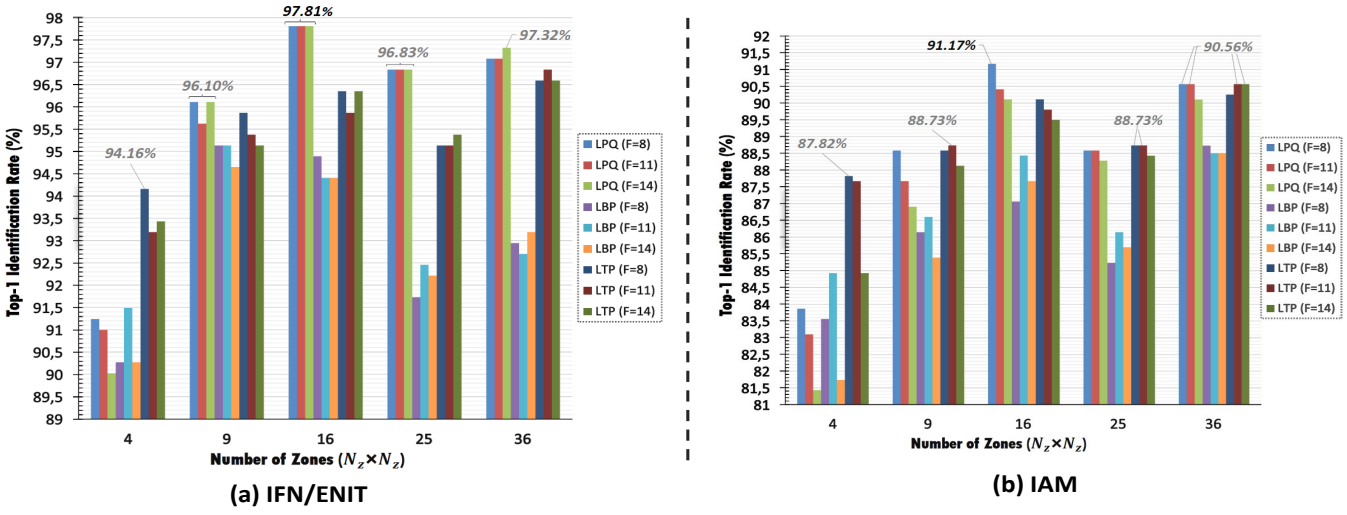


Figure 5.16: System performance as a function of the number of zones ($N_z \times N_z$) and dimensionality reduction factor (F) on IFN/ENIT and IAM databases.

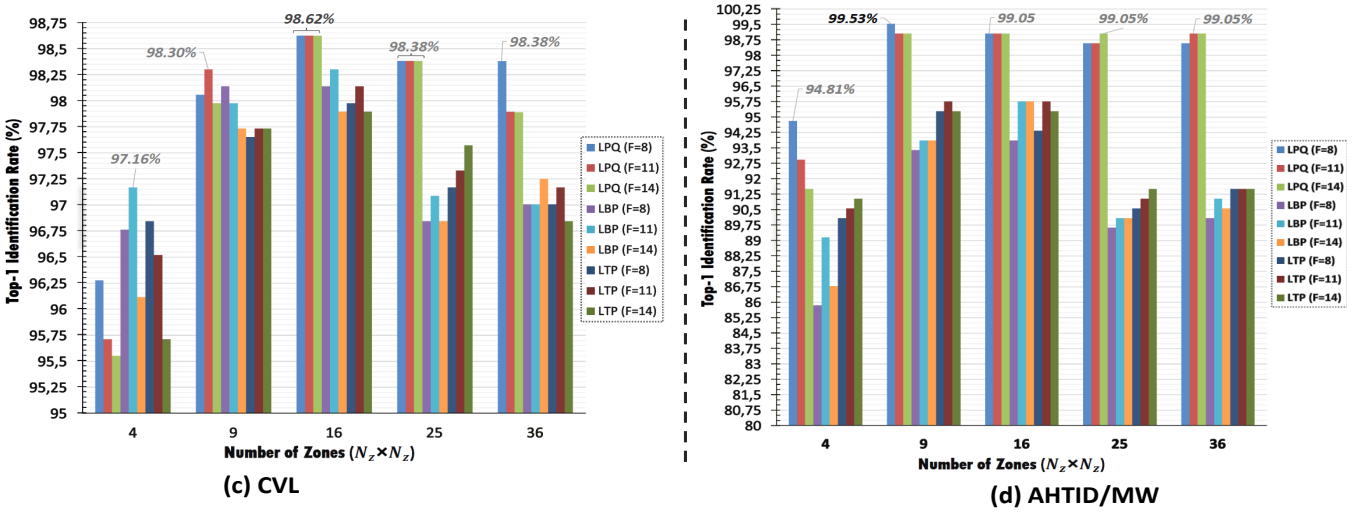


Figure 5.17: System performance as a function of the number of zones ($N_z \times N_z$) and dimensionality reduction factor (F) on CVL and AHTID/MW databases.

Table 5.6: Optimal ($N_z \times N_z$) and (F) parameter values for the 4 tested databases.

Database	Feature extraction method	Identification rate (%)	($N_z \times N_z$, F):[D_c]
IFN/ENIT	*LPQ*	97.81	(16, 8):[512] (16, 11):[384] **(16, 14):[304]**
	LTP	96.84	**(36, 11):[1692]**
	LBP	95.13	(9, 8):[512] **(9, 11):[216]**
IAM	*LPQ*	91.17	**(16, 8):[512]**
	LTP	90.56	(36, 11):[1692] **(36, 14):[1332]**
	LBP	88.73	**(36, 8):[1152]**
CVL	*LPQ*	98.62	(16, 8):[512] (16, 11):[384] **(16, 14):[304]**
	LTP	98.14	**(16, 11):[752]**
	LBP	98.30	**(16, 11):[384]**
AHTID/MW	*LPQ*	99.53	**(9, 8):[288]**
	LTP	95.75	**(9, 11):[423]** (16, 11):[752]
	LBP	95.75	(16, 11):[384] **(16, 14):[304]**

the optimal value of (N_z×N_z, F) is the one, that minimizes the dimensionality (D_c=(N_z×N_z)×($\frac{D_{im}}{F}$)) of the feature histogram, as shown in Table 5.6 (the lower the dimensionality D_c, the higher the dimensionality reduction factor). For the English IAM database, when LPQ is used as the feature extraction method, the setting ($N_z \times N_z$=16, F=8) as shown in Figure

5.16 (b) proves to be more suitable for characterizing the writing style as it realizes the higher identification rate of 91.17% with $D_c = 512$. On the Arabic AHTID/MW database, the setting ($N_z \times N_z$=9, F=8) using the LPQ operator proved to be the best configuration, yielding an average top-1 accuracy of 99.53% with $D_c = 288$. From these results, the variation in identification rate as a function of the number of zones $N_z \times N_z$ suggests that the feature zones of the normalized feature image ($\frac{FC}{F}$) should be sufficiently wide to include more discriminative measures in characterizing writer individuality for acceptable identification performance.

5.3.2.2/ CLASSIFICATION RESULTS

Table 5.7 summarizes the average top-1 identification rates of the proposed system over the four tested splits on CVL and AHTID/MW databases, and the top-1, top-3, and top-5 writer identification rates on IFN/ENIT and IAM databases are shown in Table 5.8. As can be seen from these tables, the identification rates recorded by the evaluated texture descriptors are more or less consistent across the four tested databases. Furthermore, as shown in Table 5.7, the evaluated descriptors show significant performance stability across the four splits of the CVL and AHTID/MW databases, with identification rates tending to be close to average accuracy. Comparing the effectiveness of the tested descriptors in characterizing the handwritten images, the LPQ operator consistently emerges as the best descriptor, providing high identification rates in all tested databases. The results presented in Tables 5.7 and 5.8 correspond to the best configurations of $N_z \times N_z$ and F, which can be used to obtain the highest identification rates. The optimal settings of these parameters are given as follows:

- In the CVL database, the best average accuracy (98.62% using the LPQ descriptor) is recorded with the number of zones $N_z \times N_z = 16$ and the dimensionality reduction factor $F = 14$, yielding 304 bins of the final LPQ feature histogram (according to D_c=$(N_z \times N_z) \times (\frac{\mathrm{D_{im}}}{F})$).
- The number of zones $N_z \times N_z = 9$ with dimensionality reduction factor $F = 8$ allows the LPQ descriptor to achieve the highest performance (score of 99.53% with $D_c = 288$) on the AHTID/MW database.
- In the IFN/ENIT database, the best top-1 identification rate of 97.81% is obtained when the LPQ descriptor is used as a feature extraction method with a number of zones of $N_z \times N_z = 16$ and $F = 14$ for the dimensionality reduction factor ($D_c = 304$).
- Using the LPQ descriptor, the setting ($N_z \times N_z$=16, F=8) proved to be the best configuration for characterizing the writing style on the IAM database, achieving the best top-1 identification rate of 91.17% with $D_c = 512$.

Table 5.7: Writer identification rates of the proposed system on CVL (310 writers) and AHTID/MW (53 writers) databases

Feature extraction method	Database	Split				Average accuracy	Dimension(D_c)
		Sp.1	*Sp.2*	*Sp.3*	*Sp.4*		
LPQ	CVL	99.03	**99.35**	97.41	98.70	98.62%	304
	AHTID/MW	**100**	**100**	**100**	98.11	99.53%	288
LTP	CVL	98.70	**99.35**	96.11	98.38	98.14%	752
	AHTID/MW	98.11	96.22	98.11	90.56	95.75%	423
LBP	CVL	98.38	**99.35**	97.41	98.06	98.30%	384
	AHTID/MW	96.23	96.23	98.11	92.45	95.75%	304

Table 5.8: Writer identification rates of the proposed system on IFN/ENIT (411 writers) and IAM (657 writers) databases

Feature extraction method	Database	Identification rate (%)			Dimension (D_c)
		Top-1	***Top-3***	***Top-5***	
LPQ	IFN/ENIT	**97.81**	99.51	100	304
	IAM	**91.17**	94.21	96.04	512
LTP	IFN/ENIT	**96.84**	98.78	99.27	1692
	IAM	**90.56**	94.21	95.43	1332
LBP	*IFN/ENIT*	**95.13**	98.78	100	216
	IAM	**88.73**	93.15	93.91	1152

Tables 5.9, 5.10, 5.11, and 5.12 show the top 1 identification rates along with the processing time for each feature extraction method and under different values of parameter F on a subset of writers (150 writers) from the IAM, IFN/ENIT, and CVL databases, and on 53 writers from AHTID/MW database.

In this experiment, the optimal value of the parameter $N_z \times N_z$ is determined for each feature extraction method and each database. It can be seen that there is a general trend of increasing the identification rate, depending on the descriptor and dimensionality reduction factor F used.

The processing time for the classification of 150 writers from IAM and IFN/ENIT gradually decreases as the dimensionality reduction factor (F) increases, which is natural as it corresponds to the decrease in the final dimensionality of the feature histogram, i.e., high vector dimensionality requires more time to compare the writing samples based on the Hamming distance of their respective feature vectors. The best results (identification rates and processing time) are shown in bold in Tables 5.9, 5.10, 5.11, and 5.12. From

Table 5.9: System performance as a function of dimensionality reduction factor (F) on IAM database (150 writers)

Feature extraction method	Dimensionality reduction factor (F)	Identification rate (%)	Processing time (in seconds)	Number of zones $N_z \times N_z$	Dimension (D_c)
LPQ	1	95.33	17293	16	4096
	8	94	1642.2	16	512
	11	94	1187.4	16	384
	14	**95.33**	**993.23**	**16**	**304**
LTP	1	92.66	39803	36	18432
	8	92.66	3924.2	36	2304
	11	**92.66**	**2378.9**	**36**	**1692**
	14	92	1725.7	36	1332
LBP	1	92	15271	36	9216
	8	92	1171.7	36	1152
	11	92	695.5599	36	864
	14	**92**	**478.1355**	**36**	**684**

Table 5.10: System performance as a function of dimensionality reduction factor (F) on IFN/ENIT database (150 writers)

Feature extraction method	Dimensionality reduction factor (F)	Identification rate (%)	Processing time (in seconds)	Number of zones $N_z \times N_z$	Dimension (D_c)
LPQ	1	99.33	9442.8	16	4096
	8	98.66	720.32	16	512
	11	99.33	369.25	16	384
	14	**99.33**	**180.50**	**16**	**304**
LTP	1	98.66	19983	36	18432
	8	98.66	2296.1	36	2304
	11	98	1546.2	36	1692
	14	**98.66**	**1193.2**	**36**	**1332**
LBP	1	97.33	5059.7	9	2304
	8	96.66	165.31	9	288
	11	97.33	131.71	9	216
	14	**97.33**	**104.23**	**9**	**171**

these results, we can deduce that with the factor F (where F $>$ 1), we achieve the same identification rates (on IAM and IFN/ENIT) as without dimensionality reduction ($F = 1$) with lower processing time. For example, considering the LPQ operator, which gives top results in the IAM database, the highest score of 95.33 % is recorded for both $F = 1$ and $F = 14$, and the computation time is significantly reduced with a rate of 17.41% with F=14.

As shown in Tables 5.11 and 5.12, the best results (identification rates and processing time) in CVL and AHTID/MW databases are obtained with the dimensionality reduction factor F (F $>$ 1).

Table 5.11: System performance as a function of dimensionality reduction factor (F) on CVL database (150 writers)

Feature extraction method	Dimensionality reduction factor (F)	Identification rate (%)	Processing time (in seconds)	Number of zones $N_z \times N_z$	Dimension (D_c)
LPQ	1	98.66	112820	16	4096
	8	**98.66**	**13332**	**16**	**512**
	11	97.33	8688.5	16	384
	14	97.33	6938.7	16	304
LTP	1	96.66	211760	16	8192
	8	97.33	25894	16	1024
	11	98	19690	16	752
	14	**98**	**15397**	**16**	**592**
LBP	1	98	106600	16	4096
	8	**98.66**	**12773**	**16**	**512**
	11	98	86675	16	384
	14	98	6323.6	16	304

Table 5.12: System performance as a function of dimensionality reduction factor (F) on AHTID/MW database (53 writers)

Feature extraction method	Dimensionality reduction factor (F)	Identification rate (%)	Processing time (in seconds)	Number of zones $N_z \times N_z$	Dimension (D_c)
LPQ	1	92.45	11475	9	2304
	8	**98.11**	**1257.7**	**9**	**288**
	11	96.22	863.08	9	216
	14	96.22	629.75	9	171
LTP	1	77.35	23208	9	4608
	8	88.67	2920.6	9	576
	11	**90.56**	**2149.8**	**9**	**423**
	14	88.67	1693	9	333
LBP	1	81.13	9175	16	4096
	8	86.79	1027.7	16	512
	11	88.67	633.08	16	384
	14	**92.45**	**399.74**	**16**	**304**

5.3.2.3/ STABILITY OF THE SYSTEM PERFORMANCE AS A FUNCTION OF THE NUMBER OF WRITERS

This experiment evaluates the stability of the system performance according to a different number of writers. For this purpose, we studied the system behavior by varying the number of writers N_w from 10 to the total amount of writers for each database. Similar to the previous experiments, the full 4-fold cross-validation is used in CVL and AHTID/MW databases, recording the average top-1 identification rate for each descriptor and each value of N_w. The top-1 identification accuracy vs. the number of writers N_w for the IFN/ENIT, IAM, CVL, and AHTID/MW databases is shown in Figures 5.18, 5.19, 5.20, and 5.21, respectively. For all tested descriptors, it can be seen that the identification performance gradually decreases as the number of writers grows. For example, considering the IFN/ENIT database, the identification rate starts at 100% for 30 writers and drops to

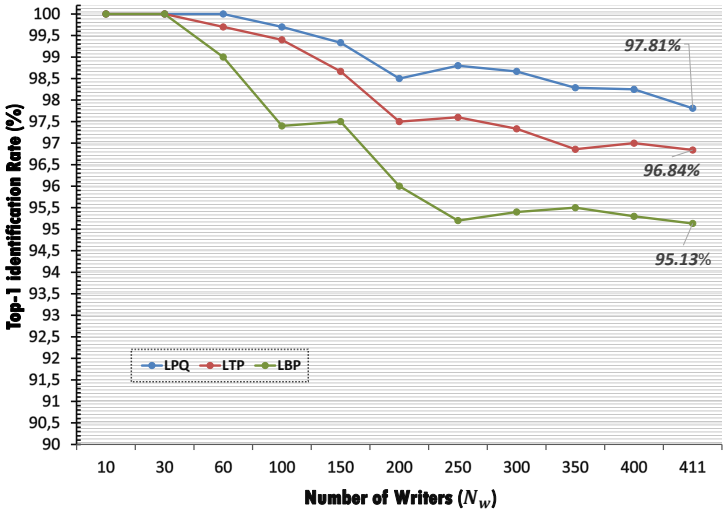


Figure 5.18: Identification rate vs. number of writers ranging from 10 to 411 writers on IFN/ENIT database.

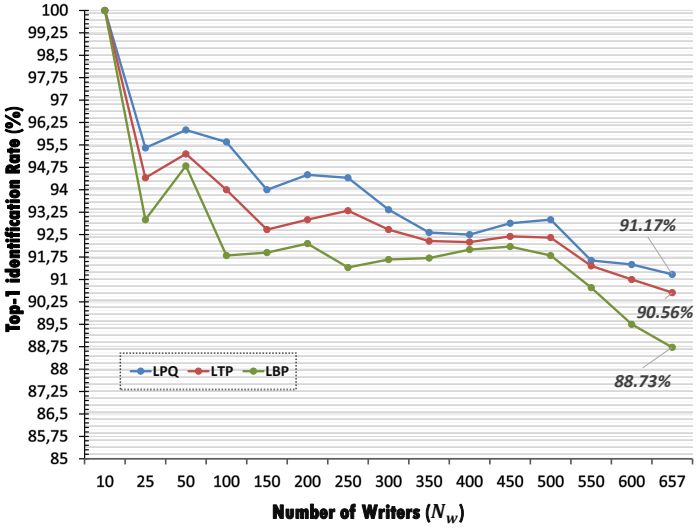


Figure 5.19: Identification rate vs. number of writers ranging from 10 to 657 writers on IAM database.

97.81%, 96.84%, and 95.13% for a total of 411 writers when LPQ, LTP, and LBP are used as feature extraction methods, respectively. This slight drop in performance is mainly due to the complexity of classification, as the dissimilarity measure of each writer is compared with those of a large number of writers and a large number of handwritten samples in both the test and training sets (writer identification becomes much more difficult when the number of classes is increased).

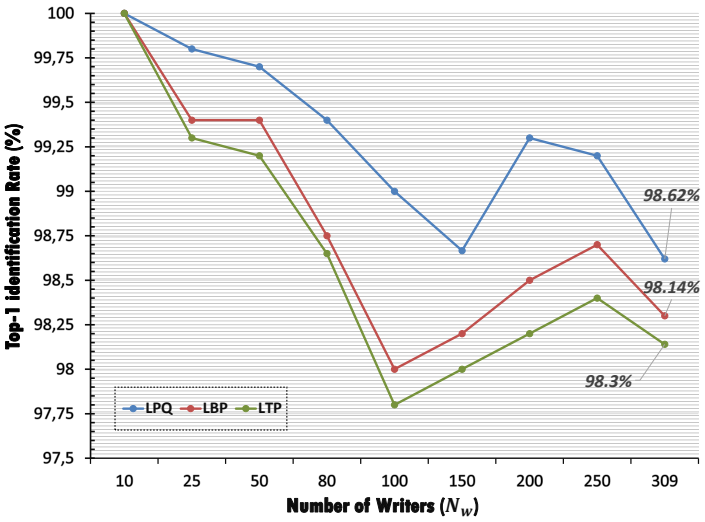


Figure 5.20: Identification rate vs. number of writers ranging from 10 to 310 writers on CVL database.

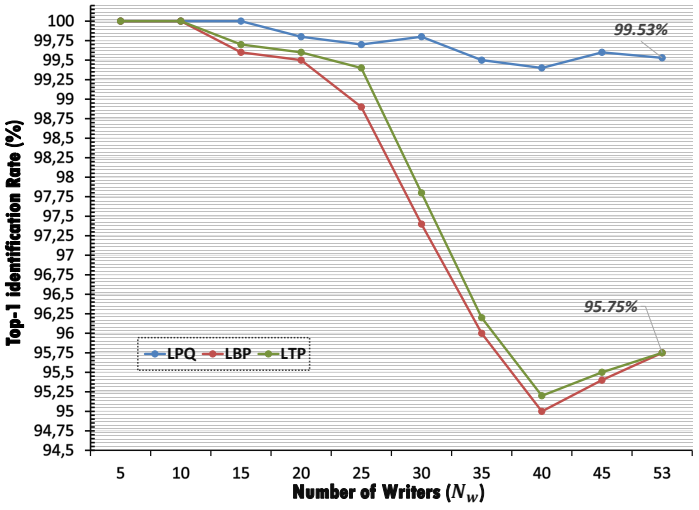


Figure 5.21: Identification rate vs. number of writers ranging from 10 to 53 writers on AHTD/MW database.

With LPQ, as expected, the overall system achieves the highest identification rates on the complete set of writers over each tested database. Moreover, it shows significant and consistent performance on the AHTID/MW database for all N_w values (cf. Figure 5.21).

5.3.2.4/ STABILITY OF THE SYSTEM PERFORMANCE AS A FUNCTION OF THE NUMBER OF HANDWRITING SAMPLES

This experiment evaluates the stability of the system performance on a different number of handwritten samples (used in the training and test sets). In other words, we empirically record the writer identification rate corresponding to different training and test partitions for each tested database. Then, for IFN/ENIT and IAM databases, the classification process is performed using 70% of the handwritten images available per writer in the test set, and the remaining 30% are used as the training set. After that, the number of samples in the training set is progressively increased by 10%, until 70% is reached. As for the CVL and AHTID/MW databases, we cannot use the training/testing partition of (30%/70% up to 70%/30%) because the AHTID/MW database is divided into 4 subsets in advance, while the CVL database (setup-1) contains four documents for each writer. For this reason, we could only apply the following configurations: 1) 75%/25% (i.e., 75% of handwritten images in the training set versus 25% in the test set) and 25%/75% partitions with an average accuracy recorded after 4-fold cross-validation; 2) 50%/50% partition (half-half configuration) using a 6-fold cross-validation setup (i.e., 6 possible permutations). The obtained classification results are summarized in Tables 5.13 and 5.14. As depicted in Table 5.13, the highest identification performance (for all tested feature extraction operators) on the IFN/ENIT and IAM databases is obtained when an appropriate amount of handwritten samples is used in both the test and training sets (partitioning of 60%/40%). For example, in the case of the LPQ operator, using 60% of the handwritten samples available per writer in the training set, the system achieves a score of 97.81% and 91.17% on the IFN/ENIT and IAM databases, respectively. The same observation can be made in Table 5.14. The partition of (75%/25%) performed better on CVL (best score of 98.62% using LPQ) and AHTID/MW (best score of 99.53% using LPQ) databases for all evaluated feature extraction operators.

Table 5.13: Identification rate (in percentage) on the IFN/ENIT and IAM databases over different training /testing partitions.

Feature extraction method	**Database**	**Partition (training (%)/testing (%))**				
		30/70	***40/60***	***50/50***	***60/40***	***70/30***
LPQ	*IAM*	81.43	87.21	89.34	**91.17**	89.95
	IFN/ENIT	88.56	93.67	97.32	**97.81**	96.59
LTP	*IAM*	81.28	86.45	87.36	**90.56**	89.34
	IFN/ENIT	87.10	93.43	95.13	**96.84**	96.10
LBP	*IAM*	77.47	82.64	85.08	**88.43**	86.30
	IFN/ENIT	83.45	90.51	94.16	**95.13**	94.65

Table 5.14: Identification rate (average in percentage) on CVL and AHTID/MW databases over different training /testing partitions.

Feature extraction method	Database	Partition (training (%)/testing (%))		
		25/75	***50/50***	***75/25***
LPQ	*CVL*	92.80	98.24	**98.62**
	AHTID/MW	95.28	99.37	**99.53**
LTP	*CVL*	89.4	97.46	**98.14**
	AHTID/MW	91.51	95.59	**95.75**
LBP	*CVL*	90.37	97.78	**98.30**
	AHTID/MW	82.55	93.39	**95.75**

5.3.3/ PERFORMANCE OF THE CLGP-BASED APPROACH

5.3.3.1/ RESULTS AND ANALYSIS

Comprehensive experiments are conducted on six handwritten databases to evaluate the performance and effectiveness of the proposed CLGP-based approach. The databases studied are: IFN/ENIT (411 writers/Arabic), AHTID/MW (53 writers/Arabic), CVL (setup-1 with 310 writers/English), IAM (text line setup with 657 writers/English), Firemaker (250 writers/Dutch) and ICDAR2011 (26 writers/hybrid language). On CVL and AHTID/MW databases, a full 4-fold cross-validation is performed, generating four split permutations for each writer. To illustrate the ability of the CLGP (Cross multi-scale Locally encoded Gradient Patterns) feature method, its feasibility and effectiveness in characterizing the large variability of handwriting, we compared its performance with the block-wise local binary count (BW-LBC) operator (Chahi et al. (2018)), LBP (Ojala et al. (2002)), LTP (Tan and Triggs (2010)), LETRIST (Song et al. (2017)), and LPQ (Ojansivu and Heikkilä (2008)) feature methods.

Table 5.15 shows the top-1, top-3, and top-5 writer identification rates recorded by the CLGP method along with those from LETRIST, LPQ, LTP, LBP, and BW-LBC on the IFN/ENIT, IAM, Firemaker, and ICDAR2011 databases. Table 5.16 depicts the average top-1 identification rates over the four splits tested on the AHTID/MW and CVL databases. The CLGP operator performs impressively and systematically best over the evaluated descriptors on all tested databases. Moreover, it can be seen from Table 5.16 that the CLGP operator shows consistent classification performance over all splits on the CVL and AHTID/MW databases as the identification accuracies converge to their average.

As stated in Chapter 3 (cf. Section 3.2.2.3), the CLGP is a $(N_{cell} \times 9 \times N_b \times (N_\sigma = 3))$-dimensional image feature descriptor. Indeed, setting the number of blocks N_b and the number of cells N_{cell} has an impact on the overall performance of the system. The CLGP classification results reported in Tables 5.15 and 5.16 correspond to the optimal setup of these two parameters (N_b and N_{cell}), which are empirically determined for each tested

Table 5.15: Classification rates on IFN/ENIT (411 writers), IAM (657 writers), Firemaker (250 writers), and ICDAR2011 (26 writers) databases. The highest classification scores are in bold.

Feature representation method	Database	Classification accuracy (%)		
		Top-1	***Top-3***	***Top-5***
CLGP	IFN/ENIT	**98.54**	100	100
	IAM	**94.06**	97.25	99.23
	Firemaker	**97.60**	98.40	99.2
	ICDAR2011	**100**	100	100
BW-LBC	IFN/ENIT	97.56	99.27	100
	IAM	90.11	93.15	94.98
	Firemaker	94.40	97.60	98.40
	ICDAR2011	97.43	98.71	100
LPQ	IFN/ENIT	75.42	77.12	78.08
	IAM	75.49	78.68	80.66
	Firemaker	37.20	54.40	64
	ICDAR2011	88.46	98.71	98.71
LTP	IFN/ENIT	83.45	85.63	86.59
	IAM	73.51	76.85	78.68
	Firemaker	30.4	45.6	54
	ICDAR2011	82.05	96.15	100
LBP	IFN/ENIT	71.29	74.43	76.11
	IAM	68.49	72.43	75.01
	Firemaker	33.60	48	55.60
	ICDAR2011	79.48	96.15	98.71
LETRIST	IFN/ENIT	77.85	80.29	81.99
	IAM	79.14	80.66	82.19
	Firemaker	35.60	52	60.4
	ICDAR2011	85.89	97.43	98.71

Table 5.16: Classification rates on CVL (310 writers) and AHTID/MW (53 writers) databases. The highest classification scores are in bold

Feature representation method	Database	Split				Average accuracy
		Sp.1	*Sp.2*	*Sp.3*	*Sp.4*	
CLGP	CVL	99.35	**99.67**	**99.67**	99.35	99.51%
	AHTID/MW	**100**	**100**	**100**	98.11	99.53%
BW-LBC	CVL	98.7	99.03	97.41	98.38	98.38%
	AHTID/MW	**100**	**100**	**100**	98.11	99.53%
LPQ	CVL	83.82	78.64	69.9	78.32	77.67%
	AHTID/MW	69.81	58.49	69.81	73.58	67.92%
LBP	CVL	75.4	71.2	65.69	72.49	71.19%
	AHTID/MW	64.15	50.94	66.04	69.81	62.73%
LTP	CVL	85.44	79.29	74.11	82.52	80.34%
	AHTID/MW	66.04	52.82	67.92	71.70	64.62%
LETRIST	CVL	85.16	78.70	73.22	81.29	79.59%
	AHTID/MW	67.92	54.71	69.81	69.81	65.56 %

database through a comprehensive evaluation described in detail later in this section. The number of blocks $N_b = 4$ with $N_{cell} = 9$ are considered as optimal settings for IAM (result of 94.06%), CVL (average result of 99.51%), Firemaker (result of 97.60%), ICDAR2011

(score of 100%), and AHTID/MW (average score of 99.53%), resulting in 972 bins of the final CLGP feature histogram (according to $D_{im} = N_{cell} \times 9 \times N_b \times (N_\sigma = 3)$). As for the IFN/ENIT database, the best top-1 identification accuracy of 98.54% is obtained when the CLGP method is used as a feature extraction operator with a number of blocks $N_b = 1$ (segmentation into blocks is unnecessary) and a number of cells $N_{cell} = 16$. In this case, we obtain a reduced length of the CLGP feature histogram ($D_{im} = N_{cell} \times 9 \times 1 \times (N_\sigma = 3) = 432$). The processing time in seconds taken by the proposed system to classify a writer (class) using CLGP with different feature methods on IAM, IFN/ENIT, CVL, Firemaker, AHTID/MW, and ICDAR2011 databases is shown in Figure 5.22.

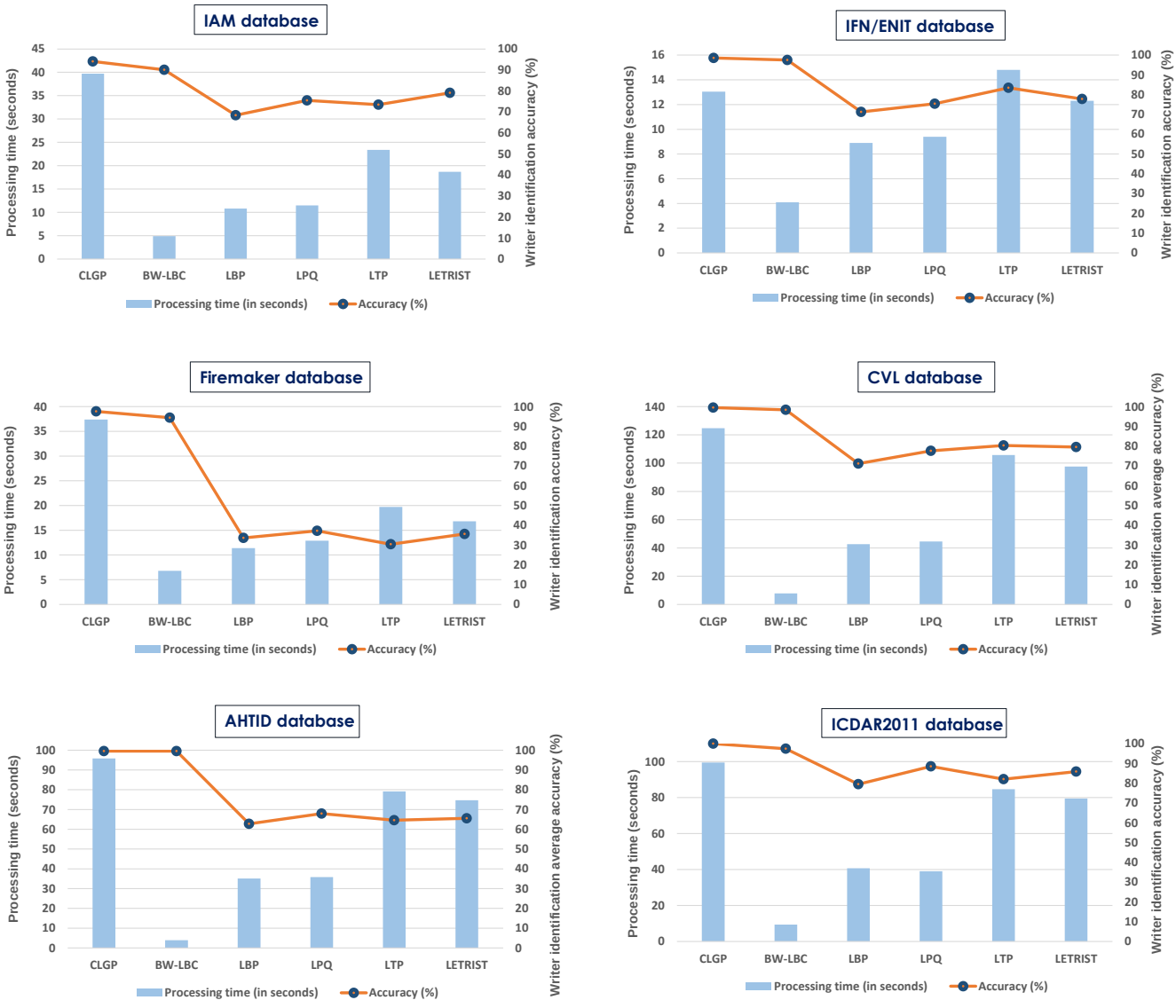


Figure 5.22: The processing time (in seconds) required by the proposed system to identify writer N°1 (class N°1) from IAM, Firemaker, CVL, IFN/ENIT, ICDAR2011, and AHTID/MW databases.

It can be seen that the recorded processing time increases when evaluating feature methods with high histogram dimensions. This limitation is quite natural since the overall system needs more time to compare and classify the writing samples when their respective Hamming distances are used. In this experiment, the BW-LBC descriptor is the computationally fastest method (IAM: 4.9s, CVL: 7.8s, AHTID/MW: 3.9s, IFN/ENIT: 4.1s, Firemaker: 6.8s, ICDAR2011: 9.4s) thanks to its small histogram size (49 different patterns over all tested databases). From Figure 5.22, the CLGP feature method significantly outperforms all other tested feature methods (in terms of identification accuracy) across all evaluated databases. The CLGP certainly takes more time to classify writers compared to the BW-LBC descriptor. However, the processing time is not necessarily an important performance indicator for offline writer identification since no real-time applications are required (offline mode). Moreover, none of the old and current state-of-the-art systems have specified their total evaluation processing time.

5.3.3.2/ CLGP-KEY PARAMETERS ANALYSIS

This section presents the results of a comprehensive experiment to evaluate the system's overall performance with respect to the number of blocks N_b and the number of cells N_{cell}. These two settings are the key user-defined parameters of the CLGP feature method. The identification accuracy is recorded for each pair of values (N_b, N_{cell}), where the parameter N_b is set for each setting of the parameter N_{cell} from 1 to 16 blocks. Figures 5.23, 5.24, and 5.25 show the system performance as a function of (N_b, N_{cell}) on the IFN/ENIT, IAM, CVL, Firemaker, ICDAR2011, and AHTID/MW databases, respectively. In order to assess all possible scenarios and evaluate the system stability performance on CVL and AHTID/MW databases, the average accuracy is reported according to the 4-fold cross-validation scheme for each (N_b, N_{cell}) setting.

From the results plotted in Figure 5.23 (a), we can clearly see that the highest identification rate in the IFN/ENIT database (score of 98.54%) is achieved with ($N_b = 1 \times 1, N_{cell} = 16$) and ($N_b = 2 \times 2, N_{cell} = 16$). Since these two parameter settings perform better on the IFN/ENIT database, the first one corresponding to a 432-dimensional CLGP feature histogram is selected, resulting in reduced classification computation time.

As shown in Figures 5.23 (b) and 5.24 (c), ($N_b = 2 \times 2, N_{cell} = 9$) and ($N_b = 2 \times 2, N_{cell} = 16$) turn out to be the best settings on the IAM and Firemaker databases with scores of 94.06% and 97.6%, respectively. Therefore, as with the IFN/ENIT database, the setting ($N_b = 2 \times 2, N_{cell} = 9$) is set as the optimal value, resulting in a CLGP feature dimension of 972 according to $D_{im} = N_{cell} \times 9 \times 1 \times (N_\sigma = 3)$. Similarly, ($N_b = 2 \times 2, N_{cell} = 9$) and ($N_b = 3 \times 3, N_{cell} = 9$) parameter configurations seem to be empirically the appropriate way to characterize the writing style on the AHTID/MW database, as they allow to achieve

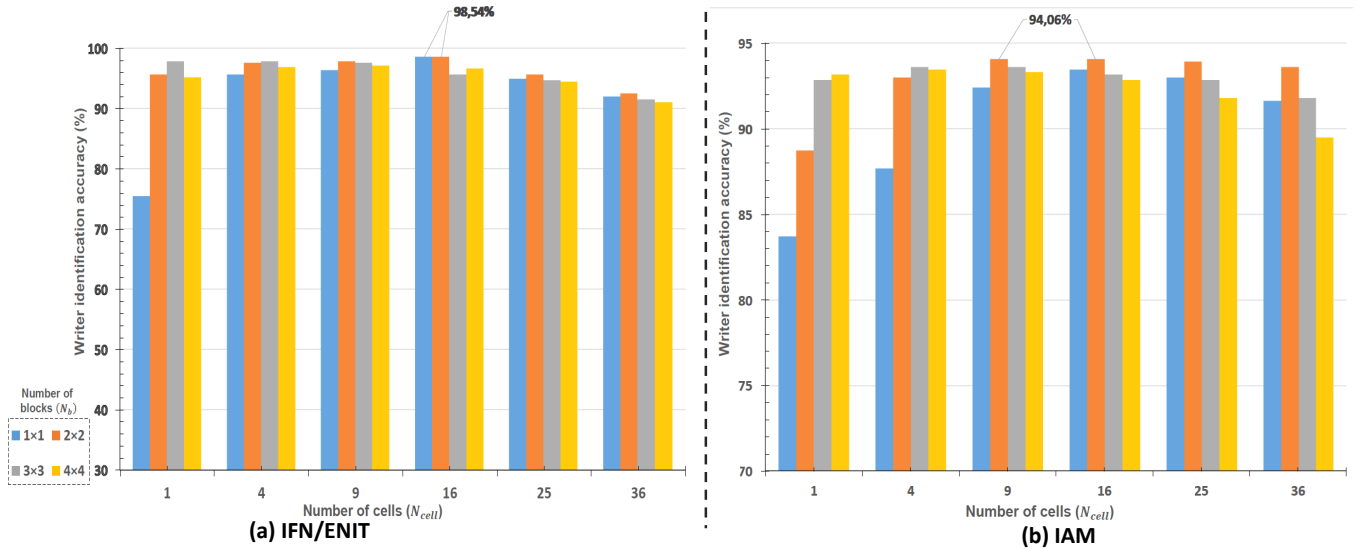


Figure 5.23: System performance with respect to the number of blocks N_b and number of cells N_{cell} on IFN/ENIT and IAM databases.

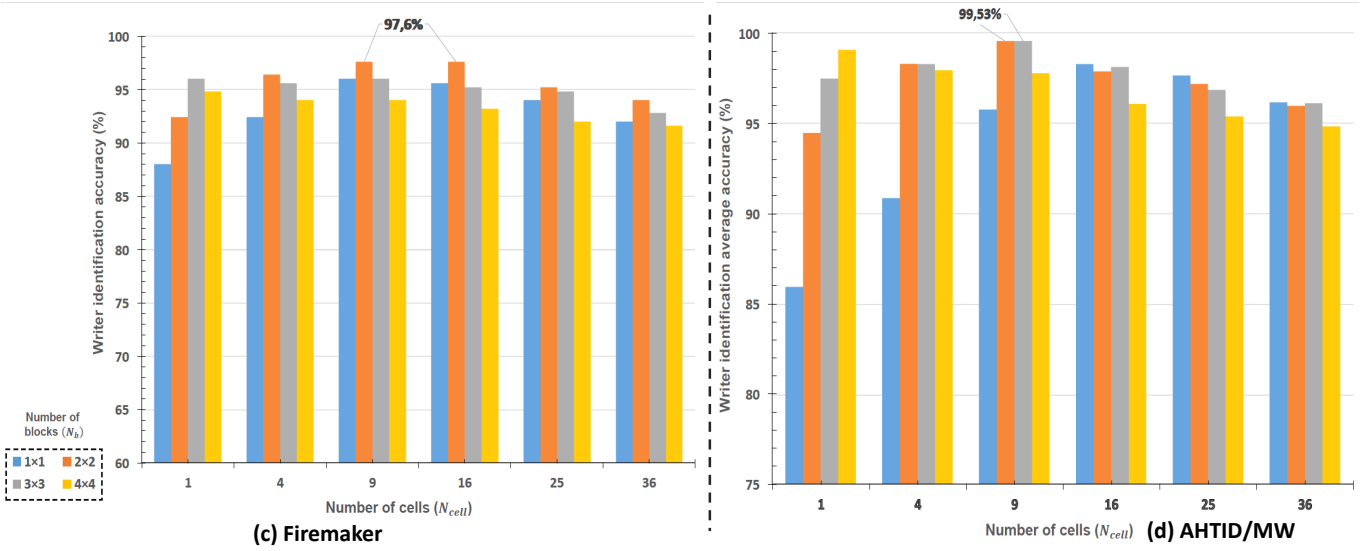


Figure 5.24: System performance with respect to the number of blocks N_b and number of cells N_{cell} on Firemaker and AHTID/MW databases.

the highest average score of 99.53% (cf. Figure 5.24 (d)). Obviously, the first parameter value is chosen to obtain a reduced length of the CLGP feature histogram (972 bins).

In the case of the ICDAR2011 database (cf. Figure 5.25 (e)), the highest identification rate of 100% is recorded with the settings ($N_b = 2 \times 2, N_{cell} = 9$) and ($N_b = 4 \times 4, N_{cell} = 4$). The optimal value of ($N_b = 2 \times 2, N_{cell} = 9$) is set as the one that minimizes the dimensionality of the CLGP feature histogram. In the CVL database, the setting ($N_b = 2 \times 2, N_{cell} = 9$)

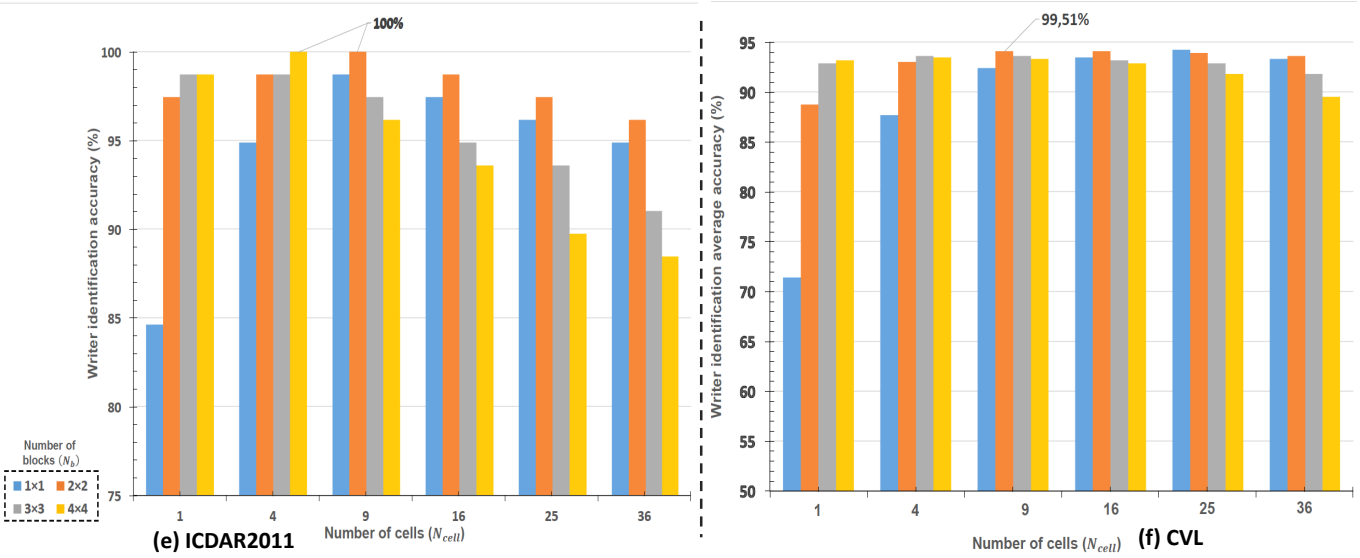


Figure 5.25: System performance with respect to the number of blocks N_b and number of cells N_{cell} on ICDAR2011 and CVL databases.

allows the highest performance with an average accuracy of 99.51%, as shown in Figure 5.25 (f). The variation in the classification rate is due to the number of blocks and the number of cells. For acceptable performance in writer identification, we use uniform block and cell sizes to capture a considerable amount of writing features, i.e., the window size must be wide enough and appropriate to ensure better characterization of writing style in it.

5.3.3.3/ Stability of the system performance according to the number of writers

Through this experiment, we evaluate and study the stability of the system performance by varying the number of writers N_{bw} from 10 to the total number of writers. The top-1 identification accuracy for the number of writers N_{bw} on the IAM database (text line setup) is shown in Figure 5.26. For the BW-LBC and CLGP feature methods, it can be seen that a gradual increase in the number of writers leads to a regular and slight decrease in the system performance. However, for LETRIST, LBP, LPQ, and LTP, a sharp decrease in classification performance is observed when the number of writers increases. The two best-performing feature methods are CLGP followed by BW-LBC, all of which provide high classification accuracy starting at 96% for 25 writers and declines to 94.06% and 90.11% for 657 writers, respectively. Concerning LETRIST, LPQ, LTP, and LBP, a classification accuracy of 90% is recorded using $N_{bw} = 10$, and thereafter the performance acutely starts to drop to 79.14%, 75.49%, 73.51% and 68.49% respectively for a total of 657 writers. The same system behavior, i.e., performance drop, is observed in the other five

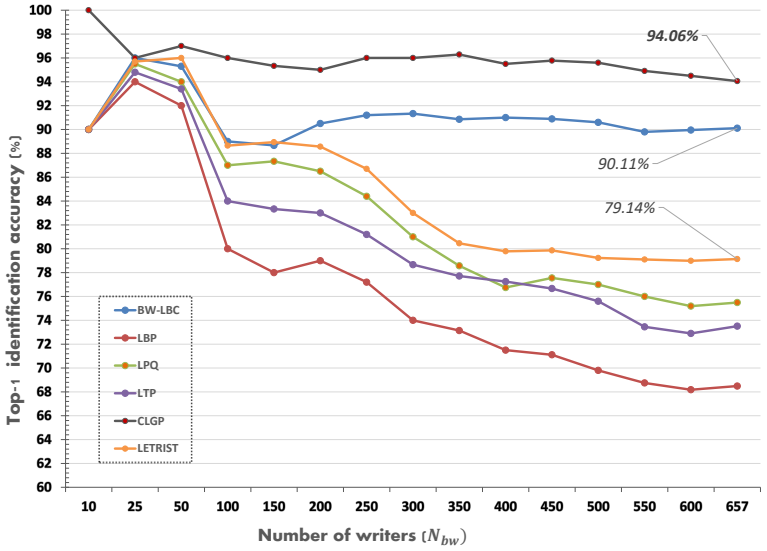


Figure 5.26: System performance with respect to the number of writers ranging from 10 to 657 writers on IAM database.

databases tested. The reason for this is the complexity of classification when comparing the dissimilarity measure of each writer with those of a large number of writers, i.e., a large amount of available handwritten data in the test and training datasets. In this case, the classification process gradually becomes more difficult as the number of classes (writers) increases. The classification results reported in Figure 5.26 are congruent with the classification results shown in Table 5.15. The proposed CLGP achieves the highest classification rates, making it the best performing method compared to the evaluated feature methods.

5.3.3.4/ Stability of the system performance according to the number of handwriting samples

This evaluation explores the stability of system performance with respect to the amount of handwritten training data on the IAM (text line setup) and CVL (setup-1) databases. Different training/testing configurations are examined, and the result of the writer identification is recorded for each configuration. In the IAM database, the writer identification process is first performed with at least 30% of handwritten data per writer for training and 70% for testing. Then, the amount of training data is gradually increased (by 10%) until reaching 70%. With the CVL database, the setup configuration 30%/70% up to 70%/30% cannot be set since four handwritten samples are available per writer. In other words, CVL only allows the specification of three possible partitions, including 75%/25% (i.e., 75% in the training data versus 25% in the test data), 25%/75%, and 50%/50% (half-

half partition) at an average rate recorded under a 6-fold cross-validation scheme (4-fold cross-validation in the cases of 75%/25% and 25%/75% partitions). By this, we evaluate and investigate all likely scenarios for performing writer identification. The goal here is to validate the effectiveness and stability of the overall system in characterizing writers under different conditions.

The classification results for IAM and CVL databases are given in Table 5.17 and Figure 5.27, respectively. From the plotted results, it can be seen that the classification results obtained by each feature method increase progressively as the training data grows. The overall system typically requires an acceptable amount of learning data (suitable training/test partition) to train the classifier to identify the writers in question with high precision. The 60%/40% partition proved to be the convenient IAM database setup, as it allows the highest performance (a result of 94.06%) when using the CLGP feature method (cf. Table 5.17). In the CVL database, the best average accuracy of 99.51% (by CLGP method) is obtained when the database configuration 75%/25% is used. From Table 5.17 and the Figure 5.27, the proposed CLGP method has significant performance stability overall database partitions compared to the evaluated descriptors.

Table 5.17: Classification results on IAM database over different data partitions

Database setup	**Feature methods**					
training-set(%)/test-set(%)	***CLGP***	***BW-LBC***	***LPQ***	***LTP***	***LBP***	***LETRIST***
30/70	91.62	80.67	64.53	62.71	56.01	68.04
40/60	93.45	85.99	70.01	69.86	65.29	72.75
50/50	93.75	87.67	73.51	70.01	64.99	74.12
60/40	**94.06**	90.11	75.49	73.51	68.49	79.14
70/30	**94.06**	89.04	74.43	72.3	64.84	75.49

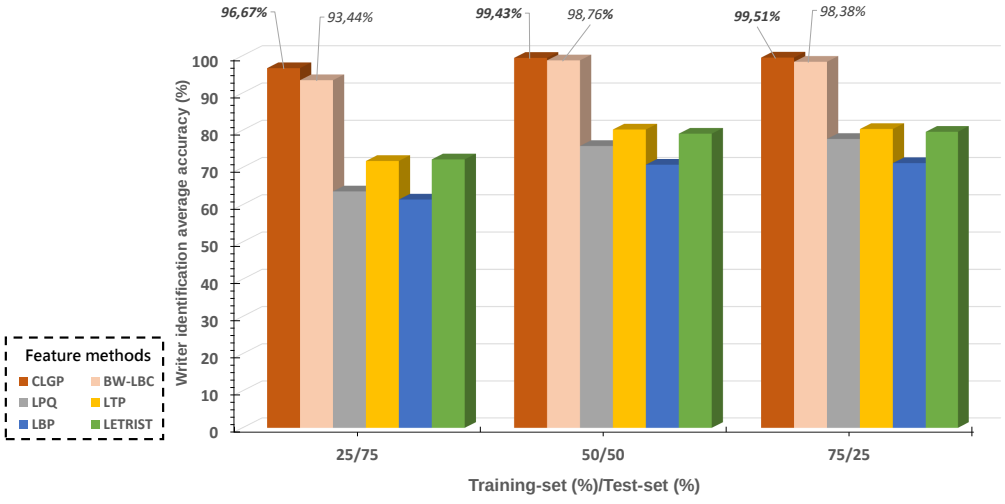


Figure 5.27: Classification results on CVL database over different data partitions.

5.3.4/ PERFORMANCE OF THE LSTP-BASED APPROACH

5.3.4.1/ RESULTS AND ANALYSIS

The LSTP-based approach is evaluated on seven benchmark databases: CVL (setup-1), IFN/ENIT, Firemaker, ICDAR2011, IAM (text line setup), CERUG, and ICDAR2013 databases. The performance of LSTP (Local gradient full-Scale Transform Patterns) is compared with BW-LBC, LBP, LTP, and LPQ feature methods. The classification results for the tested datasets are shown in Tables 5.18, 5.19, and 5.20. From these tables, it can be seen that LSTP is the best performing method. This high performance is due to the flexibility of LSTP and its effectiveness in characterizing local gradient variations of the writing. Moreover, the classification results in Table 5.20 show the ability of LSTP to keep the identification performance constant across all the evaluated splits.

The LSTP is a $(N_c \times 9 \times N_{bk})$-dimensional image feature descriptor. Therefore, tuning the number of blocks N_{bk} and the number of cells N_c can have a direct impact on the final results, i.e, writer identification rates. Optimal parameter values of N_{bk} and N_c are used to obtain LSTP classification results in Tables 5.18, 5.19, and 5.20. These two parameters are set experimentally for all databases tested. The optimal values of N_{bk} and N_c are summarized for each handwritten database in the following: **(1) IAM & CERUG-**

Table 5.18: Performance evaluation on ICDAR2013 (250 writers), CERUG-EN (105 writers), CERUG-CN (105 writers), and CERUG-MIXED (105 writers) databases.

Feature extraction model	Database	Classification rate (%)		
		Top-1	***Top-3***	***Top-5***
LSTP	ICDAR2013	**98%**	100	100
	CERUG-EN	**98.09**	100	100
	CERUG-CN	**100**	100	100
	CERUG-MIXED	**94.28**	98.09	100
BW-LBC	ICDAR2013	96.80	98.4	99.2
	CERUG-EN	91.43	93.33	96.19
	CERUG-CN	92.38	94.28	99.04
	CERUG-MIXED	83.80	85.71	91.42
LPQ	ICDAR2013	82.20	86.4	88
	CERUG-EN	66.66	71.42	79.04
	CERUG-CN	54.28	57.14	64.76
	CERUG-MIXED	59.04	65.71	71.42
LTP	ICDAR2013	72	79.2	85.2
	CERUG-EN	83.81	90.41	96.19
	CERUG-CN	69.52	75.23	81.90
	CERUG-MIXED	64.76	70.47	77.14
LBP	ICDAR2013	70.40	75.2	78
	CERUG-EN	78.09	84.76	92.38
	CERUG-CN	65.71	72.38	78.09
	CERUG-MIXED	61.90	64.76	70.47

Table 5.19: Performance evaluation on IFN/ENIT (411 writers), IAM (657 writers), Firemaker (250 writers), and ICDAR2011 (26 writers) databases.

Feature extraction model	Database	Classification rate (%)		
		Top-1	***Top-3***	***Top-5***
LSTP	IFN/ENIT	**98.28**	100	100
	IAM	**96.80**	98.17	99.54
	Firemaker	**98**	98.80	99.60
	ICDAR2011	**100**	100	100
BW-LBC	IFN/ENIT	97.56	99.27	100
	IAM	90.11	93.15	94.98
	Firemaker	94.40	97.60	98.40
	ICDAR2011	97.43	98.71	100
LPQ	IFN/ENIT	75.42	77.12	78.08
	IAM	75.49	78.68	80.66
	Firemaker	37.20	54.40	64
	ICDAR2011	88.46	98.71	98.71
LTP	IFN/ENIT	83.45	85.63	86.59
	IAM	73.51	76.85	78.68
	Firemaker	30.4	45.6	54
	ICDAR2011	82.05	96.15	100
LBP	IFN/ENIT	71.29	74.43	76.11
	IAM	68.49	72.43	75.01
	Firemaker	33.60	48	55.60
	ICDAR2011	79.48	96.15	98.71

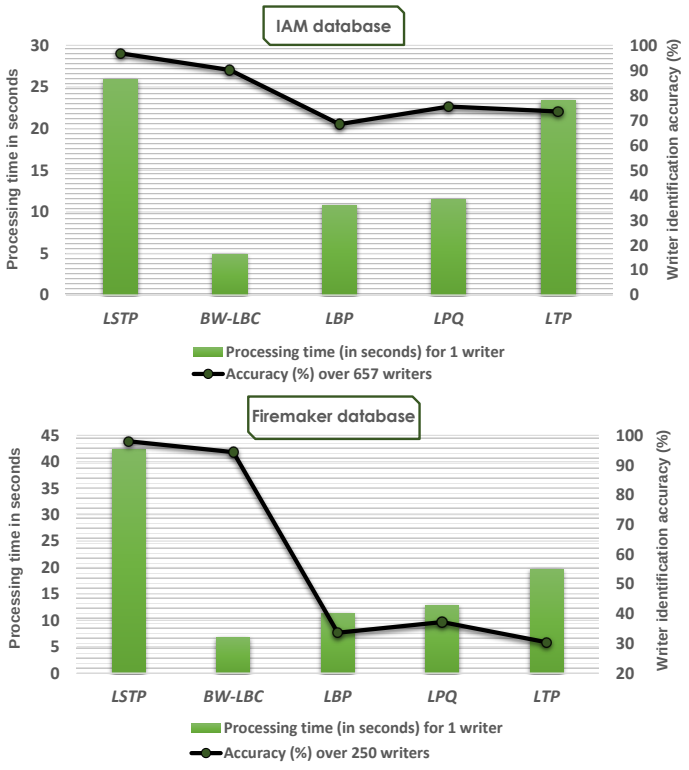
Table 5.20: Performance evaluation on CVL (310 writers) database. The highest classification rates are in bold

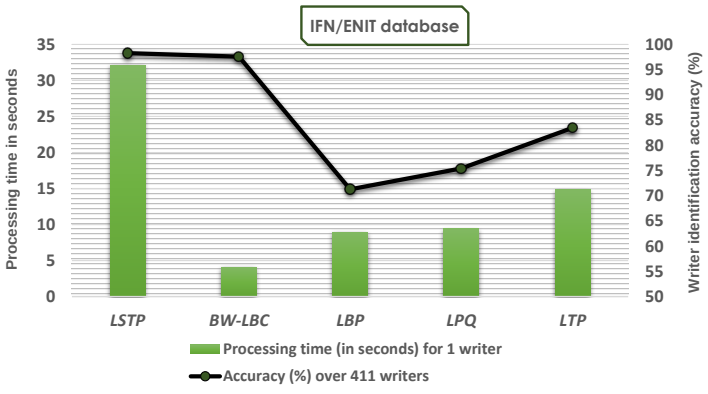
Feature extraction model	Split				Average rate
	Sp.1	*Sp.2*	*Sp.3*	*Sp.4*	
LSTP	99.67	**100**	**100**	99.67	99.83%
BW-LBC	98.7	99.03	97.41	98.38	98.38%
LPQ	83.82	78.64	69.9	78.32	77.67%
LBP	75.4	71.2	65.69	72.49	71.19%
LTP	85.44	79.29	74.11	82.52	80.34%

EN. The number of blocks $N_{bk} = 4$ with $N_c = 16$ represents the best way to handle the writing variations in IAM (score of 96.80%) and CERUG-EN (score of 98.09%). This configuration results in 576 bins in the final LSTP feature histogram (according to $D_{im} = N_c \times 9 \times N_{bk}$). **(2) Firemaker, CERUG-MIXED, IFN/ENIT & CERUG-CN**. The best results of 98% (Firemaker), 94.28% (CERUG-MIXED), 98.28% (IFN/ENIT) and 100% (CERUG-CN) are obtained when LSTP is used with the number of blocks $N_{bk} = 9$ and the number of cells $N_c = 16$ ($D_{im} = 1296$). **(3) ICDAR2013**. We experimentally found that the best top-1 identification rate of 98% is achieved with $N_{bk} = 16$ and $N_c = 9$, resulting in 1296 bins of LSTP feature histogram dimension D_{im}. **(4) ICDAR2011 & CVL**. The number of

blocks $N_{bk} = 9$ with $N_c = 9$ are set as the optimal settings lead to highest performance with a score of 100% on the ICDAR2011 and CVL databases ($D_{im} = 729$).

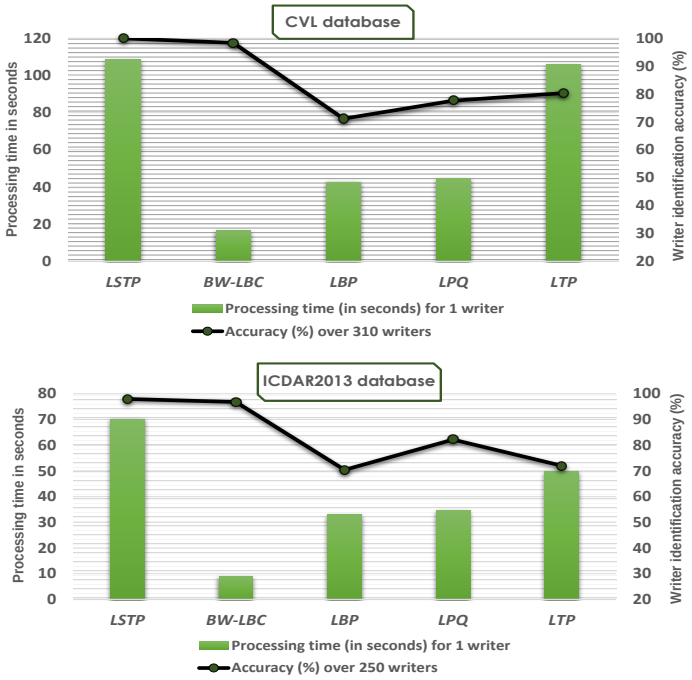
Figures 5.28, 5.29 and 5.30 illustrate the processing time and identification accuracy recorded by the proposed system when using the feature methods LSTP, BW-LBC, LPQ, LBP, and LTP. The processing time increases when the feature dimension is high. The overall system needs more time to classify the writing samples. This happened when comparing and matching feature histograms of samples using Hamming distance. Our previous BW-LBC descriptor (Chahi et al. (2018)) is the computationally fastest method across all databases tested (IAM: 4.9s, IFN/ENIT: 4.1s, ICDAR2011: 6.7s, CVL: 7.8s, Firemaker: 6.8s, ICDAR2013: 12.6s, CERUG-CN: 4.8s, CERUG-EN: 5.19s, CERUG-MIXED: 4.96s). This system behavior is due to its reduced feature histogram size (49 bins). Nevertheless, across all tested databases, the LSTP method outperforms all evaluated feature methods in writer identification accuracy. Since the writer identification task does not require a real-time response (offline mode), the processing time is not always considered as a key performance indicator. Moreover, none of the old and current state-of-the-art systems have recorded the computation time of their frameworks.

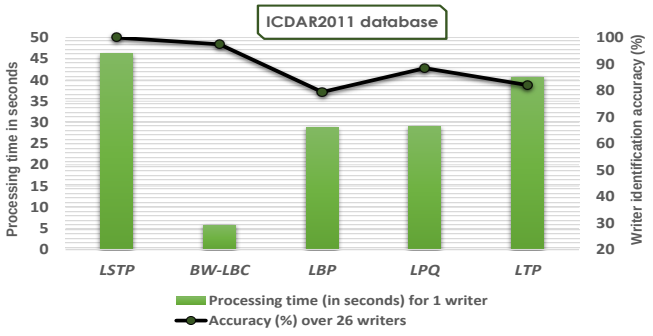




Feature method	Dimension
LSTP	IAM: 576
	IFN: 1296
	Firemaker: 1296
BW-LBC	49
LBP	256
LPQ	256
LTP	512

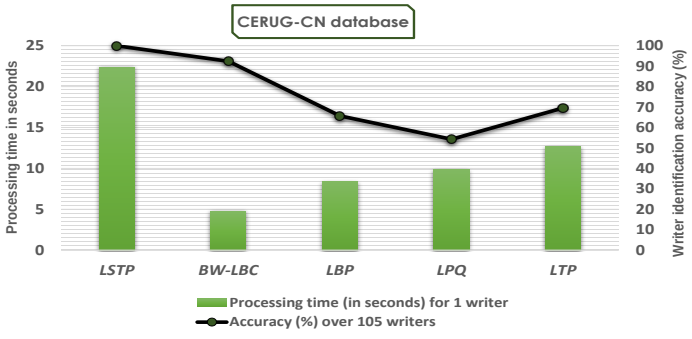
Figure 5.28: The processing time of the proposed system to identify one writer from IAM, IFN/ENIT, and Firemaker databases.

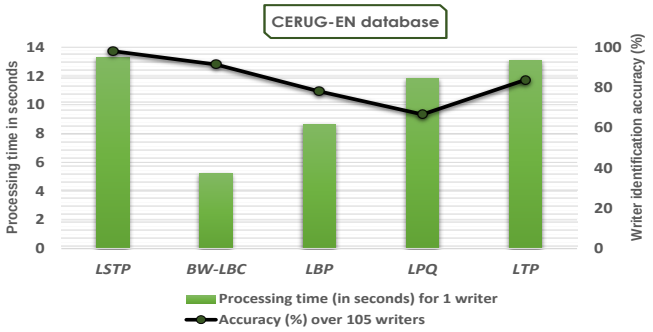


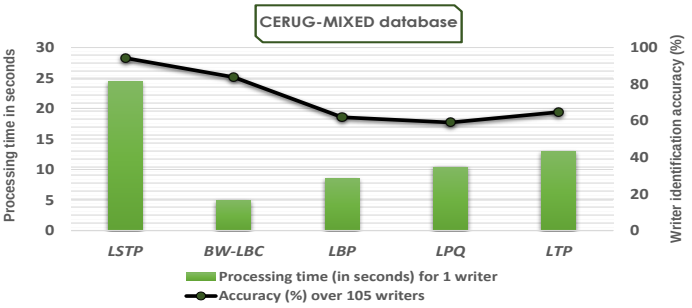


Feature method	Dimension
LSTP	CVL: 729
	ICDAR2011: 729
	ICDAR2013: 1296
BW-LBC	49
LBP	256
LPQ	256
LTP	512

Figure 5.29: The processing time of the proposed system to identify one writer from CVL, ICDAR2011, and ICDAR2013 databases.







Feature method	Dimension
LSTP	CERUG-CN: 1296
	CERUG-EN: 576
	CERUG-MIXED: 1296
BW-LBC	49
LBP	256
LPQ	256
LTP	512

Figure 5.30: The processing time of the proposed system to identify one writer from CERUG-CN, CERUG-EN, and CERUG-MIXED databases.

5.3.4.2/ CLASSIFICATION PERFORMANCE CONFORMING TO THE NUMBER OF WRITERS

This experiment investigates the sensitivity of the system when new writers are added to the evaluations. The number of writers $N_{writers}$ is tuned from 10 to the total number of writers. Figure 5.31 shows the system performance (writer identification rate) as a function of the number of writers $N_{writers}$ in the IAM database. For LBP, LPQ, and LTP descriptors, an expected peak drop in classification rate with gradually increasing $N_{writers}$ is clearly noticed. However, the best performance is achieved by LSTP and BW-LBC with superiority of LSTP. They are least affected when the number of writers increases, i.e., the system's performance slightly decreases over $N_{writers}$.

As shown in Figure 5.31, the classification rate for LSTP and BW-LBC methods starts at 96% for 25 writers and decreases to 96.80% and 90.11% for 657 writers, respectively. The next ranked feature methods are LPQ, followed by LTP and LBP, all of which fail in correctly identifying writers, especially after more than 200 writers. At $N_{writer} = 10$, a classification result of 90% is reported, which gradually drops to 75.49% (LPQ), 73.51% (LTP), and 68.49% (LBP) for a total of 657 writers. This justifies the difficulty and challenge in carrying out the classification process when comparing the dissimilarity measure of each writer with that of a large number of writers. In other words, the classification pro-

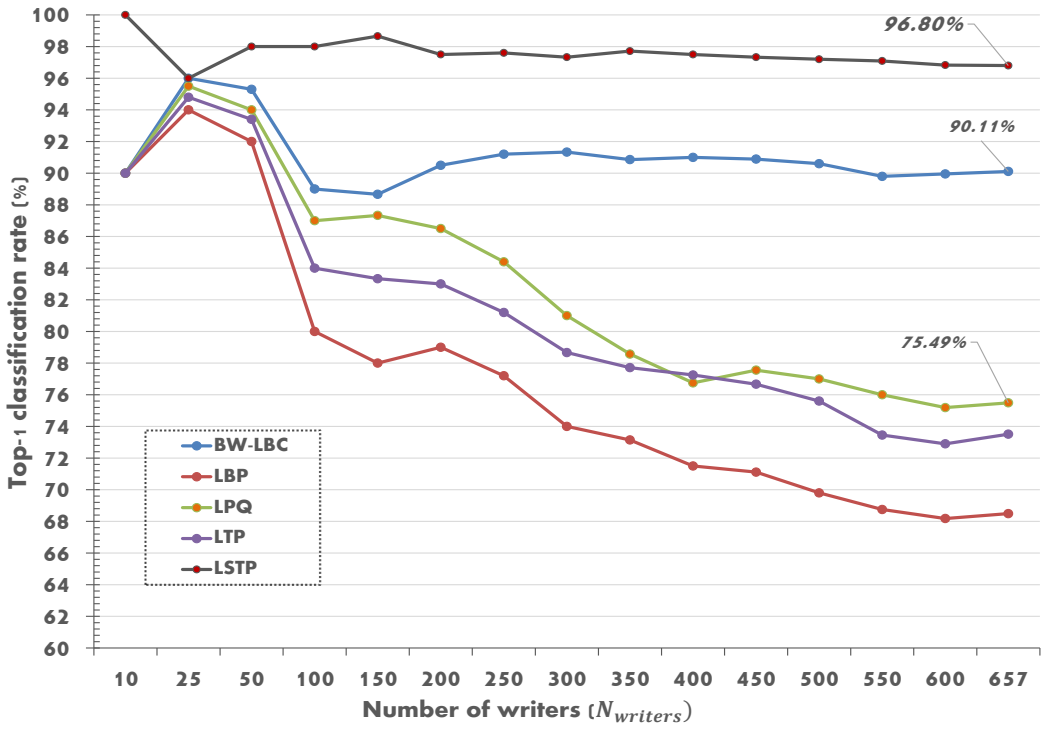


Figure 5.31: Classification performance according to the number of writers in the IAM database.

cess becomes progressively complex as the number of classes (writers) increases. Note that the same behavior is recorded experimentally for the other handwritten databases tested. As expected, the classification results recorded in Figure 5.31 show the validity and performance stability of the proposed system. These results confirm the findings in Section 5.3.4.1, i.e., LSTP is experimentally the best performing method compared to the evaluated feature methods.

5.4/ EXPERIMENTAL RESULTS: DEEP LEARNING-BASED SYSTEMS

As presented in Chapter 4, we proposed two approaches based on deep learning: -$WriterINet$- and $DeepWINet$-based writer identification systems. The writer identification result, expressed in %, is defined as the number of correctly classified samples by the total number of test instances over all writers.

The CNN models are pre-trained using the training set of the IAM database for all tested databases. The evaluations are performed on two Alienware Aurora R8 with Core i9-9900K-9th processor 5GHz Boost, 16 threads, two GPU NVIDIA-RTX2080-8GB, and 64 GB RAM. The CNN network training options are set as follows: (1) **training solver**: CNN models are trained using the Adam optimizer (Kingma and Ba (2014)); (2) **initial learning rate**: The weights of earlier layers in the CNN networks are frozen by setting the initial learning rate to 0 in these layers. This speeds up the network's training and prevents overfitting of the frozen layers when performing the fine-tuning process (transfer learning). Overall, the initial learning rate for the first 24 layers is set to 0 and changed to a small value of 0.0001 to slow down the learning of the network in the transferred layers that are not yet frozen; (3) **training cycle**: The entire training process took 8 to 10 epochs to update the weights of our CNN models. 60 is the mini-batch size for each training iteration; (4) **execution environment**: four GPUs are used in the local parallel pool.

5.4.1/ PERFORMANCE OF THE $WriterINet$-BASED APPROACH

As explained in Chapter 4 (cf. Section 4.2), the $WriterINet$ model is used to extract deep features characterizing all segmented connected components and words from each writing sample. These features are fed into a simple end-to-end artificial neural network (ANN) to perform the identification (classification) process. The ANN network consists of a 1D input, two fully connected layers ($FC1$ and $FC2$), a softmax layer, and a classification layer (cf. Figure 4.5). The $WriterINet$-based system is evaluated on the IAM (document and text-line setups), IFN/ENIT, CVL (setup-1), Firemaker, ICDAR2013, ICDAR2011, and

CERUG databases.

To investigate the performance stability and effectiveness of the proposed $WriterINet$, we study how the number of neurons of the fully connected layer $FC1$ of the ANN network affects the overall performance. The number of neurons is set to 512, 1024, 2048, and 4096 as indicated in Table 5.21, and the identification accuracy for each setting is recorded across all benchmarks evaluated. From the results in Table 5.21, it can be clearly seen that the highest performance is achieved for IFN/ENIT, ICDAR2013, CERUG-CN, CERUG-EN, and CERUG-MIXED databases when 512 is set as the number of neurons of the fully connected layer $FC1$. The number of neurons of 1024 experimentally proves to be the best configuration on CVL (setup-1), ICDAR2013, and CERUG-EN databases with the highest score of 100%. The best configuration of $(FC1 - 2048)$ gives a performance of 99.54% and 98.4% on the IAM (text-line setup) and Firemaker databases, respectively. On the ICDAR2011 database, all tested configurations of $FC1$ allow reaching 100%.

5.4.2/ PERFORMANCE OF THE $DeepWINet$-BASED APPROACH

The proposed $DeepWINet$ CNN model is employed in two scenarios as explained in Chapter 4 (cf. Section 4.3). Scenario 1: $DeepWINet$ CNN model is used as a feature learning method, where deep features are classified using Chi-Square -nearest neighbour classifier. Scenario 2: $DeepWINet$ model is implemented as an end-to-end CNN network. The overall system is evaluated on the IAM (document and text line setups), IFN/ENIT, CVL

Table 5.21: The impact of the number of neurons of the fully connected $FC1$ on system performance across all benchmarks tested.

	IFN/ENIT		**CVL**		**ICDAR2013**	
	Top-1	**Top-2**	**Top-1**	**Top-2**	**Top-1**	**Top-2**
FC1-512	**99.75**	100	99.67	100	**99**	99.6
FC1-1024	99.27	99.75	**100**	100	**99**	99.6
FC1-2048	99.02	99.51	99.03	100	98.6	99
FC1-4096	98.54	99.27	97.09	97.74	98	98.6
	IAM		**Firemaker**		**ICDAR2011**	
	Top-1	**Top-2**	**Top-1**	**Top-2**	**Top-1**	**Top-2**
FC1-512	99.39	99.69	98	99.2	**100**	100
FC1-1024	99.39	99.69	**98.4**	99.6	**100**	100
FC1-2048	**99.54**	99.54	**98.4**	99.6	**100**	100
FC1-4096	99.23	99.54	97.6	98.4	**100**	100
	CERUG-CN		**CERUG-EN**		**CERUG-MIXED**	
	Top-1	**Top-2**	**Top-1**	**Top-2**	**Top-1**	**Top-2**
FC1-512	**95.24**	99.04	**100**	**100**	**100**	**100**
FC1-1024	93.33	97.14	**100**	**100**	99.04	100
FC1-2048	92.38	94.28	98.09	100	97.14	99.04
FC1-4096	90.47	92.38	95.24	98.09	96.19	98.09

(setup-2), Firemaker, ICDAR2013 and CERUG databases.

To investigate the ability of the $DeepWINet$ model in characterizing the writer's style when using scenario 1, we compared its performance, i.e., top-1 and top-5 identification accuracies, with those of traditional hand-crafted descriptors (cf. Tables 5.22 and 5.23). These include BW-LBC (Chahi et al. (2018)), LBP (Ojala et al. (2002)), LTP (Tan and Triggs (2010)), CLGP (Chahi et al. (2020a)), and LPQ (Ojansivu and Heikkilä (2008)) handcrafted descriptors, which are the best known feature methods for the task of writer identification (Chahi et al. (2018); Hannad et al. (2016); Khan et al. (2016); Bertolini et al. (2013); Chahi et al. (2019)). Note that these feature methods are applied using the same classification process as the $DeepWINet$ model.

From the Tables 5.22 and 5.23, it can be seen that the $DeepWINet$ model with its two versions (full and light) far outperforms all the evaluated feature methods. The performance of the hand-crafted descriptors, especially the LBP, LPQ, and LTP methods, is somewhat low as they require a large amount of texture information to perform proper feature representation. However, the features captured by our $DeepWINet$ model are deeply learned in a convolutional pixel-wise mode, which provides much better performance.

Table 5.22: Writer identification results on IAM (document setup), CVL, Firemaker, and CERUG-EN databases. $DeepWINet$ model is used as CNN deep feature extraction method

Feature	**IAM**		**CVL**		**Firemaker**		**CERUG-EN**	
method	***Top-1***	***Top-5***	***Top-1***	***Top-5***	***Top-1***	***Top-5***	***Top-1***	***Top-5***
LBP	66.66	70.01	65.80	70.32	33.60	55.60	78.09	92.38
LPQ	69.86	72.90	67.74	70.96	37.20	64	66.66	79.04
LTP	72.29	75.34	69.83	71.61	30.4	54	83.81	96.19
BW-LBC	90.11	94.9	96.77	98.22	94.40	97.60	91.43	96.19
CLGP	92.99	96.95	98.38	99.35	97.60	99.2	97.14	**100**
DeepWINet (full)	**98.32**	**98.93**	**100**	**100**	**98.4**	**99.60**	**100**	**100**
DeepWINet (light)	**98.02**	**98.78**	**100**	**100**	**97.6**	**99.2**	**100**	**100**

Table 5.23: Writer identification results on IFN/ENIT, CERUG-CN, CERUG-MIXED, and ICDAR2013 databases. $DeepWINet$ model is used as a CNN deep feature extraction method

Feature	**IFN/ENIT**		**CERUG-CN**		**CERUG-MIXED**		**ICDAR2013**	
method	***Top-1***	***Top-5***	***Top-1***	***Top-5***	***Top-1***	***Top-5***	***Top-1***	***Top-5***
LBP	71.29	76.11	65.71	78.09	61.90	70.47	70.40	78
LPQ	75.42	78.08	54.28	64.76	59.04	71.42	82.20	88
LTP	30.4	54	69.52	81.90	64.76	77.14	72	85.2
BW-LBC	97.56	**100**	92.38	99.04	83.80	91.42	96.80	99.2
CLGP	98.54	**100**	90.47	98.09	93.33	**100**	97	99.6
DeepWINet (full)	**99.27**	**100**	**94.28**	**100**	**100**	**100**	**99.8**	**100**
DeepWINet (light)	**99.02**	**100**	**93.33**	**100**	**100**	**100**	**99.2**	**100**

As the evaluation of scenario 2, the performance of the $DeepWINet$ network (full and light configurations) is compared with VGG-19 (Simonyan and Zisserman (2014)), WorldImgNet (He and Schomaker (2020)), FragNet-64 (He and Schomaker (2020)) and AlexNet (Krizhevsky et al. (2012)) CNN networks. The identification results are reported in Tables 5.24 and 5.25, which indicate that $DeepWINet$ CNN model provides the best top 1 identification results over all tested databases.

For the Firemaker database, the top-1 performance of the $DeepWINet$ model is competitive with that of WorldImgNet and FragNet-64 networks. This high performance demonstrates the effectiveness of the proposed $DeepWINet$ for writer identification based on connected component sub-images. The performance of our system using scenario 1 ($DeepWINet$ as deep feature representation; Tables 5.22 and 5.23) is better than using scenario 2 ($DeepWINet$ as an end-to-end CNN network; Tables 5.24 and 5.25), as shown in Table 5.26. The classification process in scenario 1 uses a double comparison mechanism to classify the deep feature vectors. Both distance and dissimilarity metrics are jointly used to compare and match the common details contained in the feature vectors in an efficient manner, which further improves the results.

For both scenarios 1 and 2, the full configuration of $DeepWINet$ leads to better performance compared to the light configuration.

Table 5.24: Writer identification results on CERUG, CVL, and IFN/ENIT databases. $DeepWINet$, VGG-19, and AlexNet are implemented and used as an end-to-end CNN networks

CNN model	CERUG-CN		CERUG-EN		CERUG-MIXED		CVL		IFN/ENIT	
	Top-1	***Top-5***	***Top-1***	***Top-5***	***Top-1***	***Top-5***	***Top-1***	***Top-5***	***Top-1***	***Top-5***
VGG-19	88.57	95.24	92.38	96.19	90.47	99.05	96.77	98.38	79.07	81.50
WordImgNet	—	—	97.1	**100**	—	—	98.8	99.4	—	—
FragNet-64	—	—	98.1	**100**	—	—	99.1	99.4	—	—
AlexNet	92.38	97.14	98.1	**100**	97.14	**100**	98.71	99.03	88.80	96.59
DeepWINet (full)	**94.28**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**98.78**	**99.75**
DeepWINet (light)	**92.38**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**98.78**	**99.51**

Table 5.25: Writer identification results on IAM (document setup), Firemaker, and ICDAR2013 databases. $DeepWINet$, VGG-19, and AlexNet are implemented and used as an end-to-end CNN networks

CNN model	IAM		Firemaker		ICDAR2013	
	Top-1	***Top-5***	***Top-1***	***Top-5***	***Top-1***	***Top-5***
VGG-19	94.36	96.65	88	92	96	97.8
WordImgNet	95.8	98	**97.6**	98.8	—	—
FragNet-64	96.3	98	**97.6**	**99.6**	—	—
AlexNet	96.49	97.26	92	95.2	98.6	99
DeepWINet (full)	**97.41**	**98.93**	**97.6**	99.2	**99**	**100**
DeepWINet (light)	**96.95**	**98.78**	**97.2**	99.2	**99**	**99.8**

Table 5.26: Writer identification performance. (S1): $DeepWINet$ is used as a CNN deep feature extraction method. (S2): $DeepWINet$ is used as an end-to-end CNN network.

Database	DeepWINet (Full)				DeepWINet (Light)			
	S1		S2		S1		S2	
	TOP-1	*TOP-5*	*TOP-1*	*TOP-5*	*TOP-1*	*TOP-5*	*TOP-1*	*TOP-5*
IAM	**98.32**	**98.93**	97.41	**98.93**	**98.02**	**98.78**	96.95	**98.78**
IFN/ENIT	**99.27**	**100**	98.78	99.75	**99.02**	**100**	98.78	99.51
CVL	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
CERUG-CN	**94.28**	**100**	**94.28**	**100**	**93.33**	**100**	92.38	**100**
CERUG-EN	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
CERUG-MIXED	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
ICDAR2013	**99.8**	**100**	99	**100**	**99.2**	**100**	99	99.8
Firemaker	**98.4**	**99.60**	97.6	99.2	**97.6**	**99.2**	97.2	**99.2**

5.5/ PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART

In recent years, significant progress has been made in the field of offline text-independent writer identification. Extensive effort in this area of research has focused primarily on developing new and efficient yet robust frameworks that would produce a higher performance for writer identification. However, most of these researches differ in terms of the database setting tested. Even with a common database, direct one-to-one comparisons with state-of-the-art systems are still not straightforward, mainly related to the following factors: (i) imprecision about which writers are selected for the identification process because the authors used only a subset of the evaluated handwritten database, especially in the case of the IAM and IFN/ENIT databases, and (ii) different numbers of handwritten samples (for training and test sets) per writer is used. In order to provide a fair and meaningful performance comparison of our system with the state-of-the-art systems, we only consider well-known systems evaluated on the entire set of writers for the ten tested databases in our comparative evaluation.

Tables 5.27, 5.28 and 5.29 summarize the top-1 classification results obtained by our approaches together with those of the current state-of-the-art systems on IFN/ENIT, CVL, ICDAR2013, IAM, Firemaker, ICDAR2011, AHTID/MW, CERUG-CN, CERUG-EN, and CERUG-MIXED databases.

- **Results on IFN/ENIT.** # *BWLBC-based approach*: The proposed system allows achieving the highest average accuracy of 96.47% over 10 subdivisions in the IFN/ENIT database, which outperforms the nearest performing system presented in (Hannad et al. (2016)) by 1.58%, while their result was recorded only over one subdivision in training and testing sets. # *Handcrafted-based approach*: This approach provides a very accurate and efficient solution for identifying the writers from the Arabic IFN/ENIT database. Using the LPQ descriptor in feature extraction, our

Table 5.27: Performance comparison on IFN/ENIT, CVL, and ICDAR2013 databases. S1: Scenario 1, *DeepWINet* is applied as deep CNN feature method; S2: Scenario 2, *DeepWINet* is used as an end-to-end CNN network;

Approach	#Year	#Script	Benchmark	#Classes	#Feature + Classifier	Writer identification Top-1 accuracy
Bulacu et al.	2007	Arabic	IFN/ENIT	350	Contour & Grapheme + Nearest Neighbor	88%
Abdi and Khemakhem	2015	Arabic	IFN/ENIT	411	Synthetic codebooks + Chi-square	90.02%
Hannad et al.	2016	Arabic	IFN/ENIT	411	LPQ + Hamming	94.89%
Khan et al.	2017	Arabic	IFN/ENIT	411	BDCT + Nearest-center	76%
Proposed BWLBC-based approach (Chahi et al.)	**2018**	**Arabic**	**IFN/ENIT**	**411**	**BWLBC + NN-Hamming**	**96.47%**
Hadjadji and Chibani	2018	Arabic	IFN/ENIT	411	LPQ, RL, and oBIF + OC-K-Means	97.56%
Proposed LPQ-based approach (Chahi et al.)	**2019**	**Arabic**	**IFN/ENIT**	**411**	**LPQ + NN-Hamming**	**97.81%**
Proposed LTP-based approach (Chahi et al.)	**2019**	**Arabic**	**IFN/ENIT**	**411**	**LTP + NN-Hamming**	**96.84%**
Proposed LBP-based approach (Chahi et al.)	**2019**	**Arabic**	**IFN/ENIT**	**411**	**LBP + NN-Hamming**	**95.13%**
Khan et al.	2019	Arabic	IFN/ENIT	411	SIFT and RootSIFT with GMM	97.28%
Kumar and Sharma	2019	Arabic	IFN/ENIT	411	DCWI + SVM and SBC	97.50%
Proposed LSTP-based approach (Chahi et al.)	**2020**	**Arabic**	**IFN/ENIT**	**411**	**LSTP + NN-Hamming**	**98.28%**
Proposed CLGP-based approach (Chahi et al.)	**2020**	**Arabic**	**IFN/ENIT**	**411**	**CLGP + NN-Hamming**	**98.54%**
Kumar and Sharma	2020	Arabic	IFN/ENIT	411	CNN	98.24%
Proposed *WriterINet*-based approach		**Arabic**	**IFN/ENIT**	**411**	**CNN + ANN**	**99.75%**
Proposed *DeepWINet*-based approach (S2)		**Arabic**	**IFN/ENIT**	**411**	**DeepWINet (full & light)**	**98.78%**
Proposed *DeepWINet*-based approach (S1)		**Arabic**	**IFN/ENIT**	**411**	**DeepWINet (full) + NN-Chi-square**	**99.27%**
Proposed *DeepWINet*-based approach (S1)		**Arabic**	**IFN/ENIT**	**411**	**DeepWINet (light) + NN-Chi-square**	**99.02%**
Fiel and Sablatnig	2013	English & German	CVL	309	SIFT + Cosine	97.8%
Fiel and Sablatnig	2015	English & German	CVL	309	CNN + Nearest Neighbor	98.9%
Kanetkar et al.	2016	English & German	CVL	308	LDP + Chi-square	98.1%
Khan et al.	2017	English & German	CVL	310	BDCT + Nearest-center	99.6%
Proposed BWLBC-based approach (Chahi et al.)	**2018**	**English & German**	**CVL**	**310**	**BWLBC + NN-Hamming**	**99.03%**
Kessentini et al.	2018	English & German	CVL	310	SVM with DST	94.83%
Proposed handcrafted-based approach (Chahi et al.)	**2019**	**English & German**	**CVL**	**310**	**LBP & LPQ & LTP + NN-Hamming**	**99.35%**
Khan et al.	2019	English & German	CVL	310	SIFT and RootSIFT with GMM	99.03%
Chen et al.	2019	English & German	CVL	310	CNN-WLSR	99.2%
Proposed CLGP-based approach (Chahi et al.)	**2020**	**English & German**	**CVL**	**310**	**CLGP + NN-Hamming**	**99.51%**
Proposed LSTP-based approach (Chahi et al.)	**2020**	**English & German**	**CVL**	**310**	**LSTP + NN-Hamming**	**99.83%**
Javidi and Jampour	2020	English & German	CVL	310	CNN + HTD	96.16%
He and Schomaker	2020	English & German	CVL	310	CNN	99.1%
Kumar and Sharma	2020	English & German	CVL	310	CNN	99.35%
Proposed *WriterINet*-based approach	**-**	**English & German**	**CVL**	**310**	**CNN + ANN**	**100%**
Proposed *DeepWINet*-based approach (S2)		**English & German**	**CVL**	**310**	**DeepWINet (full & light)**	**100%**
Proposed *DeepWINet*-based approach (S1)		**English & German**	**CVL**	**310**	**DeepWINet (full & light) + NN-Chi-square**	**100%**
CVL-IPK method. (Louloudis et al.)	2013	English & Greek	ICDAR2013	250	Fisher Vector + Cosine	90.9%
TEBESSA-c method. (Louloudis et al.)	2013	English & Greek	ICDAR2013	250	Hinge with Run-length + Manhattan	93.4%
HIT-ICG method. (Louloudis et al.)	2013	English & Greek	ICDAR2013	250	SIFT + Chi-square	94.8%
CS-UMD-b method. (Louloudis et al.)	2013	English & Greek	ICDAR2013	250	Contour gradient + K-means	95%
Christlein et al.	2014	English & Greek	ICDAR2013	250	RootSIFT with GMM	97.1%
Fiel and Sablatnig	2015	English & Greek	ICDAR2013	250	CNN + Nearest Neighbor	88.5%
Chen et al.	2019	English & Greek	ICDAR2013	250	CNN-WLSR	97.7%
Proposed LSTP-based approach (Chahi et al.)	**2020**	**English & Greek**	**ICDAR2013**	**250**	**LSTP + NN-Hamming**	**98.4%**
Proposed *WriterINet*-based approach		**English & Greek**	**ICDAR2013**	**250**	**CNN + ANN**	**99%**
Proposed *DeepWINet*-based approach (S2)		**English & Greek**	**ICDAR2013**	**250**	**DeepWINet (full & light)**	**99%**
Proposed *DeepWINet*-based approach (S1)		**English & Greek**	**ICDAR2013**	**250**	**DeepWINet (full) + NN-Chi-square**	**99.8%**
Proposed *DeepWINet*-based approach (S1)		**English & Greek**	**ICDAR2013**	**250**	**DeepWINet (light) + NN-Chi-square**	**99.2%**

system reaches the highest top-1 identification rate of 97.81%, which is an improvement over our previous BWLBC-based approach and the system presented in (Hadjadji and Chibani (2018)) by 1.34% and 0.25%, respectively. *# (CLGP & LSTP)-based approaches*: From the Table 5.27, it can be clearly seen that the proposed system with CLGP and LSTP gives the best performance with top 1 scores of 98.28% (LSTP) and 98.54% (CLGP) among all the compared studied systems. *# (WriterINet & DeepWINet)-based approaches*: The *WriterINet*- and full *DeepWINet*-based approaches outperform all investigated SOTA systems in the top-1 identification accuracy (high scores of 99.75% (*WriterINet*), 99.27% (full *DeepWINet* with scenarios 1), and 98.78% (full *DeepWINet* with scenarios 2)). Even with the light version of *DeepWINet*, the overall approach still performs the best in this database.

Table 5.28: Performance comparison on IAM, Firemaker, ICDAR2011, and AHTID/MW databases. S1: Scenario 1, DeepWINet is applied as deep CNN feature method; S2: Scenario 2, DeepWINet is used as an end-to-end CNN network;

Approach	#Year	#Script	#Benchmark	#Classes	#Feature + Classifier	Writer identification Top-1 accuracy
Khalifa et al.	2015	English	IAM	650	Multiple codebooks + Nearest Neighbor	92%
Hannad et al.	2016	English	IAM (text-line setup)	657	LPQ + Hamming	89.54%
Proposed BWLBC-based approach (Chahi et al.)	2018	**English**	**IAM (text-line setup)**	**657**	**BWLBC + NN-Hamming**	**90.11%**
Hadjadji and Chibani	2018	English	IAM (text-line setup)	657	LPQ, RL, and oBIF + OC-K-Means	94.51%
Proposed LPQ-based approach (Chahi et al.)	2019	**English**	**IAM (text-line setup)**	**657**	**LPQ + NN-Hamming**	91.17%
Kumar and Sharma	2019	English	IAM (text-line setup)	657	DCWI + SVM and SBC	97.80%
Durou et al.	2019	English	IAM	650	OBI and Grapheme + K-nearest neighbour	92%
Proposed CLGP-based approach (Chahi et al.)	2020	**English**	**IAM (text-line setup)**	**657**	**CLGP + NN-Hamming**	94.06%
Proposed LSTP-based approach (Chahi et al.)	2020	**English**	**IAM (text-line setup)**	**657**	**LSTP + NN-Hamming**	96.80%
Bulacu and Schomaker	2007	English	IAM (document setup)	650	Contour & Grapheme + Nearest Neighbor	89%
Siddiqi and Vincent	2010	English	IAM (document setup)	650	Codebook & Contour + Chi-square	91%
Kumar et al.	2014	English	IAM (document setup)	650	Fourier & wavelet + Nearest Neighbor	88.43%
Wu et al.	2014	English	IAM (document setup)	657	SDS + SOH	98.5%
He et al.	2015	English	IAM (document setup)	650	Junclets	91.10%
He and Schomaker	2017	English	IAM (document setup)	650	COLD-LBPruns + Nearest Neighbor	89.90%
Khan et al.	2017	English	IAM (document setup)	650	BDCT + Nearest-center	97.2%
Nguyen et al.	2019	English	IAM (document setup)	650	CNN	90.12%
Khan et al.	2019	English	IAM (document setup)	650	SIFT and RootSIFT with GMM	97.85%
He and Schomaker	2020	English	IAM	657	CNN	96.3%
Kumar and Sharma	2020	English	IAM (document setup)	657	CNN	97.27%
Javidi and Jampour	2020	English	IAM	657	CNN + HTD	97.50%
Proposed $WriterINet$-based approach		**English**	**IAM (text-line setup)**	**657**	**CNN + ANN**	**99.54%**
Proposed $WriterINet$-based approach		**English**	**IAM (document setup)**	**657**	**CNN + ANN**	**98.17%**
Proposed $DeepWINet$-based approach (S2)		**English**	**IAM (text-line setup)**	**657**	**DeepWINet (full)**	**98.93%**
Proposed $DeepWINet$-based approach (S1)		**English**	**IAM (text-line setup)**	**657**	**DeepWINet (full) + NN-Chi-square**	**99.54%**
Proposed $DeepWINet$-based approach (S1)		**English**	**IAM (document setup)**	**657**	**DeepWINet (full) + NN-Chi-square**	**98.32%**
Proposed $DeepWINet$-based approach (S1)		**English**	**IAM (document setup)**	**657**	**DeepWINet (light) + NN-Chi-square**	**98.02%**
Ghiasi and Safabakhsh	2013	Dutch	Firemaker	250	Contour codebook fragments + Nearest Neighbor	91.80%
Wu et al.	2014	Dutch	Firemaker	250	SDS + SOH	92.40%
He et al.	2015	Dutch	Firemaker	250	Junclets	89.80%
He and Schomaker	2017	Dutch	Firemaker	250	COLD-LBPruns + Nearest Neighbor	86.2%
Nguyen et al.	2019	Dutch	Firemaker	250	CNN	92.38
Khan et al.	2019	Dutch	Firemaker	250	SIFT and RootSIFT with GMM	97.98%
Proposed CLGP-based approach (Chahi et al.)	**2020**	**Dutch**	**Firemaker**	**250**	**CLGP + NN-Hamming**	97.60%
Proposed LSTP-based approach (Chahi et al.)	**2020**	**Dutch**	**Firemaker**	**250**	**LSTP + NN-Hamming**	98%
He and Schomaker	2020	Dutch	Firemaker	250	CNN	97.6%
Javidi and Jampour	2020	Dutch	Firemaker	250	CNN + HTD	99.6%
Proposed $WriterINet$-based approach		**Dutch**	**Firemaker**	**250**	**CNN + ANN**	**98.4%**
Proposed $DeepWINet$-based approach (S1)		**Dutch**	**Firemaker**	**250**	**DeepWINet (full)+ NN-Chi-square**	**98.4%**
TEBESSA method (Louloudis et al.)	2011	Hybrid	ICDAR2011	26	GLRL + Manhattan	87.50%
TSINGHUA method (Louloudis et al.)	2011	Hybrid	ICDAR2011	26	GMSF + Variance Weighted Chi-square	90.90%
Wu et al.	2014	Hybrid	ICDAR2011	26	SDS + SOH	95.20%
Fiel and Sablatnig	2015	Hybrid	ICDAR2011	26	CNN + Nearest Neighbor	94.7%
Mohammed et al.	2017	Hybrid	ICDAR2011	26	FAST keypoints + Local NBNN	98.6%
Khan et al.	2019	Hybrid	ICDAR2011	26	SIFT and RootSIFT with GMM	100%
Proposed CLGP-based approach (Chahi et al.)	**2020**	**Hybrid**	**ICDAR2011**	**26**	**CLGP + NN-Hamming**	**100%**
Proposed LSTP-based approach (Chahi et al.)	**2020**	**Hybrid**	**ICDAR2011**	**26**	**LSTP + NN-Hamming**	**100%**
Proposed $WriterINet$-based approach		**Hybrid**	**ICDAR2011**	**26**	**CNN + ANN**	**100%**
Slimane and Märgner	2014	Arabic	AHTID/MW	53	Gaussian Mixture Model (GMM)	69.48%
Khan et al.	2016	Arabic	AHTID/MW	53	MSLTPH + majority voting	87.5%
Khan et al.	2017	Arabic	AHTID/MW	53	BDCT + Nearest-center	71.6%
Schomaker and Bulacu (implemented in Khan et al.)	2017	Arabic	AHTID/MW	53	Contour features + Nearest Neighbor	66.4%
Hannad et al (implemented in Khan et al.)	2017	Arabic	AHTID/MW	53	LPQ + Hamming	77.3%
Proposed BWLBC-based approach (Chahi et al.)	**2018**	**Arabic**	**AHTID/MW**	**53**	**BWLBC + NN-Hamming**	**99.53%**
Proposed LPQ-based approach (Chahi et al.)	**2019**	**Arabic**	**AHTID/MW**	**53**	**LPQ + NN-Hamming**	**99.53%**
Khan et al.	2019	Arabic	AHTID/MW	53	SIFT and RootSIFT with GMM	95.60%
Proposed CLGP-based approach (Chahi et al.)	**2020**	**Arabic**	**AHTID/MW**	**53**	**CLGP + NN-Hamming**	**99.53%**

- **Results on CVL**. # *BWLBC-based approach*: The system reaches a classification rate of 99.03% in $split.2$ (see Table 5.3), only outperformed by 0.57%, compared to the best result obtained in (Khan et al. (2017)) over one split in training and test sets. # *Handcrafted-based approach*: Comparing the overall performance on the CVL database, the LPQ-based system reaches an average accuracy of 98.62% and a maximum score of 99.35% in $split.2$ (cf. Table 5.7), which is exceeded by only

Table 5.29: Performance comparison on CERUG-CN, CERUG-EN, and CERUG-MIXED databases (105 writers). S1: Scenario 1, DeepWINet is applied as deep CNN feature method; S2: Scenario 2, DeepWINet is used as an end-to-end CNN network;

Approach	#Year	#Classes	CERUG-CN (%)	CERUG-EN (%)	CERUG-MIXED (%)
Hinge (Bulacu and Schomaker)	2007	105	90.8	12.3	84.7
Quill (Brink et al.)	2012	105	82.7	15.8	74.8
Junclets (He et al.)	2015	105	90.4	87.1	85.7
COLD + LBPruns (He and Schomaker)	2017	105	93.8	97.1	98.5
Proposed LSTP-based approach (Chahi et al.)	**2020**	**105**	**100**	**98.09**	**94.28**
FragNet-64 (He and Schomaker)	2020	105	-	100	-
Proposed $WriterINet$-based approach		**105**	**95.24**	**100**	**100**
Proposed $DeepWINet$-based approach (full) (S2 & S1)		**105**	**94.28**	**100**	**100**
Proposed $DeepWINet$-based approach (light) (S1)		**105**	**93.33**	**100**	**100**

0.25%, compared to the most competitive system in (Khan et al. (2017)), which reaches a score of 99.6% over one split. # *(CLGP & LSTP)-based approaches*: The CLGP-based system reaches the highest average accuracy of 99.51% and a maximum score of 99.67% in $split.2$ and $split.3$ (cf. Table 5.16) compared to most state-of-the-art systems, including deep learning ones. This performance is competitive to the classification score of 99.6% achieved in (Khan et al. (2017)). The LSTP-based system provides a reliable and efficient solution to accurately identify writers with a top-1 score of 99.83%, which outperforms all SOTA systems investigated. # *($WriterINet$ & $DeepWINet$)-based approaches*: Both $WriterINet$ and $DeepWINet$ approaches achieve further improvements over our previous systems. They manage to perfectly differentiate all classes with a score of 100%. This high performance confirms the reliability and validity of our CNN-based frameworks for writer identification.

- **Results on ICDAR2013**. # *LSTP-based approach*: As shown in Table 5.27, the LSTP-based approach performs better in the ICDAR2013 database with a score of 98.4%. It outperforms all other SOTA systems so far. # *$WriterINet$-based approach*: The framework improves the overall performance with a score of 99% outperforming our previous LSTP-based approach. # *$DeepWINet$-based approach*: From Table 5.27, it is clear that the proposed framework with the full and light versions of $DeepWINet$ outperforms all the studied SOTA systems with identification rates of 99.8% (full) and 99.2% (light).

- **Results on IAM**. # *BWLBC-based approach*: As shown in Table 5.28, the system records a promising result on the IAM database (text line setup) with an identification rate of 90.11% ($split.4$ in Table 5.2). Our framework outperforms the hand-crafted system presented in (Hannad et al. (2016)) but is still competitive to (Khalifa et al. (2015)). Note that the system in (Khalifa et al. (2015)) used a different IAM setup in its evaluations. # *(LPQ & CLGP & LSTP)-based approaches*: The proposed approaches improve the performance in the IAM database (text line setup) over the previous BWLBC-based approach. The LSTP-based approach is the top 2 system

with a classification result of 96.80%, which is outperformed by Kumar and Sharma (2019) with a result of 97.80%. For this database, we believe that the performance drop using our texture-based approaches is due to the bad adaptation of the feature extraction methods with the segmented component sub-images. The original text line images contain unwanted random traces. As a result, some representative writing traces are lost during the image segmentation step. This affects the overall system to perform correct writer identification. # *($WriterINet$ & $DeepWINet$)-based approaches*: Our deep learning proposals provide further enhancements to the IAM database with text line and document setups. As shown in Table 5.2, the highest identification accuracy of 99.54% is reached on IAM (text line setup) with $WriterINet$-based and $DeepWINet$ (S1 & full)-based approaches, outperforming all SOTA systems. In the IAM database with document setup, the second-best SOTA system is the $DeepWINet$-based approach (S1 & full) with a classification result of 98.32%, slightly outperformed by 0.18% versus the system presented in (Wu et al. (2014)) (score of 98.5%). However, our deep learning-based approaches outperform the same system in (Wu et al. (2014)) by 6% and 4.8% on Firemaker and ICDAR2011 databases, respectively (cf. Table 5.28).

- **Results on Firemaker**. # *CLGP-based approach*: As depicted in Table 5.28, the top 1 identification accuracy of 97.60% is achieved on the Firemaker database, slightly outperformed by the nearest best system in (Khan et al. (2019)) by about 0.38%. # *LSTP-based approach*: Using the LSTP-based approach, the classification results are improved in this database with a top-1 score of 98%, outperforming the system introduced in (Khan et al. (2019)). # *($WriterINet$ & $DeepWINet$)-based approaches*: The experimental results show that the approach presented in (Javidi and Jampour (2020)) is the best performing system with a classification score of 99.61%. However, ($WriterINet$ & $DeepWINet$)-based approaches are still competitive with the best second score of 98.4%. Our approaches outperform the same work in (Javidi and Jampour (2020)) on CVL and IAM databases, respectively.

- **Results on ICDAR2011 and AHTID/MW**. In the ICDAR2011 database, all our proposals (with the system in Khan et al.) report a top 1 writer identification accuracy of 100%, outperforming all experimental classification results reported in the literature. Our approaches achieve the highest SOTA performance (score of 99.53%) on the AHTID/MW database. They show a significant improvement of 3.93% and 12.03% over the nearest best-performing systems presented in (Khan et al. (2019)) and (Khan et al. (2016)), respectively.

- **Results on CERUG**. # *LSTP-based approach*: As shown in Table 5.29, top-1 identification rates of 100% and 98.09% are reached in the Chinese CERUG-CN and English CERUG-EN databases, respectively, which exceed all experimen-

tal results reported in the literature. In the English-Chinese CERUG-MIXED, a score of 94.28% is recorded, which still rivals the score of 98.5% obtained by He and Schomaker (2017b). # *(WriterINet & DeepWINet)-based approaches*: In the CERUG-CN database, the highest performance (score of 95.24%) is reached with our *WriterINet*-based approach but outperformed by 4.76% over the LSTP-based approach. In CERUG-EN and CERUG-MIXED databases, 100% accuracy in writer identification is achieved using (*WriterINet* & *DeepWINet*)-based approaches.

5.6/ RESULTS OF THE ICFHR2020 COMPETITION

5.6.1/ EVALUATION PROTOCOL AND ERROR METRICS

As indicated in Chapter 4 (cf. Section 4.4), contestants are asked to hand a 20019×20019 distance matrix, since the test set consists of 20019 fragment images. The evaluation is performed using a leave-one-image-out cross-validation strategy. This means that each fragment image of the test set is used as a query, for which the other test fragment images are ranked in a hit list according to their distance similarities (the smaller, the more similar). The metrics are then averaged over all unseen images. Participants' methods are assessed in two ways (two main tasks): (i) writer task, i.e., finding all similar fragment images belonging to the same writer ID based on the writing style, and (ii) page task, i.e., finding all fragment images generated from the same page ID.

The test set includes 20 019 fragments from 1152 writers generated from 2732 historical documents. Seuret et al. (2020) report that the test set is class imbalanced. More precisely, the test set consists of four relevant samples per writer, up to 69 fragment samples that need to be identified and retrieved. Note that the IDs of the writers in the training set are different from those in the test set, and that there is no overlap between the two sets. Details of the origin and arrangement of the database can be found in Chapter 4 (cf. Section 4.4).

Mean average precision (mAP), or sometimes referred to as AP, is a popular metric for evaluating the performance of systems and models for document/information retrieval and object detection tasks (Nicolaou et al. (2018)). The mAP is used as the key performance metric to rank the winners of the two contest tasks (writer-level & page-level). It is calculated from the submitted participants' distance matrices given in a CSV file format. The first row and the first column of the CSV file denote the query fragment and the gallery file, where the corresponding cell entries contain the per- patch distances. The average precision (AP) is computed for each query fragment Q. Thus, the precision is calculated over all ranks i of the retrieved list.

$$AP_Q = \frac{1}{D}\sum_{i=1}^{R} Pr_Q(i) \times IF_Q(i) \tag{5.1}$$

The parameter R is the size of the retrieved list, where D is the number of relevant fragments to the query Q. $Pr_Q(i)$ is the precision at rank i and $IF_Q(i)$ is an indication function that returns 1 if the fragment at rank i is relevant and 0 otherwise. An evaluation platform [1] to measure the mAP was provided to participants to perform validation tests (from the training set). Top-1 accuracy is also used as an evaluation metric in the competition. It is defined as the average accuracy at rank 1. In addition, $Pr@10$ and $Pr@100$ are provided as error metrics to investigate the participants' methods' performance further. Formally, the $Pr@K$ is calculated as follows:

$$Pr@K = \frac{1}{N}\sum_{j=1}^{N} \frac{R_{j,\,K}}{min\left(R_{j,\,N}, K\right)} \tag{5.2}$$

where N is the total number of fragments and $R_{j,\,n}$ is the number of relevant samples for query fragment j up to rank n. The $min(,K)$ is added to indicate whether a method can achieve higher precision for each fragment.

5.6.2/ EXPERIMENTAL RESULTS

As mentioned earlier, the 20 019 test fragments are classified at two different levels: (1) writer level (Task 1) and (2) page level (Task 2). The fragment images have the following naming convention: WID-PID-FID.jpg, where WID=Writer-ID, PID: Page-ID, FID=Fragment-ID. The evaluation is based on five methods submitted by different universities. These include the University of Bourgogne Franche-Comte (UBFC), the University of Groningen (RUG), the University of Tebessa (ULT), and the University of Belfort-Montbeliard Technology (UTBM). The description of the approaches including our proposed system (submitted with two variants ($TwoPath_{writer}$) and ($TwoPath_{page}$)) is provided in Chapter 4 (cf. Section 4.4).

The performance is evaluated using the error metrics mAP, top-1 accuracy, $Pr@10$ and $Pr@100$. The results of Task 1, i.e., retrieving all similar fragment images matching the same writer ID, are summarized in Table 5.30. From the results of Task 1, it can be seen that the UBFC method achieved the best mAP score of 33.7%, followed by our CNN model $TwoPath_{writer}$ with a score of 33.5%. However, the $TwoPath_{writer}$ network is the best-performing system with the highest scores on top-1 accuracy, $Pr@10$ and $Pr@100$. The $TwoPath_{page}$ network is competitive in terms of accuracy; it is the top 3 best perform-

[1] https://github.com/anguelos/wi19evaluate

ing system with a score of 61.1%, and the top 4 system in terms of mAP, $Pr@10$ and $Pr@100$. As for the results of Task 2 (cf. Table 5.31) (retrieving all similar fragment images corresponding to the same page ID), the highest performance is achieved with our $TwoPath_{writer}$ network over all error metrics. The $TwoPath_{page}$ network is ranked second best method regarding the top 1 accuracy metric with a score of 27.4% and ranked the top 4 system in terms of the mAP, $Pr@10$ and $Pr@100$ metrics. We believe that the drop in performance when using the $TwoPath_{page}$ network compared to the other $TwoPath_{writer}$ variant is related to the training process. Since validation data was not available during the submission phase, the $TwoPath_{page}$ network was over-fitted, which affected the overall performance for fragment retrieval (from the test set). The performance of the RUG method is somewhat low. One possible reason for this could be the complex irregular shapes of the fragments, as the method fails to characterize the different amounts of text within these fragments accurately. Seuret et al. (2020) explained that this method was initially introduced for word images and had some training issues.

In summary, the mean Average Precision (mAP) determines the winner of the two tasks. Our proposed approach won first place in Task 2 and second place in Task 1.

Table 5.30: Experimental results of Task 1 (writer-level).

	#Method + Distance Metric	**#mAP**	**#Top-1 Accuracy**	**#Pr@10**	**#Pr@100**
RUG	FragNet + Euclidean	6.4	32.5	16.8	14.5
ULT	oBIF + Correlation	24.1	55.4	39.2	37.9
ICFHR2020 Baseline	SRS-LBP + Manhattan	33.4	60	46.8	45.9
UBFC	ResNet20$_{ssl}$ + Cosine	**33.7**	68.9	52.5	46.5
UTBM	$TwoPath_{writer}$ + Chi-square	33.5	**77.1**	**53.1**	**50.4**
	$TwoPath_{page}$ + Chi-square	25.2	61.1	41.2	44.1

Table 5.31: Experimental results of Task 2 (page-level).

	#Method + Distance Metric	**#mAP**	**#Top-1 Accuracy**	**#Pr@10**	**#Pr@100**
RUG	FragNet + Euclidean	4.1	8.4	6.9	16.6
ULT	oBIF + Correlation	16.1	23	22.4	45.5
ICFHR2020 Baseline	SRS-LBP + Manhattan	18.5	25.7	25.9	53.1
UBFC	ResNet20$_{ssl}$ + Cosine	18.4	24.1	26.2	53.2
UTBM	$TwoPath_{writer}$ + Chi-square	**22.6**	**36.4**	**31.2**	**58.9**
	$TwoPath_{page}$ + Chi-square	17.4	27.4	24.6	52.6

5.7/ CONCLUSION

In this chapter, we presented the experimental results obtained with the different methods we proposed, including CNN-based and texture-based writer identification systems. A comprehensive description of the benchmarks used with their standard protocol setup is also given. Extensive experiments are conducted on 10 challenging handwritten

databases in different languages, including English, Arabic, Dutch, Chinese, French, German, and Greek. For texture-based systems, we analyzed the key parameters of each feature technique and investigated the sensitivity of system performance as a function of the number of writers and the amount of data for each database tested. All of our proposed feature methods showed a high ability to handle and better characterize the writing variability. Moreover, the BWLBC method proved to be very efficient in computation time due to its small dimension size. Texture-based approaches achieved competitive, or the highest SOTA performance in the benchmarks studied.

CNN-based systems are evaluated under different conditions and configurations. For the $WriterINet$ framework, we analyzed how the number of neurons of the fully connected layer $FC1$ of the ANN network affects the overall performance through an extensive experiment. The $DeepWINet$ network was evaluated in two scenarios: scenario 1 consists of using $DeepWINet$ as a feature learning representation and the Chi-Square -nearest neighbor rule as a classifier; scenario 2 evaluates $DeepWINet$ as an end-to-end CNN network. Experiments have shown that scenario 1 leads to better performance compared to scenario 2. This is due to the dual comparison mechanism used in scenario 1 to classify deep feature vectors. In fact, the performance is further improved as distance and dissimilarity metrics are jointly used to compare and efficiently match deep features. In summary, our CNN-based systems proved their effectiveness and significantly improved the task of writer identification achieving excellent SOTA results. We also presented the experimental results of our submitted approach in the ICFHR2020 competition. Our approach achieved excellent results compared to the participants' methods.

III

CONCLUSION

6

CONCLUSIONS AND FUTURE WORKS

This chapter summarizes the contributions and research results presented in this thesis along with its main conclusions and recommendations for future research directions.

6.1/ SUMMARY AND CONTRIBUTIONS

Handwriting-based writer identification is based on two fundamental principles. Their clear statements are as follows: (1) no two individuals produce exactly the same writing style twice, and (2) no two individuals have the same writing style. These two natural factors make the writer identification a challenging task, considering the large within-writer and between-writer style variability. The main goal of this work was to automate the process of writer identification using scanned handwritten images (offline mode) to provide a complete computer analysis of the writing variability. Our proposed approaches for writer identification have an impact on forensic science. They enable the ”one-to-many” searching in an extensive handwritten database, where the output is a predicted sorted list of writer candidates corresponding to the samples matching the handwriting style of the query sample. In this way, the search space for comparing handwritten samples is reduced, and a hit list is an output that scribe analysts and forensic experts can quickly examine. The present thesis addressed all the main steps of an automatic system for writer identification using handwriting images. Our proposals contributed to the numerous challenges encountered in these steps: (i) image preprocessing and segmentation to discard background noise and extract local regions of interest (words, characters, connected components, etc.) from the handwriting images; (ii) feature extraction to capture a synthetic feature representation of the writing style to be recognized in a two-dimensional space. It consequently characterizes and indexes common writing patterns belonging to the same writing style; (iii) classification to compare and match the previously extracted features to identify the authorship of handwritten documents accurately.

Chapter 2 of the dissertation presented a comprehensive literature review of recent pub-

lications in the field of writer identification and provided the necessary context in which to place our research work. We have classified the state-of-the-art, in chronological order, into texture-based, grapheme-based, contour-based, and deep learning-based methods. We have also addressed he challenges, factors, performance, and criticisms of these approaches to better characterize the handwriting variability. After a careful review of the literature, we concluded that texture-based methods are conceptually simple but sometimes have a higher number of parameters, leading to an increase in processing time when used in conjunction with a classifier. In general, these methods capture recurring writing features and allow correct performance when using an appropriate classifier. For grapheme- and contour-based methods, the recognition and identification performance highly depends on the capability of the image segmentation phase as they are based on extracting features from small parts of the writing like graphemes, strokes, edges, and contours. With the advent of deep learning, the writing variability can be automatically extracted and learned using Convolutional Neural Network (CNN). CNN-based methods provide an accurate solution for the in-depth computation of features representing the individual's handwriting style. However, one of the drawbacks of these methods is the limitation of data size. Large labeled data is required for the training process to learn how to capture the within-writer and between-writer style variability. With less training data, traditional methods can perform better or equivalent to deep learning.

Our contributions in Chapter 3 focused on the feature extraction phase, as it is challenging to model the writing style patterns in the image. If the writing features are inappropriately captured from the writing, it will have an unfavorable impact on the classifier used to determine the writer's identity for the documents in question. Indeed, extracting relevant features allows to reduce misclassification and improve the writer identification task. We have proposed four texture-based systems for text-independent offline writer identification. An image preprocessing and segmentation step was commonly used for the four texture-based approaches. It consists of removing background noise and diacritical marks from the writing and segmenting the document image into connected component sub-images. In the feature extraction phase (for all texture-based approaches), these connected components are input to capture the texture information of the writing, which is represented by feature vectors. The classification phase is performed using the Hamming-based nearest neighbor method with a new strategy to compute similarities between handwritten documents. Our first approach is based on the Block Wise Local Binary Count (BWLBC) descriptor. It characterizes individual writing styles in small blocks by capturing the white pixels' distribution corresponding to the ink in binary component sub-images. The encoding of BW-LBC feature vectors depends on the number of blocks and the component sub-image window size. Therefore, these two parameters were tuned experimentally to find the optimal setting for better writing characterization.

The overall system was evaluated on four challenging handwritten databases, the Arabic

IFN/ENIT and AHTID/MW databases, and the English IAM and CVL databases. Unlike other SOTA approaches, where only a partition into training and test sets was used in the IFN/ENIT and IAM databases, the proposed BWLBC-based system was evaluated on ten different subsets randomly generated for each writer. For the CVL and AHTID/MW databases, the four cross-validation setup was used to test all possible subsets. These two protocol strategies were used to evaluate and validate the performance stability of the system thoroughly. Overall, the experimental results reported in Chapter 5 showed that the BWLBC-based approach achieved superior SOTA performance in the IFN/ENIT and AHTID/MW and competitive performance in the CVL and IAM. Moreover, the BWLBC feature method offers an excellent trade-off between classification accuracy and computational complexity thanks to its smaller feature histogram length. The second approach relies on the hand-crafted LBP, LPQ, and LTP descriptors applied to small regions (zones) of interest in connected components to capture textural features of the writing style. The final feature histogram representing each connected component comprises a set of sub-histograms computed sequentially within each zone. In this approach, we introduced a dimensionality reduction rule to reduce the computational cost of the subsequent classification process. As with the BWLBC method, the optimal settings of the feature extraction step's key parameters were determined through extensive experimentation for each database. From the results in Chapter 5, it can be concluded that the overall system achieved further improvements in writer identification performance over the previous BWLBC-based approach for the same benchmarks investigated.

Furthermore, we proposed two feature extraction methods, namely Cross multi-scale Locally encoded Gradient Patterns (CLGP) and Local gradient full- Scale Transform Patterns (LSTP). The CLGP descriptor computes the distribution of local intensity gradients within small regions, called cells, over multiple feature code maps. These feature code maps were uniformly computed from resized handwritten connected components using a cross-scale joint coding process. The LSTP method is an extended variant of CLGP for capturing relevant gradient representations of the writing patterns using the HOG operator. The gradient information is computed within non-overlapping blocks (of the component sub-image) in a full-scale code map (obtained using transformation features across multiple scale-spaces). The CLGP-based and LSTP-based approaches outperformed or provided competitive performance to SOTA systems on several benchmarks with different scripts, including English, Dutch, Chinese, Arabic, French, Greek, and German (results were reported in Chapter 5). For each texture-based approach proposed in Chapter 3, a series of experiments were conducted in Chapter 5 to evaluate and validate the stability of the system performance under different conditions. This includes the analysis of each approach's key parameters and the evaluation of the system sensitivity as a function of the number of writers and the amount of data for each benchmark studied.

In this thesis, we also investigated CNN-based deep learning approaches to improve

writer identification (Chapter 4). Our first CNN-based approach, called $WriterINet$, consists of the following steps: (1) segmenting handwritten samples (documents or lines of text) into word images and related component sub-images. Words were segmented using a scale-space approach based on blob analysis and Gaussian filters. Connected components were collected from the previously segmented words using a label-based bounding box technique to detect connected neighbor pixels, i.e., connected writing traces; (2) fine-tuning our multi-stream CNN model with different input data for feature learning. The global average pooling layers of the proposed model were activated to extract deep features of the input word and its associated component sub-images; (3) feeding the learned features into a conceptually simple and effective artificial neural network (1D- ANN) to identify the authorship of the query documents. We adapted the 1D- ANN classifier with two fully connected layers $FC1$ and $FC2$ to classify the CNN features efficiently. The number of neurons in the $FC2$ layer is defined as the number of output classes, while the number of neurons in the $FC1$ layer was experimentally tuned to find the optimal setting for better identification performance. This analysis experiment was provided in Chapter 5. Our second contribution using deep learning proposed a reliable and efficient deep CNN architecture, called $DeepWINet$: a full network version with 30-weighted layers and 24-weighted layers for the light network version (Chapter 4). Compared to the VGG network structure (144 million parameters), our $DeepWINet$ (light version) has a lower network complexity with only 22 million parameters. The $DeepWINet$ model was implemented and evaluated in two ways. The first scenario aimed to use the $DeepWINet$ model as a feature learning representation to extract deep features from the connected components of the writing. In the classification phase, the learned features were passed to a nearest-neighbor classifier that used the Chi-square metric as distance similarity to identify the query samples. For the second scenario, the $DeepWINet$ model was trained as an end-to-end CNN network, where the predicted scores are averaged using a new and efficient strategy, namely the score-averaging component-decision combiner. The results reported in Chapter 5 showed that our CNN-based approaches delivered high state-of-the-art performance and showed significant improvement over our previous texture-based systems for the task of writer identification.

To the best of our knowledge, we were one of the first to perform extensive experiments on ten challenging handwritten databases with different languages (English, Arabic, Dutch, Chinese, French, German, and Greek).

Another major finding of this thesis contributed to the task of historical documents retrieval based on writer identification. At the well-known ICFHR2020 conference, a competition was held to award the best approach to image retrieval of historical handwritten fragments. There were two main tasks: retrieving all similar fragments corresponding to the same writer ID (task 1) and the same page ID (task 2). We proposed an effective CNN-based approach consisting of two CNN streams, both used for deep feature

learning, each trained with different fragment patches from the ICFHR2020 training set. For the retrieval process, the Chi-square similarity measure was used to compute the distances between the fragment images (of the test set) characterized by their learned features (Chapter 4). Four contestant's approaches were submitted by different universities, including University Bourgogne Franche-Comte (UBFC), University of Groningen (RUG), and Tebessa University (ULT). As reported in Chapter 5, the proposed approach achieved excellent results, winning first place in task 2 (page retrieval) and second place in task 1 (writer retrieval).

6.2/ RESEARCH DIRECTIONS

This thesis has proposed several approaches based on texture and deep learning methods for automatic text-independent offline writer identification from handwriting. Although we have tried to explore as many aspects of the pipeline as possible, the scope of the work is a flourishing research topic with much room for further improvement. Here we provide some research suggestions and ideas.

The texture-based approaches presented in this work typically use a pre-processing step to clearly distinguish the writing ink from the background of the image (binarization process) and remove unwanted background noise while preserving the useful maximal information in the scanned image. In some databases like IAM, the original background of the samples is filled with unwanted random traces. During the processing of these samples, some representative writing traces are misclassified as noise and then discarded or lost during the segmentation step to extract related components. This limitation can affect the overall system to correctly identifying writers. To improve this pre-processing step, we plan to evaluate a deep binarization based on CNNs instead of the classical Otsu binarization method (Otsu (1979)), such as the U-Net model, which has been shown to be well suited to deal with the degradation of documents with complex backgrounds (Sadekar et al. (2021)). This will reduce the complexity of the subsequent segmentation algorithm to extract the connected component sub-images properly.

Recall that after image segmentation (Chapter 3), we resized the connected components to the same fixed window size to uniform the number of regions of interest in those components. The reason for using a fixed window size is to normalize the final dimension of the histogram output for the feature extraction step for all components. The original components (without resizing the window) consist of different characters with various shapes stored in separate bounding boxes. Therefore, after resizing the image, some of the components are either enlarged or compressed, which could affect the feature method to characterize the writing content within these resized components correctly. To solve this problem, we can perform dynamic resizing of the window based on the original pixels,

i.e., the process can be performed iteratively depending on the number of writing pixels (the ink) on each component sub-image separately. For example, if one or more words become mostly a single connected component, the algorithm will roughly segment the component into a number of subcomponents based on the fixed threshold representing the writing pixels' distribution and the selected window size. This would unify the regions of interest of the writing without overwriting the original shape of the text content.

To increase the scale complexity of the benchmarks studied in this work, it will be interesting to mix English handwritten databases (IAM + CVL + ICDAR2011) to create a large-scale database with more data and classes (the same suggestion can be made for the other languages). The interest is to investigate further writer identification in a mixed-script environment, which could be very potential and challenging for researchers who want to evaluate their proposals on large-scale benchmarks. Additionally, we plan to evaluate the robustness of our proposals (to be compared with other SOTA methods) by adding noise density (salt and pepper), contrast change, distortion, and blur to the handwritten samples. Moreover, it could be valuable to extend the evaluation of our proposals to other challenging databases such as the QUWI and other datasets of the ICFHR and ICDAR conference series, while proposing to evaluate other types of feature methods with their possible combinations and to develop new and effective algorithms for characterizing writing style.

Methods based on CNN offer an accurate solution for writer identification. However, large handwriting training data is needed to learn how to characterize the individual's handwriting style, which is not always provided in some benchmarks. In this case, we believe that the Generative Adversarial Networks or GANs can be used as a tool for data augmentation by generating slightly modified handwriting samples of the benchmark in question while preserving the writer-related features, i.e., creating synthetic handwriting data from other existing benchmarks with the same script. Specifically, the generator model (in the GAN architecture) will learn how to generate new possible handwriting data relying on the adversarial discriminator feedback, which attempts to discriminate between real images from the training datasets and new images output by the generator model. The augmented data would allow the CNN models to reduce the overfitting of the training process and then improve the identification task.

From the results in Chapter 5, it can be seen that our proposed CLGP-based and LSTP-based approaches require more time to compare the writing samples based on the Hamming distance of their respective feature histograms. This is mainly due to the size dimensionality of the CLGP and LSTP features, although processing time is not necessarily a critical performance indicator for offline writer identification since real-time applications are not required (offline mode). Post-processing methods such as principal component analysis (PCA), linear discriminant analysis (LDA), generalized discriminant

analysis (GDA), and auto-encoders can be used to transform the CLGP and LSTP features into a low-dimensional representation space. This helps to reduce the overcount of feature co-occurrences and results in fewer parameters and computations in the classification phase. Moreover, Field-Programmable Gate Array (FPGA) devices can also be used as implementation platforms to speed up the system thanks to their structure capable of exploiting spatial and temporal parallelism.

Writer identification can be applied to historical documents. Automatic algorithms for script identification of historical documents can serve as a valuable means for historians. Our proposal in the competition on image retrieval of historical handwritten fragments (Chapter 4) motivates us to investigate this topic further, which may be a worthwhile future research direction. To this end, we plan to evaluate the fusion of textural features, mainly the oriented Basic Image Feature (oBIF) and CNN-based features (design of a new CNN architecture). Technically, the feature map of oBIF (instead of the original image pixels) is fed into the CNN model to improve the deep feature extraction step. Note that oBIF features have shown their effectiveness for high-level characterization of texture patterns in character recognition (Newell and Griffin (2011)) and texture recognition (Timofte and Van Gool (2012); Newell et al. (2010)). Identifying writers in historical documents would open up some interesting perspectives, such as identifying the printing house in machine-printed documents, automatically dating historical documents, and indexing and retrieving manuscripts based on writing style.

We would also like to explore signature verification, one of the most potential applications of handwriting analysis. It is a challenging task considering the between-genuine signature variability, i.e., the same person can produce quite different signatures. The interest is to evaluate our previous approaches (for writer identification) and develop a new framework to deal with the individual's signature style. A well-defined automated signature verification system can assist human experts in verifying bank checks' authenticity and reducing fraud. Similarly, gender classification based on handwriting will be investigated as another research direction, using new feature methods to capture visual and recurring patterns that distinguish between male and female handwriting.

In the classification step, an additional verification step can be added to improve the identification results. The idea is to develop a verification system based on the handwriting that rechecks whether the probe sample matches the class predicted by the classification module. This may correct the prediction output. Technically, the k-predicted handwriting samples that are most similar to the query are retrieved from the reference database in a hit list. This list will be element-wise examined by the verification system using a binary classification rule. The query is compared to each sample in the list, and this process is repeated k times. Each time, the verification system would answer the question of whether these two handwriting samples (query and another sample from the k-hit list) are

from the same person or not. If the decision matches the class predicted by the classification system, the probe class is validated as the final output of the overall system. If the two decisions are mismatching, the probe sample is reclassified, ignoring the first predicted class. The mismatching procedure is repeated until reaching the match.

BIBLIOGRAPHY

[Abdeljalil et al. 2018] ABDELJALIL, Gattal ; DJEDDI, Chawki ; SIDDIQI, Imran ; ALMAADEED, Somaya: **"Writer identification on historical documents using oriented basic image features"**. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)* IEEE (event), 2018, pages 369–373

[Abdi and Khemakhem 2015] ABDI, Mohamed N. ; KHEMAKHEM, Maher: **"A model-based approach to offline text-independent Arabic writer identification and verification"**. In *Pattern Recognition* 48 (2015), number 5, pages 1890–1903. – ISSN 0031-3203

[Abdi et al. 2009] ABDI, Mohamed N. ; KHEMAKHEM, Maher ; BEN-ABDALLAH, Hanene: **"An effective combination of MPP contour-based features for off-line text-independent arabic writer identification"**. In *Signal processing, image processing and pattern recognition*. Springer, 2009, pages 209–220

[Amend and Ruiz 2000] AMEND, Karen K. ; RUIZ, Mary S.: *Handwriting analysis: The complete basic book*. Red Wheel/Weiser, 2000

[Arabadjis et al. 2013] ARABADJIS, D. ; GIANNOPOULOS, F. ; PAPAODYSSEUS, C. ; ZANNOS, S. ; ROUSOPOULOS, P. ; PANAGOPOULOS, M. ; BLACKWELL, C.: **"New mathematical and algorithmic schemes for pattern classification with application to the identification of writers of important ancient documents"**. In *Pattern Recognition* 46 (2013), number 8, pages 2278 – 2296. – ISSN 0031-3203

[Awaida and Mahmoud 2013] AWAIDA, Sameh M. ; MAHMOUD, Sabri A.: **"Writer identification of arabic text using statistical and structural features"**. In *Cybernetics and Systems* 44 (2013), number 1, pages 57–76

[Bensefia et al. 2002] BENSEFIA, Ameur ; NOSARY, Ali ; PAQUET, Thierry ; HEUTTE, Laurent: **"Writer identification by writer's invariants"**. In *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition* IEEE (event), 2002, pages 274–279

[Bensefia et al. 2003] BENSEFIA, Ameur ; PAQUET, Thierry ; HEUTTE, Laurent: **"Information Retrieval Based Writer Identification."**. In *PRIS* Citeseer (event), 2003, pages 56–63

[Bensefia et al. 2005] BENSEFIA, Ameur ; PAQUET, Thierry ; HEUTTE, Laurent: **"Handwritten document analysis for automatic writer recognition"**. In *ELCVIA: electronic letters on computer vision and image analysis* (2005), pages 72–86

[Bertolini et al. 2013] BERTOLINI, D. ; OLIVEIRA, L.S. ; JUSTINO, E. ; SABOURIN, R.: **"Texture-based descriptors for writer identification and verification"**. In *Expert Systems with Applications* 40 (2013), number 6, pages 2069 – 2080. – ISSN 0957-4174

[Brink et al. 2012a] BRINK, A. ; SMIT, J. ; BULACU, M. ; SCHOMAKER, L.: **"Writer identification using directional ink-trace width measurements"**. In *Pattern Recognition* 45 (2012), number 1, pages 162–171

[Brink et al. 2012b] BRINK, AA ; SMIT, J ; BULACU, ML ; SCHOMAKER, LRB: **"Writer identification using directional ink-trace width measurements"**. In *Pattern Recognition* 45 (2012), number 1, pages 162–171

[Bulacu and Schomaker 2005] BULACU, Marius ; SCHOMAKER, Lambert: **"A comparison of clustering methods for writer identification and verification"**. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)* IEEE (event), 2005, pages 1275–1279

[Bulacu and Schomaker 2006] BULACU, Marius ; SCHOMAKER, Lambert: **"Combining multiple features for text-independent writer identification and verification"**. In *Tenth International Workshop on Frontiers in Handwriting Recognition* Suvisoft (event), 2006

[Bulacu and Schomaker 2007] BULACU, Marius ; SCHOMAKER, Lambert: **"Text-independent writer identification and verification using textural and allographic features"**. In *IEEE transactions on pattern analysis and machine intelligence* 29 (2007), number 4, pages 701–717

[Bulacu et al. 2007] BULACU, Marius ; SCHOMAKER, Lambert ; BRINK, Axel: **"Text-independent writer identification and verification on offline arabic handwriting"**. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on* Volume 2 IEEE (event), 2007, pages 769–773

[Bulacu 2007] BULACU, Marius L.: **"Statistical pattern recognition for automatic writer identification and verification"**. (2007)

[Chahi et al. 2018] CHAHI, Abderrazak ; ELKHADIRI, Issam ; ELMERABET, Youssef ; RUICHEK, Yassine ; TOUAHNI, Raja: **"Block wise local binary count for off-Line text-independent writer identification"**. In *Expert Systems with Applications* 93 (2018), number Supplement C, pages 1 – 14

[Chahi et al. 2019] CHAHI, Abderrazak ; ELMERABET, Youssef ; RUICHEK, Yassine ; TOUAHNI, Raja: **"An effective and conceptually simple feature representation for off-line text-independent writer identification"**. In *Expert Systems with Applications* 123 (2019), pages 357 – 376. – ISSN 0957-4174

[Chahi et al. 2020a] CHAHI, Abderrazak ; ELMERABET, Youssef ; RUICHEK, Yassine ; TOUAHNI, Raja: **"Cross multi-scale locally encoded gradient patterns for off-line text-independent writer identification"**. In *Engineering Applications of Artificial Intelligence* 89 (2020), pages 103459

[Chahi et al. 2020b] CHAHI, Abderrazak ; ELMERABET, Youssef ; RUICHEK, Yassine ; TOUAHNI, Raja: **"Local gradient full-scale transform patterns based off-line text-independent writer identification"**. In *Applied Soft Computing* 92 (2020), pages 106277

[Chawki and Labiba 2010] CHAWKI, Djeddi ; LABIBA, Souici-Meslati: **"A texture based approach for Arabic Writer Identification and Verification"**. In *Machine and Web Intelligence (ICMWI), 2010 International Conference on* IEEE (event), 2010, pages 115–120

[Chen et al. 2019] CHEN, Shiming ; WANG, Yisong ; LIN, Chin-Teng ; DING, Weiping ; CAO, Zehong: **"Semi-supervised feature learning for improving writer identification"**. In *Information Sciences* 482 (2019), pages 156 – 170

[Christlein et al. 2015a] CHRISTLEIN, Vincent ; BERNECKER, David ; ANGELOPOULOU, Elli: **"Writer identification using vlad encoded contour-zernike moments"**. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)* IEEE (event), 2015, pages 906–910

[Christlein et al. 2014] CHRISTLEIN, Vincent ; BERNECKER, David ; HÖNIG, Florian ; ANGELOPOULOU, Elli: **"Writer identification and verification using GMM supervectors"**. In *IEEE Winter Conference on Applications of Computer Vision* IEEE (event), 2014, pages 998–1005

[Christlein et al. 2017a] CHRISTLEIN, Vincent ; BERNECKER, David ; HÖNIG, Florian ; MAIER, Andreas ; ANGELOPOULOU, Elli: **"Writer identification using GMM supervectors and exemplar-SVMs"**. In *Pattern Recognition* 63 (2017), pages 258–267

[Christlein et al. 2015b] CHRISTLEIN, Vincent ; BERNECKER, David ; MAIER, Andreas ; ANGELOPOULOU, Elli: **"Offline writer identification using convolutional neural network activation features"**. In *German Conference on Pattern Recognition* Springer (event), 2015, pages 540–552

[Christlein et al. 2017b] CHRISTLEIN, Vincent ; GROPP, Martin ; FIEL, Stefan ; MAIER, Andreas: **"Unsupervised feature learning for writer identification and writer retrieval"**. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)* Volume 1 IEEE (event), 2017, pages 991–997

[Christlein and Maier 2018] CHRISTLEIN, Vincent ; MAIER, Andreas: **"Encoding CNN activations for writer recognition"**. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)* IEEE (event), 2018, pages 169–174

[Christlein et al. 2019] CHRISTLEIN, Vincent ; NICOLAOU, Anguelos ; SEURET, Mathias ; STUTZMANN, Dominique ; MAIER, Andreas: **"ICDAR 2019 competition on image retrieval for historical handwritten documents"**. In *2019 International Conference on Document Analysis and Recognition (ICDAR)* IEEE (event), 2019, pages 1505–1509

[Crosier and Griffin 2010] CROSIER, Michael ; GRIFFIN, Lewis D.: **"Using basic image features for texture classification"**. In *International journal of computer vision* 88 (2010), number 3, pages 447–460

[Dalal and Triggs 2005] DALAL, Navneet ; TRIGGS, Bill: **"Histograms of oriented gradients for human detection"**. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* Volume 1 IEEE (event), 2005, pages 886–893

[Dargan and Kumar 2019] DARGAN, Shaveta ; KUMAR, Munish: **"Writer identification System for Indic and non-Indic scripts: state-of-the-art survey"**. In *Archives of Computational Methods in Engineering* 26 (2019), number 4, pages 1283–1311

[Djeddi et al. 2013] DJEDDI, Chawki ; SIDDIQI, Imran ; SOUICI-MESLATI, Labiba ; ENNAJI, Abdellatif: **"Text-independent writer recognition using multi-script handwritten texts"**. In *Pattern Recognition Letters* 34 (2013), number 10, pages 1196–1202

[Durou et al. 2019] DUROU, Amal ; AREF, Ibrahim ; AL-MAADEED, Somaya ; BOURIDANE, Ahmed ; BENKHELIFA, Elhadj: **"Writer identification approach based on bag of words with OBI features"**. In *Information Processing & Management* 56 (2019), number 2, pages 354–366

[Ferrari et al. 2007] FERRARI, Vittorio ; FEVRIER, Loic ; JURIE, Frederic ; SCHMID, Cordelia: **"Groups of adjacent contour segments for object detection"**. In *IEEE transactions on pattern analysis and machine intelligence* 30 (2007), number 1, pages 36–51

[Fiel and Sablatnig 2013] FIEL, S. ; SABLATNIG, R.: **"Writer Identification and Writer Retrieval Using the Fisher Vector on Visual Vocabularies"**. In *2013 12th Interna-*

tional Conference on Document Analysis and Recognition, Aug 2013, pages 545–549. – ISSN 1520-5363

[Fiel et al. 2017] FIEL, Stefan ; KLEBER, Florian ; DIEM, Markus ; CHRISTLEIN, Vincent ; LOULOUDIS, Georgios ; NIKOS, Stamatopoulos ; GATOS, Basilis: **"Icdar2017 competition on historical document writer identification (historical-wi)"**. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)* Volume 1 IEEE (event), 2017, pages 1377–1382

[Fiel and Sablatnig 2015] FIEL, Stefan ; SABLATNIG, Robert: **"Writer identification and retrieval using a convolutional neural network"**. In *International Conference on Computer Analysis of Images and Patterns* Springer (event), 2015, pages 26–37

[Fornés et al. 2008] FORNÉS, A. ; LLADÓS, J. ; SÁNCHEZ, G. ; BUNKE, H.: **"Writer Identification in Old Handwritten Music Scores"**. In *2008 The Eighth IAPR International Workshop on Document Analysis Systems*, Sept 2008, pages 347–353

[Fournier et al. 1982] FOURNIER, Alain ; FUSSELL, Don ; CARPENTER, Loren: **"Computer rendering of stochastic models"**. In *Communications of the ACM* 25 (1982), number 6, pages 371–384

[Francks et al. 2003] FRANCKS, Clyde ; DELISI, Lynn E. ; FISHER, Simon E. ; LAVAL, Steve H. ; RUE, Judith E. ; STEIN, John F. ; MONACO, Anthony P.: **"Confirmatory evidence for linkage of relative hand skill to 2p12-q11"**. In *American Journal of Human Genetics* 72 (2003), number 2, pages 499

[Franke and Köppen 2001] FRANKE, Katrin ; KÖPPEN, Mario: **"A computer-based system to support forensic studies on handwritten documents"**. In *International Journal on Document Analysis and Recognition* 3 (2001), number 4, pages 218–231. – ISSN 1433-2833

[Freeman et al. 1991] FREEMAN, William T. ; ADELSON, Edward H. ; OTHERS: **"The design and use of steerable filters"**. In *IEEE Transactions on Pattern analysis and machine intelligence* 13 (1991), number 9, pages 891–906

[Frias-Martinez et al. 2006] FRIAS-MARTINEZ, Enrique ; SANCHEZ, Angel ; VELEZ, Jose: **"Support vector machines versus multi-layer perceptrons for efficient off-line signature recognition"**. In *Engineering Applications of Artificial Intelligence* 19 (2006), number 6, pages 693–704

[Gaceb et al. 2006] GACEB, Djamel ; EGLIN, Véronique ; BRES, Stéphane ; EMPTOZ, Hubert: **"Handwriting similarities as features for the characterization of writer's style invariants and image compression"**. In *International Conference Image Analysis and Recognition* Springer (event), 2006, pages 776–789

[Garz et al. 2016] GARZ, Angelika ; WÜRSCH, Marcel ; FISCHER, Andreas ; INGOLD, Rolf: **"Simple and fast geometrical descriptors for writer identification"**. In *Electronic Imaging* 2016 (2016), number 17, pages 1–12

[Gattal et al. 2016] GATTAL, Abdeljalil ; DJEDDI, Chawki ; CHIBANI, Youcef ; SIDDIQI, Imran: **"Isolated handwritten digit recognition using oBIFs and background features"**. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)* IEEE (event), 2016, pages 305–310

[Ghiasi and Safabakhsh 2013] GHIASI, Golnaz ; SAFABAKHSH, Reza: **"Offline text-independent writer identification using codebook and efficient code extraction methods"**. In *Image and Vision Computing* 31 (2013), number 5, pages 379–391

[Goel 2019] GOEL, Harshit: **"Comparison of various PCA algorithms"**. In *IIT Delhi* (2019)

[Gordo et al. 2013] GORDO, Albert ; PERRONNIN, Florent ; VALVENY, Ernest: **"Large-scale document image retrieval and classification with runlength histograms and binary embeddings"**. In *Pattern Recognition* 46 (2013), number 7, pages 1898–1905

[Hadjadji and Chibani 2018] HADJADJI, Bilal ; CHIBANI, Youcef: **"Two combination stages of clustered One-Class Classifiers for writer identification from text fragments"**. In *Pattern Recognition* 82 (2018), pages 147–162

[Hannad et al. 2016] HANNAD, Yaacoub ; IMRAN ; KETTANI, Mohamed El Youssfi E.: **"Writer identification using texture descriptors of handwritten fragments"**. In *Expert Systems with Applications* 47 (2016), pages 14 – 22. – ISSN 0957-4174

[Hanusiak et al. 2012] HANUSIAK, Regiane K. ; OLIVEIRA, Luiz S. ; JUSTINO, E ; SABOURIN, Robert: **"Writer verification using texture-based features"**. In *International Journal on Document Analysis and Recognition (IJDAR)* 15 (2012), number 3, pages 213–226

[Harrison et al. 2009] HARRISON, Diana ; BURKES, Ted M. ; SEIGER, Danielle P.: **"Handwriting examination: Meeting the challenges of science and the law"**. In *Forensic Science Communications* 11 (2009), number 4, pages 1–13

[Hassaïne and Al Maadeed 2012] HASSAÏNE, Abdelâali ; AL MAADEED, Somaya: **"ICFHR 2012 competition on writer identification challenge 2: Arabic scripts"**. In *2012 International Conference on Frontiers in Handwriting Recognition* IEEE (event), 2012, pages 835–840

[He et al. 2016a] HE, Kaiming ; ZHANG, Xiangyu ; REN, Shaoqing ; SUN, Jian: **"Deep residual learning for image recognition"**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pages 770–778

[He et al. 2015] HE, S. ; WIERING, M. ; SCHOMAKER, L.: **"Junction detection in handwritten documents and its application to writer identification"**. In *Pattern Recognition* 48 (2015), number 12, pages 4036–4048

[He et al. 2016b] HE, Sheng ; SAMARA, Petros ; BURGERS, Jan ; SCHOMAKER, Lambert: **"Image-based historical manuscript dating using contour and stroke fragments"**. In *Pattern Recognition* 58 (2016), pages 159–171

[He and Schomaker 2014] HE, Sheng ; SCHOMAKER, Lambert: **"Delta-n hinge: rotation-invariant features for writer identification"**. In *2014 22nd International Conference on Pattern Recognition* IEEE (event), 2014, pages 2023–2028

[He and Schomaker 2016] HE, Sheng ; SCHOMAKER, Lambert: **"General pattern run-length transform for writer identification"**. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)* IEEE (event), 2016, pages 60–65

[He and Schomaker 2017a] HE, Sheng ; SCHOMAKER, Lambert: **"Beyond OCR: Multi-faceted understanding of handwritten document characteristics"**. In *Pattern Recognition* 63 (2017), pages 321–333

[He and Schomaker 2017b] HE, Sheng ; SCHOMAKER, Lambert: **"Writer identification using curvature-free features"**. In *Pattern Recognition* 63 (2017), pages 451–464

[He and Schomaker 2020] HE, Sheng ; SCHOMAKER, Lambert: **"Fragnet: Writer identification using deep fragment networks"**. In *IEEE Transactions on Information Forensics and Security* 15 (2020), pages 3013–3022

[He et al. 2010] HE, Zhenyu ; YOU, Xinge ; ZHOU, Long ; CHEUNG, Yiuming ; DU, Jianwei: **"Writer identification using fractal dimension of wavelet subbands in gabor domain"**. In *Integrated Computer-Aided Engineering* 17 (2010), number 2, pages 157–165

[He and Tang 2004] HE, ZY ; TANG, YY: **"Chinese handwriting-based writer identification by texture analysis"**. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)* Volume 6 IEEE (event), 2004, pages 3488–3491

[Helli and Moghaddam 2010] HELLI, Behzad ; MOGHADDAM, Mohsen E.: **"A text-independent Persian writer identification based on feature relation graph (FRG)"**. In *Pattern Recognition* 43 (2010), number 6, pages 2199–2209

[Huang et al. 2017] HUANG, Gao ; LIU, Zhuang ; VAN DER MAATEN, Laurens ; WEINBERGER, Kilian Q.: **"Densely connected convolutional networks"**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pages 4700–4708

[Jain and Doermann 2011] JAIN, Rajiv ; DOERMANN, David: **"Offline writer identification using k-adjacent segments"**. In *2011 International Conference on Document Analysis and Recognition* IEEE (event), 2011, pages 769–773

[Jain and Doermann 2014] JAIN, Rajiv ; DOERMANN, David: **"Combining local features for offline writer identification"**. In *2014 14th International Conference on Frontiers in Handwriting Recognition* IEEE (event), 2014, pages 583–588

[Javidi and Jampour 2020] JAVIDI, Malihe ; JAMPOUR, Mahdi: **"A deep learning framework for text-independent writer identification"**. In *Engineering Applications of Artificial Intelligence* 95 (2020), pages 103912

[Jégou and Zisserman 2014] JÉGOU, Hervé ; ZISSERMAN, Andrew: **"Triangulation embedding and democratic aggregation for image search"**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pages 3310–3317

[John et al. 1994] JOHN, George H. ; KOHAVI, Ron ; PFLEGER, Karl: **"Irrelevant features and the subset selection problem"**. In *Machine Learning Proceedings 1994*. Elsevier, 1994, pages 121–129

[Kanetkar et al. 2016] KANETKAR, Salil ; PATHANIA, Ayush ; VENUGOPAL, Vivek ; SUNDARAM, Suresh: **"Offline Writer Identification Using Local Derivative Pattern"**. In *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on* IEEE (event), 2016, pages 355–360

[Keglevic et al. 2018] KEGLEVIC, Manuel ; FIEL, Stefan ; SABLATNIG, Robert: **"Learning Features for Writer Retrieval and Identification using Triplet CNNs"**. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)* IEEE (event), 2018, pages 211–216

[Kessentini et al. 2018] KESSENTINI, Yousri ; BENABDERRAHIM, Sana ; DJEDDI, Chawki: **"Evidential combination of SVM classifiers for writer recognition"**. In *Neurocomputing* 313 (2018), pages 1–13

[Khalifa et al. 2015] KHALIFA, Emad ; AL-MAADEED, Somaya ; TAHIR, Muhammad A. ; BOURIDANE, Ahmed ; JAMSHED, Asif: **"Off-line writer identification using an ensemble of grapheme codebook features"**. In *Pattern Recognition Letters* 59 (2015), pages 18–25

[Khan et al. 2019] KHAN, F. A. ; KHELIFI, F. ; TAHIR, M. A. ; BOURIDANE, A.: **"Dissimilarity Gaussian Mixture Models for Efficient Offline Handwritten Text-Independent Identification Using SIFT and RootSIFT Descriptors"**. In *IEEE Transactions on Information Forensics and Security* 14 (2019), Feb, number 2, pages 289–303

[Khan et al. 2016] KHAN, F. A. ; TAHIR, M. A. ; KHELIFI, F. ; BOURIDANE, A.: **"Offline text independent writer identification using ensemble of multi-scale local ternary pattern histograms"**. In *2016 6th European Workshop on Visual Information Processing (EUVIP)*, Oct 2016, pages 1–6. DOI: 10.1109/EUVIP.2016.7764587

[Khan et al. 2017] KHAN, Faraz A. ; TAHIR, Muhammad A. ; KHELIFI, Fouad ; BOURIDANE, Ahmed ; ALMOTAERYI, Resheed: **"Robust off-line text independent writer identification using bagged discrete cosine transform features"**. In *Expert Systems with Applications* 71 (2017), pages 404–415

[Kingma and Ba 2014] KINGMA, Diederik P. ; BA, Jimmy: **"Adam: A method for stochastic optimization"**. In *arXiv preprint arXiv:1412.6980* (2014)

[Kleber et al. 2013] KLEBER, F. ; S. ; DIEM, M. ; SABLATNIG, R.: **"CVL-DataBase: An Off-Line Database for Writer Retrieval, Writer Identification and Word Spotting"**. In *2013 12th International Conference on Document Analysis and Recognition*, Aug 2013, pages 560–564. – ISSN 1520-5363

[Krizhevsky et al. 2012] KRIZHEVSKY, Alex ; SUTSKEVER, Ilya ; HINTON, Geoffrey E.: **"Imagenet classification with deep convolutional neural networks"**. In *Advances in neural information processing systems*, 2012, pages 1097–1105

[Kumar and Sharma 2019] KUMAR, Parveen ; SHARMA, Ambalika: **"DCWI: Distribution descriptive curve and Cellular automata based Writer Identification"**. In *Expert Systems with Applications* 128 (2019), pages 187–200

[Kumar and Sharma 2020] KUMAR, Parveen ; SHARMA, Ambalika: **"Segmentation-free writer identification based on convolutional neural network"**. In *Computers & Electrical Engineering* 85 (2020), pages 106707

[Kumar et al. 2014] KUMAR, Rajesh ; CHANDA, Bhabatosh ; SHARMA, JD: **"A novel sparse model based forensic writer identification"**. In *Pattern Recognition Letters* 35 (2014), pages 105–112

[Lai et al. 2020] LAI, Songxuan ; ZHU, Yecheng ; JIN, Lianwen: **"Encoding Pathlet and SIFT Features With Bagged VLAD for Historical Writer Identification"**. In *IEEE Transactions on Information Forensics and Security* 15 (2020), pages 3553–3566

[Lee et al. 2013] LEE, Seung E. ; MIN, Kyungwon ; SUH, Taeweon: **"Accelerating histograms of oriented gradients descriptor extraction for pedestrian recognition"**. In *Computers & Electrical Engineering* 39 (2013), number 4, pages 1043–1048

[Liwicki et al. 2006] LIWICKI, Marcus ; SCHLAPBACH, Andreas ; BUNKE, Horst ; BENGIO, Samy ; MARIÉTHOZ, Johnny ; RICHIARDI, Jonas: *Writer Identification for Smart*

Meeting Room Systems. pages 186–195. In *Document Analysis Systems VII: 7th International Workshop, DAS 2006, Nelson, New Zealand, February 13-15, 2006. Proceedings*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2006. – ISBN 978-3-540-32157-6

[Louloudis et al. 2013] LOULOUDIS, Georgios ; GATOS, Basilios ; STAMATOPOULOS, Nikolaos ; PAPANDREOU, A: **"Icdar 2013 competition on writer identification"**. In *2013 12th International Conference on Document Analysis and Recognition* IEEE (event), 2013, pages 1397–1401

[Louloudis et al. 2011] LOULOUDIS, Georgios ; STAMATOPOULOS, Nikolaos ; GATOS, Basilios: **"ICDAR 2011 writer identification contest"**. In *Document analysis and recognition (ICDAR), 2011 international conference on* IEEE (event), 2011, pages 1475–1479

[MacQueen et al. 1967] MACQUEEN, James ; OTHERS: **"Some methods for classification and analysis of multivariate observations"**. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* Volume 1 Oakland, CA, USA (event), 1967, pages 281–297

[Manmatha and Srimal 1999] MANMATHA, Raghavan ; SRIMAL, Nitin: **"Scale space technique for word segmentation in handwritten documents"**. In *International conference on scale-space theories in computer vision* Springer (event), 1999, pages 22–33

[Marti and Bunke 2002] MARTI, U-V ; BUNKE, Horst: **"The IAM-database: an English sentence database for offline handwriting recognition"**. In *International Journal on Document Analysis and Recognition* 5 (2002), number 1, pages 39–46

[Mezghani et al. 2012] MEZGHANI, A. ; KANOUN, S. ; KHEMAKHEM, M. ; ABED, H. E.: **"A Database for Arabic Handwritten Text Image Recognition and Writer Identification"**. In *2012 International Conference on Frontiers in Handwriting Recognition*, Sept 2012, pages 399–402

[Miller et al. 2017] MILLER, John J. ; PATTERSON, Robert B. ; GANTZ, Donald T. ; SAUNDERS, Christopher P. ; WALCH, Mark A. ; BUSCAGLIA, JoAnn: **"A set of handwriting features for use in automated writer identification"**. In *Journal of forensic sciences* 62 (2017), number 3, pages 722–734

[Mohammed et al. 2017] MOHAMMED, Hussein ; MÄERGNER, Volker ; KONIDARIS, Thomas ; STIEHL, H S.: **"Normalised Local Naïve Bayes Nearest-Neighbour Classifier for Offline Writer Identification"**. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)* Volume 1 IEEE (event), 2017, pages 1013–1018

[Mohammed et al. 2018] MOHAMMED, Hussein ; MÄRGNER, Volker ; STIEHL, H S.: **"Writer identification for historical manuscripts: Analysis and optimisation of a classifier as an easy-to-use tool for scholars from the humanities"**. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)* IEEE (event), 2018, pages 534–539

[Nejad and Rahmati 2007] NEJAD, F ; RAHMATI, Mohammad: **"A new method for writer identification and verification based on Farsi/Arabic handwritten texts"**. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)* Volume 2 IEEE (event), 2007, pages 829–833

[Newell and Griffin 2011] NEWELL, Andrew J. ; GRIFFIN, Lewis D.: **"Natural Image Character Recognition Using Oriented Basic Image Features."**. In *DICTA*, 2011, pages 191–196

[Newell and Griffin 2014] NEWELL, Andrew J. ; GRIFFIN, Lewis D.: **"Writer identification using oriented basic image features and the delta encoding"**. In *Pattern Recognition* 47 (2014), number 6, pages 2255–2265

[Newell et al. 2010] NEWELL, Andrew J. ; GRIFFIN, Lewis D. ; MORGAN, Ruth M. ; BULL, Peter A.: **"Texture-based estimation of physical characteristics of sand grains"**. In *2010 International Conference on Digital Image Computing: Techniques and Applications* IEEE (event), 2010, pages 504–509

[Nguyen et al. 2019] NGUYEN, Hung T. ; NGUYEN, Cuong T. ; INO, Takeya ; INDURKHYA, Bipin ; NAKAGAWA, Masaki: **"Text-independent writer identification using convolutional neural network"**. In *Pattern Recognition Letters* 121 (2019), pages 104 – 112. – Graphonomics for e-citizens: e-health, e-society, e-education. – ISSN 0167-8655

[Nicolaou et al. 2015] NICOLAOU, Anguelos ; BAGDANOV, Andrew D. ; LIWICKI, Marcus ; KARATZAS, Dimosthenis: **"Sparse radial sampling lbp for writer identification"**. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on* IEEE (event), 2015, pages 716–720

[Nicolaou et al. 2018] NICOLAOU, Anguelos ; DEY, Sounak ; CHRISTLEIN, Vincent ; MAIER, Andreas ; KARATZAS, Dimosthenis: **"Non-deterministic Behavior of Ranking-based Metrics when Evaluating Embeddings"**. In *International Workshop on Reproducible Research in Pattern Recognition* Springer (event), 2018, pages 71–82

[Ojala et al. 2002] OJALA, T. ; PIETIKAINEN, M. ; MAENPAA, T.: **"Multiresolution gray-scale and rotation invariant texture classification with local binary patterns"**. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002), Jul, number 7, pages 971–987. – ISSN 0162-8828

[Ojansivu and Heikkilä 2008] OJANSIVU, Ville ; HEIKKILÄ, Janne: *Blur Insensitive Texture Classification Using Local Phase Quantization*. pages 236–243. In *Image and Signal Processing: 3rd International Conference, ICISP 2008. Cherbourg-Octeville, France, July 1 - 3, 2008. Proceedings*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008. – ISBN 978-3-540-69905-7

[Otsu 1979] OTSU, Nobuyuki: **"A threshold selection method from gray-level histograms"**. In *IEEE transactions on systems, man, and cybernetics* 9 (1979), number 1, pages 62–66

[Pandey and Seeja 2018] PANDEY, Pallavi ; SEEJA, KR: **"Forensic writer identification with projection profile representation of graphemes"**. In *Proceedings of First International Conference on Smart System, Innovations and Computing* Springer (event), 2018, pages 129–136

[Pechwitz et al. 2002] PECHWITZ, Mario ; MADDOURI, Samia S. ; MÄRGNER, Volker ; ELLOUZE, Noureddine ; AMIRI, Hamid: **"IFN/ENIT - database of handwritten Arabic words"**. In *In Proc. of CIFED 2002*, 2002, pages 129–136

[Pervouchine and Leedham 2007] PERVOUCHINE, Vladimir ; LEEDHAM, Graham: **"Extraction and analysis of forensic document examiner features used for writer identification"**. In *Pattern Recognition* 40 (2007), number 3, pages 1004–1013

[Rehman et al. 2019] REHMAN, Arshia ; NAZ, Saeeda ; RAZZAK, Muhammad I.: **"Writer identification using machine learning approaches: a comprehensive review"**. In *Multimedia Tools and Applications* 78 (2019), Apr, number 8, pages 10889–10931

[Sadekar et al. 2021] SADEKAR, Kaustubh ; SINGH, Prajwal ; RAMAN, Shanmuganathan: **"HDIB1M–Handwritten Document Image Binarization 1 Million Dataset"**. In *arXiv preprint arXiv:2101.11674* (2021)

[Said et al. 2000] SAID, H.E.S. ; TAN, T.N. ; BAKER, K.D.: **"Personal identification based on handwriting"**. In *Pattern Recognition* 33 (2000), number 1, pages 149 – 160. – ISSN 0031-3203

[Schomaker and Vuurpijl 2000] SCHOMAKER, L. ; VUURPIJL, L.: **"Forensic writer identification: A benchmark data set and a comparison of two systems [internal report for the Netherlands Forensic Institute]"** / Nijmegen: NICI. 2000. – Research Report

[Schomaker 1998] SCHOMAKER, Lambert: **"From handwriting analysis to pen-computer applications"**. In *Electronics & Communication Engineering Journal* 10 (1998), number 3, pages 93–102

[Schomaker and Bulacu 2004] SCHOMAKER, Lambert ; BULACU, Marius: **"Automatic writer identification using connected-component contours and edge-based features of uppercase western script"**. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (2004), number 6, pages 787–798

[Schomaker et al. 2004] SCHOMAKER, Lambert ; BULACU, Marius ; FRANKE, Katrin: **"Automatic writer identification using fragmented connected-component contours"**. In *Ninth International Workshop on Frontiers in Handwriting Recognition* IEEE (event), 2004, pages 185–190

[Seuret et al. 2020] SEURET, Mathias ; NICOLAOU, Anguelos ; MAIER, Andreas ; CHRISTLEIN, Vincent ; STUTZMANN, Dominique: **"ICFHR 2020 Competition on Image Retrieval for Historical Handwritten Fragments"**. In *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)* IEEE (event), 2020, pages 216–221

[Shahabi and Rahmati 2006] SHAHABI, F ; RAHMATI, M: **"Comparison of Gabor-based features for writer identification of Farsi/Arabic handwriting"**. In *Tenth International Workshop on Frontiers in Handwriting Recognition* Suvisoft (event), 2006

[Siddiqi and Vincent 2010] SIDDIQI, Imran ; VINCENT, Nicole: **"Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features"**. In *Pattern Recognition* 43 (2010), number 11, pages 3853–3865

[Simonyan and Zisserman 2014] SIMONYAN, Karen ; ZISSERMAN, Andrew: *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014

[Singh et al. 2018] SINGH, Priyanka ; ROY, Partha P. ; RAMAN, Balasubramanian: **"Writer identification using texture features: A comparative study"**. In *Computers & Electrical Engineering* 71 (2018), pages 1–12

[Slimane and Märgner 2014] SLIMANE, F. ; MÄRGNER, V.: **"A New Text-Independent GMM Writer Identification System Applied to Arabic Handwriting"**. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, Sept 2014, pages 708–713. – ISSN 2167-6445. DOI: 10.1109/ICFHR.2014.124

[Song et al. 2017] SONG, Tiecheng ; LI, Hongliang ; MENG, Fanman ; WU, Qingbo ; CAI, Jianfei: **"Letrist: locally encoded transform feature histogram for rotation-invariant texture classification"**. In *IEEE Transactions on circuits and systems for video technology* 28 (2017), number 7, pages 1565–1579

[Song et al. 2015] SONG, Tiecheng ; LI, Hongliang ; MENG, Fanman ; WU, Qingbo ; LUO, Bing: **"Exploring space–frequency co-occurrences via local quantized pat-**

terns for texture representation". In *Pattern Recognition* 48 (2015), number 8, pages 2621–2632

[Srihari et al. 2008] SRIHARI, Sargur ; HUANG, Chen ; SRINIVASAN, Harish: **"On the discriminability of the handwriting of twins"**. In *Journal of Forensic Sciences* 53 (2008), number 2, pages 430–446

[Srihari et al. 2012] SRIHARI, Sargur N. ; CHEN, Gang ; XU, Zhen ; HANSON, Lisa: **"Studies in Individuality: Can Students, Teachers and Schools Be Determined from Children's Handwriting?"**. In *Computational Forensics*. Springer, 2012, pages 20–30

[Srihari et al. 2016] SRIHARI, Sargur N. ; MENG, Lu ; HANSON, Lisa: **"Development of individuality in children's handwriting"**. In *Journal of forensic sciences* 61 (2016), number 5, pages 1292–1300

[Tan and Triggs 2010] TAN, X. ; TRIGGS, B.: **"Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions"**. In *IEEE Transactions on Image Processing* 19 (2010), June, number 6, pages 1635–1650. – ISSN 1057-7149

[Timofte and Van Gool 2012] TIMOFTE, Radu ; VAN GOOL, Luc: **"A Training-free Classification Framework for Textures, Writers, and Materials."**. In *BMVC* Volume 13, 2012, pages 14

[Ünlü et al. 2006] ÜNLÜ, Atilla ; BRAUSE, Rüdiger ; KRAKOW, Karsten: **"Handwriting analysis for diagnosis and prognosis of parkinson's disease"**. In *International Symposium on Biological and Medical Data Analysis* Springer (event), 2006, pages 441–450

[Van Der Maaten and Postma 2005] VAN DER MAATEN, Laurens ; POSTMA, Eric O.: **"Improving automatic writer identification."**. In *BNAIC*, 2005, pages 260–266

[Van Galen et al. 1993] VAN GALEN, Gerard P. ; PORTIER, Stanley J. ; SMITS-ENGELSMAN, Bouwien C. ; SCHOMAKER, Lambert R.: **"Neuromotor noise and poor handwriting in children"**. In *Acta Psychologica* 82 (1993), number 1-3, pages 161–178

[Varma and Zisserman 2009] VARMA, Manik ; ZISSERMAN, Andrew: **"A statistical approach to material classification using image patch exemplars"**. In *IEEE transactions on pattern analysis and machine intelligence* 31 (2009), number 11, pages 2032–2047

[Wu et al. 2014] WU, X. ; TANG, Y. ; BU, W.: **"Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform"**. In *IEEE Transactions*

on Information Forensics and Security 9 (2014), number 3, pages 526–536. DOI: 10.1109/TIFS.2014.2301274

[Xing and Qiao 2016] XING, Linjie ; QIAO, Yu: **"Deepwriter: A multi-stream deep CNN for text-independent writer identification"**. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)* IEEE (event), 2016, pages 584–589

[Xu et al. 2010] XU, Minqiang ; ZHOU, Xi ; LI, Zhen ; DAI, Beiqian ; HUANG, Thomas S.: **"Extended hierarchical Gaussianization for scene classification"**. In *2010 IEEE International Conference on Image Processing* IEEE (event), 2010, pages 1837–1840

[Zhang et al. 2013] ZHANG, Jun ; ZHAO, Heng ; LIANG, Jimin: **"Continuous rotation invariant local descriptors for texton dictionary-based texture classification"**. In *Computer Vision and Image Understanding* 117 (2013), number 1, pages 56–75

List of Figures

List of Tables

A

PUBLICATIONS

A.1/ JOURNAL ARTICLES

1. Abderrazak Chahi, Issam El khadiri, Youssef El merabet, Yassine Ruichek and Raja Touahni, **Block wise local binary count for off-line text-independent writer identification**, In *Expert Systems with Applications Journal, Elsevier B.V.*, Volume 93, 2018, Pages 1-14.

2. Abderrazak Chahi, Youssef EL merebat, Yassine Ruichek and Raja Touahni, **An effective and conceptually simple feature representation for off-line text-independent writer identification**, In *Expert Systems with Applications Journal, Elsevier B.V.*, Volume 123, 2019, Pages 357-376.

3. Abderrazak Chahi, Youssef EL merebat, Yassine Ruichek and Raja Touahni, **Cross multi-scale locally encoded gradient patterns for off-line text-independent writer identification**, *Engineering Applications of Artificial Intelligence Journal, Elsevier B.V.*, Volume 89, 2020, Pages 103459.

4. Abderrazak Chahi, Youssef El merebat, Yassine Ruichek and Raja Touahni, **Local gradient full-scale transform patterns based off-line text-independent writer identification**, *Applied Soft Computing Journal, Elsevier B.V.* Volume 92, 2020, Pages 106277.

5. Abderrazak Chahi, Youssef El merebat, Yassine Ruichek and Raja Touahni, **WriterINet: a multi-path deep CNN for off-line text-independent writer identification (In Peer Review)**, *Expert Systems with Applications Journal, Elsevier B.V.*

6. Abderrazak Chahi, Youssef El merebat, Yassine Ruichek and Raja Touahni, **An effective DeepWINet CNN model for off-line text-independent writer identification (In Peer Review)**, *Applied Soft Computing Journal, Elsevier B.V.*

A.2/ CONFERENCES

1. Abderrazak Chahi, Issam El-Khadiri, Youssef El merabet, Yassine Ruichek and Raja Touahni, **Effective feature descriptor-based new framework for off-line text-independent writer identification**, In *The third International Conference on Intelligent Systems and Computer Vision, IEEE ISCV2018*, Fes Maroc.

2. Abderrazak Chahi, Youssef El merabet, Yassine Ruichek, Raja Touahni, **Hand-crafted descriptors-based new framework for off-line text-independent writer identification**, In the *International Conference on Soft Computing and Pattern Recognition, SoCPaR 2018, Springer Cham.*, Portugal, Porto, Cham, 2018.

3. Abderrazak Chahi, Youssef El merabet, Yassine Ruichek and Raja Touahni, **Off-line Text-independent Writer Identification Using Local Convex Micro- Structure Patterns**. Proceedings of the *New Challenges in Data Sciences: Acts of the Second Conference of the Moroccan Classification Society, ACM*, Maroc 2019.

Document generated with LaTeX and:
the LaTeX style for PhD Thesis created by S. Galland — https://github.com/gallandarakhneorg/tex-templates
the tex-upmethodology package suite — http://www.arakhne.org/tex-upmethodology/

école doctorale **sciences pour l'ingénieur et microtechniques**

Title: Automatic Off-line Text-independent Writer Identification from Handwriting

Keywords: Feature extraction, Writer identification, Deep learning, Convolutional neural network, Handwriting analysis

Abstract:

Handwriting-based writer identification has experienced a resurgence in recent years and continues to attract a great deal of interest and attention in the field of biometrics and pattern recognition. It is a challenging task considering the large within-writer and between-writer style variability. The automatic offline writer identification systems consider handwriting as scanned image containing certain recurring patterns that need to be analyzed. The motivation for this work stems from the need to improve behavioral biometric tasks that have been mainly used for writer identification to enhance security and forensic applications in today's world. The interest is to develop near real-time, effective, and robust approaches for writer identification by leveraging theoretical and technical advances in image analysis and artificial intelligence. This dissertation contributes to the numerous challenges encountered in all the main steps of an automatic system for offline writer identification, including image pre-processing and segmentation, feature extraction, and classification methods. Our first contribution investigates writer identification based on texture features. We propose four texture-based approaches to improve the task of writer identification: (1) The first approach, namely the Block Wise Local Binary Count (BWLBC)-based system, characterizes the variability of writing style within small blocks by capturing the pixels' distribution corresponding to writing ink in binary components ; (2) In the second approach, the Local Binary Patterns (LBP), Local Ternary Patterns (LTP), and Local Phase Quantization (LPQ) hand-crafted descriptors are applied to small regions of interest in the writing, called zones, to extract related texture features. They are performed in an efficient way using a new learning framework ; (3) The task of writer identification is improved thanks to a well-defined approach based on the Cross multi-scale Locally encoded Gradient Patterns (CLGP) descriptor to better represent salient local writing structures. It extracts transform features from connected components and encodes the obtained texture codes in multiple scales over the Histograms of Oriented Gradients (HOG) ; (4) The fourth approach computes local intensity gradients of the writing within non-overlapping blocks using the Local gradient full- Scale Transform Patterns (LSTP) method. This feature gives the overall system the ability to extract more relevant information to characterize the writing better. Convolutional Neural Networks (CNN) are also investigated to further improve the identification performance. Two computationally efficient and high-quality deep CNN-based approaches named DeepWINet and WriterINet are proposed. Extensive experiments are conducted on ten challenging handwritten benchmarks in different languages (English, Arabic, French, German, Chinese, Dutch, Greek, and hybrid). All the proposed approaches achieve competitive, or the highest SOTA performance in the benchmarks studied. We also participated in the ICFHR2020 competition to award the best approach for image retrieval for historical handwritten fragments. We proposed an effective deep learning-based approach based on multi-path CNN streams trained with different input data. The overall approach achieved excellent results and won first place in one of two tracks of the contest.

UBFC UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

Université Bourgogne Franche-Comté
32, avenue de l'Observatoire
25000 Besançon, France

école doctorale **sciences pour l'ingénieur et microtechniques**

Titre : Identification Automatique des Scripteurs à partir de l'écriture manuscrite par apprentissage machine et analyse d'image

Mots-clés : Extraction de caractéristiques, Identification des scripteurs, Apprentissage profond, Réseau neuronal convolutif, Analyse de l'écriture manuscrite

Résumé :

L'identification des scripteurs à partir de l'écriture manuscrite constitue un domaine de recherche en pleine expansion. Elle est devenue une thématique de recherche importante avec de nombreux verrous scientifiques/techniques et challenges/potentiels applicatifs, avec un besoin croissant de développement de systèmes biométriques destinés à de nombreuses applications sécuritaires. Les systèmes d'identification automatique des scripteurs (hors ligne) reposent sur des informations statiques. Les représentations numériques se présentent généralement sous la forme d'une image d'écriture manuscrite contenant des motifs récurrents à analyser et à caractériser. L'objectif ici est de développer des approches efficaces et robustes pour l'identification des scripteurs en tirant profit des avancées théoriques et techniques de l'analyse d'images et de l'intelligence artificielle. Cette thèse de doctorat aborde toutes les étapes essentielles d'un processus automatique de l'identification des scripteurs, y compris la phase de prétraitement, normalisation et segmentation, l'extraction des primitives, et l'étape de classification des scripteurs. Notre première contribution étudie l'identification des scripteurs en fonction des caractéristiques de texture. Nous proposons quatre approches basées sur la texture pour améliorer l'identification des documents de test : (1) le premier système proposé est basé sur la méthode Block Wise Local Binary Count (BWLBC). Il s'agit d'un descripteur d'image local pour caractériser la variabilité du style d'écriture en petits blocs en capturant la distribution des pixels correspondant à l'écriture dans des composantes connexes binaires ; (2) dans la seconde approche, les descripteurs de texture Local Binary Patterns (LBP), Local Ternary Patterns (LTP), and Local Phase Quantization (LPQ) sont appliqués sur des petites régions d'intérêt de l'écriture appelées zones, pour extraire les caractéristiques de texture associées. Les trois descripteurs sont utilisés efficacement dans un nouveau processus d'identification des scripteurs ; (3) la tâche de la caractérisation de l'écriture est améliorée par une approche basée sur le descripteur Cross multi-scale Locally encoded Gradient Patterns (CLGP). La méthode CLGP proposée caractérise les structures locales saillantes des composantes connexes de l'écriture. Elle encode la transformation de texture obtenue à plusieurs échelles en utilisant l'opérateur Histograms of Oriented Gradients (HOG) ; (4) la quatrième approche utilise la méthode LSTP (Local gradient full-scale Transform Patterns) pour calculer des histogrammes locaux de l'orientation du gradient sur des blocs non superposés de l'écriture. Cette caractéristique permet d'obtenir de meilleures performances en capturant des informations plus pertinentes sur la variabilité du style d'écriture. Les réseaux de neurones convolutifs (CNN) sont également étudiés pour améliorer l'identification des scripteurs. Deux approches efficaces basées sur les CNNs, appelées DeepWINet et WriterINet, sont proposées. Des expériences approfondies sont menées sur 10 bases de données manuscrites dans différentes langues (Anglais, Arabe, Français, Allemand, Chinois, Néerlandais, Grec et hybride). Toutes les approches proposées atteignent des performances compétitives ou supérieures à celles de la littérature pour les benchmarks étudiés. Nous avons également participé au concours ICFHR2020 pour récompenser la meilleure approche de « la récupération des fragments manuscrits historiques sur la base de l'identification des scripteurs ». Nous avons proposé une approche basée sur l'apprentissage profond en utilisant des multi-flux CNN entrainés avec différentes données d'entrée. L'approche globale a obtenu d'excellents résultats et a remporté la première place dans l'une des deux catégories de la compétition.

UBFC UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

Université Bourgogne Franche-Comté
32, avenue de l'Observatoire
25000 Besançon, France