sPeople's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Kasdi Merbah Ouargla

Faculty of New Information and Communication Technologies

Department of Computer Science and Information Technology

Arabic calligraphy style identification

Dissertation Submitted to the Department of Computer Science and Information Technology in Candidacy for the Degree of "Doctor" 3rd Cycle LMD in Computer Science.

Publicly defended on: 05/28/2022

Presented by:

Zineb Kaoudja

Supervised by:

Prof. Kerfi Mohammed Lamine

Co- supervised by:

Dr. Belal Khaldi

Jury Members

Cheriet Abdelhakim	President	MCA, Ouargla University
Mohamed Lamine Kherfi	Supervisor	Pr, Ouargla University
Belal Khaldi	Co-Supervisor	MCA, Ouargla University
Oussam Aiadi	Examiner	MCA, Ouargla University
Guessoum Ahmed	Examiner	Pr, USTHB
Moussaoui Abdelouahab	Examiner	Pr, Université de Setif 1

Academic Year: 2021 -2022

Dedication

I dedicate this work especially to the world dearest persons to me, who inspired me and gave me courage and hope,

to my lovely ***Mom*** *and* ***Dad.***

to My love, my husband, ***Bachir.***

to My Honey, my sweetheart, my daughter ***Saja Meriem.***

With great pleasure and joy:

To my dear brothers ***Youcef, Hichem, Abdelhak and Amine :)***

To My dearest sisters ***Aya and Amira***

To all my friends (specially ***Hadjer, Samar,and Nour Elhouda)***

To all my ***uncles*** *and my* ***aunts.***

To all those who were giving me any kind of support.

Acknowledgments

First of all, I would like to acknowledge the assistance of my supervisor **Pr. M. L. Kherfi** for his patience, support, and guidance throughout the course of this research study. I am deeply indebted to him also for his prompt and constructive feedback to accomplish this dissertation.

Furthermore, I would like to express my sincere appreciation to my co-supervisor **Dr. Bilal. KHALDI** for his guidance that was so helpful, I would like to thank him for the support that he gave me throughout the stiff times, and his patience. **GOD** bless him.

Additionally, I would like to express my deep gratitude for the generous support of the chief of our doctoral program, **Dr. LAALLAM Fatima Zohra.**

I would also like to extend my special thanks to the honorable members of the jury who accepted to examine and evaluate this study namely: **Dr.Cheriet Abdelhakim** from the University of Kasdi Merbah Ouargla, **Dr.Oussama Aiadi** from the University of Kasdi Merbah Ouargla; **Pr. Guessoum Ahmed** from the University of USTHB and **Pr. Moussaoui Abdelouahab** from the University of Setif 1.

I am particularly thankful for the contribution of my dearest colleague **AICHA. KORICHI.** Who, made this study feasible through her cooperation and her helpful advices.

I would like to thank with gratitude **Dr. KORICHI. Meriem** for her support that means a lot to me. Additionally, special thanks go to **Mr. EUCHI. SALAH** for his great support.

Abstract

Calligraphy is the most highly regarded and most fundamental element of Islamic art; and its strong relation to Islamic history. Defining the style could help to determine the period or the region where the text was written. Calligraphy text presents a group of challenges: letter shape is context-sensitive, orthography is very complex, text calligraphy is very related to the used style, and the calligrapher personal touch. This dissertation aims at contributing to the current research in the field of Arabic Calligraphy Style Recognition (ACSR) by developing novel techniques to analyze and to improve the performance of ACSR systems, with a focus on handwritten Arabic Calligraphy texts.

The contribution of the present dissertation is fourfold: (1) we collected and made available a dataset of images for Arabic handwritten calligraphy containing 1685 text (line\ sentence). This dataset is freely available for the scientific community; (2) we investigate the impact of combining classifiers on the task of ACSR; we use the Local Phase Quantization (LPQ) descriptor for Arabic calligraphy feature extraction and three different classifiers namely a Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and a Multi-Layer Perceptron (MLP) for style identification; and we compare between four different classifier combination techniques. The observed Arabic calligraphy style recognition rates are in the range from 93.7 % to 94.6 % for the individual classifier and respectively from 93.8 % to 96.5% for classifier decision combination. (3) We study the effect of using different texture descriptors for feature extraction, separately. For text style recognition, we use five different classifiers. with an accuracy rate in the range from 60.4% to 89.8% for RF classifier and respectively from 87.3% to 94.7% for SVM classifier. We find that the pattern-based descriptor with SVM machine learning is the best combination with classification rate 94.7%. (4) we introduce a new descriptor inspired by the Arabic Calligraphy styles: we find that each style has its distinguishing characteristics, so we propose a new computational method to extract these Arabic Calligraphy characteristics, and we use SVM machine learning with a classifier combination technique for text style identification. The mean recognition rate was from 96.8% to 97.8%.

Key Words: Document Analysis Systems (DAS), Arabic calligraphy style, Feature extraction, Machine learning, Arabic calligraphy style recognition.

Résumé

Le texte de calligraphie présente un groupe de défis : la forme de la lettre est contextuelle, l'orthographe est très complexe, la calligraphie de texte est très liée au style utilisé et le personnel du calligraphe est touché. Cette thèse vise à contribuer aux recherches actuelles dans le domaine de la reconnaissance du style de calligraphie arabe (ACSR) en développant de nouvelles techniques pour analyser le comportement et améliorer les performances des systèmes ACSR, en mettant l'accent sur les textes de calligraphie arabe manuscrits.

La contribution de la présente thèse est quadruple : (1) nous avons collecté et mis à disposition un ensemble de données d'images pour la calligraphie manuscrite arabe contenant 1685 texte (ligne). Cet ensemble de données est disponible gratuitement pour la communauté scientifique, (2) nous étudions l'impact de la combinaison de classificateurs sur la tâche de l'ACSR, nous utilisons le descripteur de quantification de phase locale (LPQ) pour l'extraction de caractéristiques de calligraphie arabe et trois classificateurs différents, à savoir Support Vector Machine (SVM), K-Nearest Neighbor (KNN) et Multi-Layer Perceptron (MLP) pour l'identification de style, nous comparons quatre techniques différentes de combinaison de classificateurs. Les taux de reconnaissance du style de calligraphie arabe observés sont compris entre 93,7 % et 94,6 % pour le classificateur individuel et respectivement entre 93,8 % et 96,5 % pour la combinaison de décision de classificateur. (3) nous étudions l'effet de l'utilisation de différents descripteurs de texture pour l'extraction de caractéristiques, séparément. Pour la reconnaissance de style de texte, nous utilisons cinq classificateurs différents. Avec un taux de précision compris entre 60,4 % et 89,8 % pour le classificateur RF et respectivement de 87,3 % à 94,7 % pour le classificateur SVM. Nous constatons que le descripteur basé sur des modèles avec l'apprentissage automatique SVM est la meilleure combinaison avec un taux de classification de 94,7 %. (4) nous introduisons un nouveau descripteur inspiré des styles de calligraphie arabe, nous constatons que chaque style a ses caractéristiques distinctives, nous proposons donc une nouvelle méthode de calcul pour extraire ces caractéristiques de la calligraphie arabe, et nous utilisons l'apprentissage automatique SVM avec une technique de combinaison de classificateurs pour l'identification du style de texte. Le taux de reconnaissance moyen était de 96,8 % à 97,8 %.

Mots clés :

Systèmes d'analyse de documents (DAS), style de calligraphie arabe, extraction de caractéristiques, apprentissage automatique, reconnaissance de style de calligraphie arabe.

ملخص

الخط هو العنصر الأكثر شهرة والأكثر أساسية في الفن الإسلامي. وعلاقته القوية بالتاريخ الإسلامي. يمكن أن يساعد تحديد النمط في تحديد الفترة أو المنطقة التي تمت كتابة النص فيها. يقدم الخط المكتوب باليد مجموعة من التحديات منها: كون شكل الحرف حساس للسياق ، والتهجئة معقدة للغاية ، والخط االمكتوب باليد مرتبط جدًا بالنمط المستخدم ، واللمسة الشخصية للخطاط. تهدف هذه الرسالة إلى المساهمة في مجال التعرف على أسلوب الخط العربي من خلال تطوير تقنيات جديدة لتحليل الانماط وتحسين أداء أنظمة التعرف على الخطوط العربية، مع التركيز على نصوص الخط العربي المكتوبة بخط اليد.

مساهمة الرسالة الحالية تتمثل في أربعة: (1) قمنا بجمع وتوفير مجموعة من الصور للخط العربي المكتوب بخط اليد تحتوي على 1685 صورة (سطرًا). مجموعة البيانات هذه متاحة مجانًا للمجتمع العلمي. (2) درسنا تأثير الجمع بين المصنفات في مهمة التعرف على الخط العربي، وباستخدام واصف تكميم المرحلة المحلية (LPQ) لاستخراج ميزة الخط العربي وثلاثة مصنفات مختلفة وهي آلة دعم المتجهات (SVM) و K-Nearest Neighbor (KNN) و Multi-Layer Perceptron (MLP) ، نقارن بين أربعة تقنيات مختلفة لتركيبة المصنفات. تتراوح معدلات التعرف على نمط الخط العربي من 93.7٪ إلى 94.6٪ للمصنف الفردي وعلى التوالي من 93.8٪ إلى 96.5٪ لمجموعة قرارات المصنفات. (3) نقوم بدراسة تأثير استخدام واصفات نسيج مختلفة لاستخراج الميزات بشكل منفصل. للتعرف على نمط النص ، نستخدم خمسة مصنفات مختلفة. بمعدل دقة في النطاق من 60.4٪ إلى 89.8٪ لمصنف RF ومن 87.3٪ إلى 94.7٪ على التوالي لمصنف SVM. وجدنا أن الواصف القائم على النمط مع التعلم الآلي SVM هو أفضل مزيج مع معدل التصنيف 94.7٪. (4) نقدم واصفًا جديدًا مستوحى من أنماط الخط العربي ، وجدنا أن كل نمط له خصائصه المميزة ، لذلك نقترح طريقة حسابية جديدة لاستخراج هذه الخصائص ونستخدم التعلم الآلي SVM مع تقنية جمع المصنفات من أجل تحديد نمط النص. كان متوسط معدل الاعتراف من 96.8٪ إلى 97.8٪.

الكلمات المفتاحية:

أنظمة تحليل الوثائق (DAS) ، أنماط الخط العربي ، استخلاص الميزات ، التعلم الآلي ، التعرف على أسلوب الخط العربي.

Table of Contents

List of Figures

List of Tables

List of Publication

Published Journal Papers

- Kaoudja, Zineb, Mohammed Lamine Kherfi, and Belal Khaldi. "A New Computational Method for Arabic Calligraphy Style Representation and Classification." Applied Sciences-MDPi 11, no. 11 (2021): 4852. **Impact factor: 2.68**

Published Conference Papers

- Kaoudja, Z., Kherfi, M. L., & Khaldi, B. (2019, June). An efficient multiple-classifier system for Arabic calligraphy style recognition. In 4th International Conference on Networking and Advanced Systems (ICNAS) (pp. 1-5). Annaba IEEE.

- Kaoudja, Z., Khaldi, B., & Kherfi, M. L. (2020, May). Arabic Artistic Script Style Identification Using Texture Descriptors. In the 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP) (pp. 113-118). El Oued IEEE.

.

Chapter 1

Introduction

Chapter 1 : Introduction

Contents

1.1 Introduction

In computer vision, an image is taken as input, aiming to understand its contents. For that, Computer vision uses image processing algorithms to solve some of its tasks. Image processing is a field to perform some procedures on an image, to get an improved image, or to extract some useful information. The feature is a concept that is used to describe one kind of information that could be in an image like lines, corners, shapes, colors, textures [1] [2], etc. Feature extraction is a key step that is significantly addressed in many image classification systems. Moreover, machine learning is also considered one of the main steps for image classification systems [3]. These two fields (image processing and machine learning) combined successfully in different areas such as robotics, pattern recognition [4], medical diagnosis, handwriting recognition [5], and many other applications.

One of the main computer vision applications is Document Analysis System (DAS). This type of application aims to analyze document images and reproduce the text they contain in printed format. DAS is divided according to the purpose for which it was established, like application for text recognition called Optical Character Recognition (OCR), application for text language recognition which called Optical Script Recognition (OSR), and Optical Font Recognition (OFR). The last one is the category to which our study belongs, concentering on the Arabic language.

Many works have been conducted for other languages. However, the proposed solutions are not efficient and applicable for the Arabic text, due to the following reasons:

- The nature of The Arabic letters which are so smooth, interconnected, and have a unique characteristic in their form (figure 1.1).

- The possibility of overlapping the letters above one another. This characterization could not be found in other language's writing (figure 1.1).

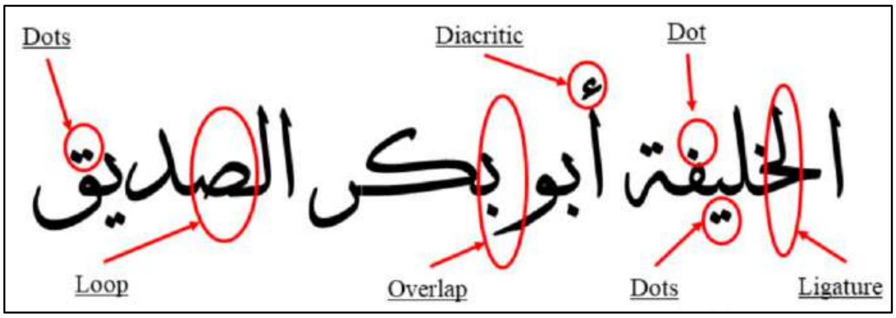


Figure 1.1 Characteristics of the Arabic text.

- Arabic has 28 letters, each one has a single/isolated form and a connected form (at the beginning, center, and end) (figure 1.2).

final	medial	initial	isolated
ـط	ـطـ	طـ	ط
ـف	ـفـ	فـ	ف
ـق	ـقـ	قـ	ق
ـل	ـلـ	لـ	ل

Figure 1.2 Some of the Arabic letters in different situations: isolated or connected, at the beginning, in the middle or at the end of the words.

- The Arabic letter shape can also be affected by the writing style.
- The Arabic calligraphy is affected by the writer's writing style,
- The purpose of writing, whether it is for decoration or other purposes?

Considering the pre-mentioned Arabic characteristics, the handwritten Arabic Calligraphy text recognition has not yet reached research maturity. It remains an active research topic and presents major challenges because of the multiple problems that face the researchers [6]. These problems are:

- The special challenges in the Arabic script.
- The techniques developed for other languages cannot be applied to Arabic writing.
- The absence of good support in terms of conferences, books, journals, and funding, and the lack of interaction between researchers in this research area.
- The lack of general resources like Arabic calligraphy text dataset.

1.2 Digital Document Analysis

Document Analysis (DA) is one of the computer vision applications, which aims to analyze digital documents containing texts of different types such as digits, or alphabet. Recurrently, Document Analysis is a discipline that combines image processing and machine learning techniques to process and extract information from documents from different sources. DA systems are divided mainly into two categories; bottom-up and top-down. First, bottom-up approaches iteratively analyze a document based on the pixel data. These types of approaches usually analyze a document starting with the small connected components of black and white, then these components are assembled into words, then into text lines, and finally into text blocks/paragraphs [7]. Second, there are top-down approaches that endeavor to iteratively separate the document content into columns and blocks based on white space and geometric information [8]. DA systems attempt to extract the layout information from a scanned document page and then reuse them to generate a new visually similar document with new content.

1.3 Optical Font Recognition

Optical Font Recognition systems (OFR) is one of the digital Documents Analysis Systems; it aims to define the image text style, whatever the text nature, be it printed or handwritten. The main goal behind these systems is to make the machine capable to analyze and define the text's style by itself. The existing literature suggests that OFR systems for Arabic calligraphy handwritten texts are fewer compared to the importance of the Arabic language, and the huge number of digitally scanned Arabic documents in different styles. Treating texts with multiple fonts is a challenge in Optical Character Recognition (OCR): text with different fonts makes the OCR system handle large variations of character appearance[9]. If the text is labeled with a font type, then an expert OCR system for that font can potentially achieve higher recognition rates than an OCR system trained on many fonts.

1.4 Arabic calligraphy: Definition and History

Arabic is the fourth most spoken language in the world. Also, Arabic is the first language of more than 200 million people across the world. As with any language, it has its grammar, spelling, and punctuation rules, its slang and idioms, and its pronunciation. Arabic characteristics make it distinctive from other languages, however, including the number of varieties and its written form, Arabic handwritten text is classified into two main parts, artistic writing (called calligraphy) and non-artistic (the normal handwriting) scripts. Arabic calligraphy (AC) is one of the most significant arts in the world. The AC was written by hand at first. The first calligraphy style was developed at the end of the 7th century. Over time, many other calligraphic styles developed in the Muslim and Arab world.

What you type by machine is different from handwritten calligraphy content. Fonts may provide alternatives and ligature formats and some stylized versions, but in essence, you can only be as creative as a font designed. In calligraphy, there are no restrictions when Letters are written by hand. Each version of this letter may look different. You can merge any letter with another letter and connect letters that are not adjacent to each other, so there is no flexibility in typing compared to manually writing letters.

Humans invented writing for documentation and communication, the same applies to the Arabs civilization. Arabic writing was originally a tool for communication, but with time, it began to be used in architecture and decoration. Its evolution into these major roles was a reflection of the early Muslims' need to avoid figures and pictorials [10]. This artistic writing is called calligraphy (which means, literally "beautiful writing").

While the Arabic tribes preferred to memorize texts and poetry, the first Muslims tried to document their holy book (Qur'an Kareem) using scripts. Over the course of their development, the Arabic scripts were created in different periods and locations of the growing Islamic Empire. There is also a close relationship between each Arabic script and its common usage throughout history.

Due to the influence of Islam, the Arabic alphabet is one of the most widespread writing systems in the world, found in large parts of Africa and Western and Central Asia, as well as in ethnic communities in East Asia, Europe, and the Americas. While originally used to write the Arabic language, the Arabic alphabet has been adopted by other groups to write their languages, such as Persian, Pashto, Urdu, and more.

the origin of the Arabic alphabet dates way back deeper in time. The Nabataeans, who established a kingdom in what is modern-day Jordan from the 2nd century BCE, were Arabs. They wrote with a highly cursive Aramaic-derived alphabet that would eventually evolve into the Arabic alphabet. The Nabataeans endured until the year 106 CE when the Romans conquered them, but Nabataean inscriptions continue to appear until the 4th century CE, coinciding with the first inscriptions in the Arabic alphabet (which is also found in Jordan).

1.5 Motivations

Arabic Calligraphy styles dates back many centuries ago, text style recognition is an essential part of Paleography science. Paleography is the study of ancient writing systems and the deciphering and dating of historical manuscripts. Arabic Calligraphy Style Recognition (ACSR) is a Document Analysis (DA) application, it is targeted to define the text style from text document images. ACSR is a necessary step for the following reasons:

- Document reprinting: defining the text style helps to read the text character, because characters have different shapes in each style. For that, ACSR considered as a key step for OCR application.
- Writer identification: As we already mentioned, calligraphy is considered as an art; calligraphy writers use their own touch to distinguishes them from the other calligraphers, this was done since the invention of the Arabic text writing. So ACSR helps to define the writer from his writing style.
- The epoch or time specification: Due to the historical connection of calligraphy with Arabic and Islamic civilizations, defining the text style in the historical document helps to define the time when the document has been written.
- Region/ Area where the style was used mostly: each calligraphy style was related mostly to a specific region where that it originally invented on, and where it was used the most, so, the same as previous, defining the text style in the historical documents helps to define the area where the text document written.

1.6 Contributions

The contributions of this research work are presented in this dissertation and are introduced in two folds: feature extraction and recognition approach.

- ✓ A detailed study about the Arabic script and Arabic calligraphy: a deep study both Arabic writing and calligraphy to understand the nature of the Arabic text, its specifics that distinguished it from the other languages. And, precisely, calligraphy text;
- ✓ A detailed study of the state of calligraphy text recognition: a study of the related works, to help us to determine our new solutions;
- ✓ Conception and validation of different kinds of feature extraction techniques: studying various feature extraction techniques, and the effect of using them for Arabic Calligraphy style Recognition;
- ✓ Experiments with text images from different resources: we create a new dataset using different resources, where our solutions established on this new dataset;
- ✓ Elaboration of a robust system for Arabic calligraphy style recognition based on:
    - classifier combination with LPQ texture descriptor,
    - texture descriptors and machine learning, and
    - a new machine learning-based tool and a new feature extraction technique;
- ✓ Extensive evaluation of our work with a comparison with existing systems: to determine the validity of our proposed models we compare them with other existed researches; and
- ✓ Design and collection of a large and public dataset of Arabic calligraphy images[1]: the newly created dataset is available for the research community.

[1] **NOTE:**
The web link to our Arabic Calligraphy +Database: https://drive.google.com/open?id=1dC7pwzT_RHL9B42H8-Nzf5Sant-86NV6

1.7 Arabic calligraphy dataset

The motivations for building this dataset are the following. First, at the time of starting the thesis, there was no public, large-scale, and comprehensive Arabic calligraphy text dataset that was freely available. Second, we wanted to study new dimensions in this thesis such as multi-font and multi-size aspects that are present, amongst others, in screen-rendered applications.

For these reasons, we have initiated the development of a large collection of images of Arabic calligraphy. This dataset contains 1685 text images, divided into 9 classes, each class represents a calligraphy style. We have used different tools and resources to group these image sets, which are digital camera, books, digital books and websites.

1.8 The 6 main Arabic calligraphy styles

Arabic calligraphy developed from two major styles: Kufic and Naskh. There are several variations of each, in addition to regionally specific styles. In the history of Arabic calligraphy, there are six main styles namely Kufi, Diwani, Farisi, Naskh, Rekaa, and Thuluth, each one has its history, characteristics, and rules.

A. Kufi

The Kufi script is the oldest calligraphic form that was developed by the end of the 7th century in Kufa, Iraq. It was the first script used to write the Qu’ran. It is angular with a thick extended stroke. Under the Kufi classification falls a great variety of scripts such as Ornamental Kufi, script Kufi, square Kufi, drawn Kufi, Maghribi, etc.

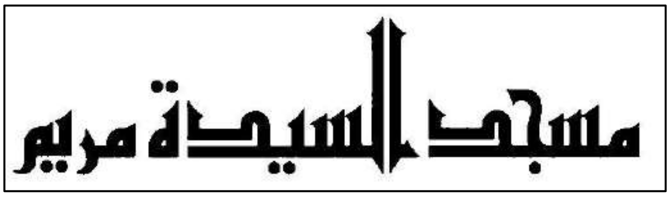


Figure 1.3 the Kufi style

B. Diwani

The Diwani script is a cursive script developed during the reign of the early Ottoman Turks. It reached its popularity in the 16th century. It is called Diwani which means ministerial. Due to its heavy stylization, it became the ideal script for writing court documents as it ensured confidentiality and prevented forgery. They made this script the official script of the Ottoman sultans. It had mysterious rules that none but the master calligraphers of the Ottoman sultan and their pupils knew about. This writing was used in the writing of all royal decrees, endowments, and resolutions. It is cursive, beautiful, and harmonious. The Diwani style is characterized by the rounded shape of letters. Words are written skewed, descend from right to left. The lines start thick then become thinner.

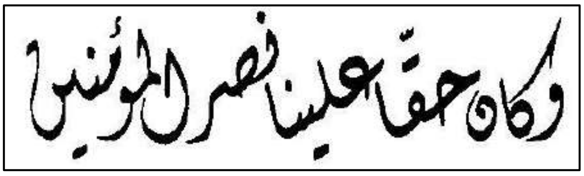


Figure 1.4 the Diwani style.

C. Farisi

The Fersian style appeared in Persia in the seventh century AH (13th century AD), called the style of suspension, a beautiful line characterized by letters accuracy and extension, and is characterized by ease and clarity and lack of complexity, Farisi or Nasta'liq is a cursive style originally devised to write the Persian language for literary and non-Qur'anic works. Farisi is thought to be a later development of the naskh and the earlier ta'liq script used in Iran.

Figure 1.5 the Farisi style.

D. Naskh

Naskh is one of the first scripts of Islamic calligraphy to develop. Naskh was standardized by Ibn Muqla as one of the six primary scripts of Islamic calligraphy in the 10th century CE". It is one of the simplest types of fonts for writing as well as for reading. It is suitable for learning and reading for different ages and stages. It is called 'NASKH' means the copy font because it is used for copying books. It is recognizable by its balance and its plain clear forms, its proportions are respected and measured by the dot system: 'Nizam al Nokat' or by the diameter system: 'Nizam al Da'irah'. Nowadays, this script is used primarily in print.

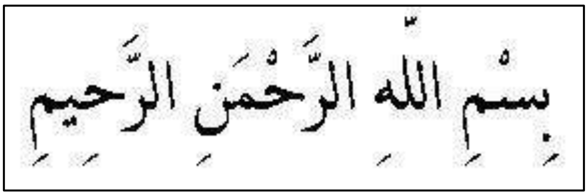


Figure 1.6 the Naskh style.

E. Rekaa

The word Rekaa means "a small sheet" which could be an indication of the medium on which it was created. It is easy to learn and easy to use. Mostly used without diacritics except for the Quran verses. Rekaa is the most common type of handwriting in the Arabic script. It is known for its clipped letters composed of short, straight lines and simple curves, as well as its straight and even lines of text. It was probably derived from the Thuluth and Naskh styles. The rekaa text is written on the baseline except for these letters (ج، ع، م، هـ) 'meem' 'jeem' 'haa' 'khaa' 'ain' where they go under the baseline at the end of the word.

Figure 1.7 the Rekaa style.

F. Thuluth

The Thuluth is one of the basic Arabic calligraphy styles. It is a script variety of Islamic calligraphy invented by Ibn Muqlah Shirazi. It is one of the most difficult pieces of writing to master. It is recognized by the slopes that appear on the one-third of each letter thus derives its name that means "a third" in Arabic. An alternative theory to its meaning is that the smallest width of the letter is one-third of the widest part. This script was invented in the 11th century, however; it went through an extensive evolution during the Ottoman Empire in the 15th century and onwards. It is an elegant, cursive script that has large, elongated, and elegant letters. Used in medieval times on mosque decorations. Various calligraphic styles evolved from Thuluth, with slight changes of form. It has a certain role, but it could be changed according to the calligrapher's needs. It has a big letter, which takes a big space above the baseline; the big curves and the overlap between them make big holes. The calligrapher fills in these holes with artistic diacritics.

Figure 1.8 the Thuluth style.

1.8.1 The extended styles from the 6 main styles

By and by, many styles have appeared, driven from the basic styles. Over 100 different styles of Arabic calligraphy have been identified, becoming ever more refined and elegant, some of these styles are still used and some others has disappeared. The most used style nowadays is defiantly the six basic style and three other styles that are driven from them which are Maghribi, Mohakik, Square-Kufi.

A. Maghribi

The Maghribi script was invented on the west side of North Africa in the Islamic Empire, it is driven from the Kufi style. Because of its decorative appearance, The Maghribi script is suitable for decorations as well as titles. The Maghribi script developed during the 10th century and is still used today in Spain and western North Africa, particularly in Morocco, Algeria, and Tunisia. The special form of its letters gives it a unique beauty and makes it easy to read, even in a long text. The Maghribi script is marked by descending lines written with very large bowls and by letters of a unified thickness.

Figure 1.9 the Maghribi style.

B. Mohakik

The Mohakik style is inspired by Thuluth calligraphy. The Arabic word Mohakik (مُحقَّق) means "consummate" or "clear". Often used to copy Quran verses, this majestic type of script was considered one of the most beautiful, as well as one of the most difficult to execute well. Mohakik characters are not as round as those of Thuluth. That means Mohakik and Thuluth calligraphy had the same characteristics, Except for the Thuluth, the calligrapher Exaggerate with overlapping and extending lines of letters. Letters in Mohakik style are longer than we can see under the baseline.

Figure 1.10 the Mohakik style.

C. Square-Kufi

Square Kufi is unique among the styles of Arabic calligraphy because it developed on the walls of buildings rather than on paper. It is executed in mosaic faience, decorative glazed facing tiles, or simple bricks, rather than with reed pens and ink. Square Kufi has only one strict rule: absolute evenness of full and empty spaces, plus the square angles and strict lines.

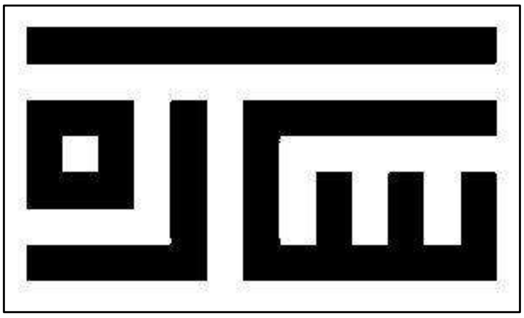


Figure 1.11 the Square-Kufi style.

1.9 Challenges of Automatic Arabic calligraphy style Recognition

The Arabic texts are so smooth, interconnected, and have a unique characteristic in their form. In addition, Arabic text has 28 letters, each one has a single/isolated form and a connected form (at the beginning, center, and end). In addition, the possibility of overlapping the letters above one another. This characterization could not be found in the other language’s writing.

Dealing with Images from different resources make us deal with diverse challenges like:

- Image quality: images with low resolution are made difficult to separate text from the background.
- Text quality: images for historical documents might contain damaged texts.
- Different text sizes.
- Different text and backgrounds colors.

For that, we try to create a solution that could deal with all these problems, using the Arabic calligraphy text characteristics, aiming to define the text style.

1.10 Thesis structure

The remaining parts of this thesis are structured as follows:

Chapter 2: In this part, we present an overview of the literature that relates to the work presented here. The listed works are all conducted in Arabic text style for both artistic and non-artistic text.

Chapter 3: This chapter attempts to present an automatic system for Arabic-calligraphy style classification, on the newly created Arabic calligraphy dataset. We introduce a system for Arabic calligraphy style recognition based on the combination of multi-classifier decisions. We use the LPQ at the image representation level for image feature extraction, whereas, at the decision level we combine three different classifiers, namely SVM, MLP and KNN. This work has been published in the 4th International Conference on Networking and Advanced Systems (ICNAS) indexed by IEEE.

Chapter 4: this chapter deals with Arabic Artistic Script Style Identification Using Texture Descriptors. We design and evaluate a system for Arabic artistic style identification using numerous texture descriptors. The system has been evaluated using five different classifiers. The Results indicated that texture descriptors appropriately fit the task and give good results for most of the styles. This work has been published in the 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP) indexed by IEEE.

Chapter 5: This is a new machine learning-based tool for Arabic calligraphy style recognition. We propose a new tool for Arabic calligraphy style recognition (ACSR). This chapter aims to identify Arabic calligraphy style (ACS) from images. To this end, we use the indices used by experts to distinguish different calligraphy styles. These indices were transformed into a computational feature, for each calligraphy style, a set of specific features were extracted. This work was published in the Applied Science journal of the MDPI publisher.

Chapter 6: The General Conclusion concludes this research and demonstrates how far we have progressed toward our goals, and the future objectives.

Chapter 2

State of the art

Chapter 2 : State of the art.

Contents

2.1 introduction

In this part, we present an overview of the literature that relates to the work presented here. The listed works are all conducted in Arabic text style for both artistic and non-artistic text. Arabic Font recognition is treated mainly with two different approaches. First is the Global approach, it refers to the image independent content approach. For that the whole image is taken as input; which means no need for image segmentation or text localization. Second, the local approach refers to the approach that is dependent on image content. In this type of approaches the text image segmented into different parts (background, text, word, letter…); which means the need to segment the image and localize the text.

2.2 Related works for Arabic Calligraphy texts (Artistic)

While there has been much research on processing Arabic Font, few researchers have considered Arabic calligraphy (Table 2.1). We could chalk it up to the lack of resources like public datasets and insufficiently related works, as mentioned previously in this thesis, we will discuss them in two groups; global approaches and local approaches.

Table 2.1 Arabic calligraphy style recognition related works

References	Script	Dataset	# Of Fonts
Batainah et al. (2012) [11]	Arabic JAWI	700 block images[2] (private)	6
Batainah et al. (2013) [12]	Arabic JAWI	700 block images (private)	7
Batainah et al. (2011) [13]	Arabic	14 document images (private)	Not indicated
Batainah et al. (2011) [14]	Arabic	100 images (private)	6
Talab et al.(2011) [15]	Arabic	700	7
Azmi et al. (2011) [16]	Arabic JAWI	100-isolated character image (private)	5

[2] block image= repeating the original image n*n time

Azmi et al. (2011) [17]	Arabic JAWI	1019- isolated character images (private)	4
Adam et al. (2017) [18]	Arabic	330- isolated character images (private)	6
Allaf et al.(2016) [19]	Arabic	267 sentence/word Images (private)	3
Elhmouz & al. (2020) [20]	Arabic	421 sentence/word Images (private)	3
M. Khayyat(2020) [21]	Arabic	2653 documents images (private)	6

2.2.1 Arabic calligraphy style recognition based on global approaches

Starting with global approaches for Arabic calligraphy recognition, Bataineh & Talab in all their works [11-15] attend to classify the Arabic calligraphy with a proposed statistical descriptor, named Edge Direction Matrix (EDMs), 22 moments were extracted from a 3*3 matrix that counts the edges pixels adjacency, passing through two steps EDM1 then EDM2. EDM1 for the pixels adjacency, then EDM2 counted from EDM1, which contains the most repeated adjacency relations. Finally, the 22 moments were extracted from both EDM1 and EDM2. The achieved accuracy rate was 96.65% in [11], 96.7% for [12], 92.8% for [13], 95.9% for [14] and 99.4% [15]. Another solution proposed by Elhmouz [20] used a common deep model (auto-encoder) for image feature extraction and a softmax layer for font type recognition with accuracy rate equals to 92.4%. overall, the prementioned solution is based on a general idea that could be utilized for other objects rather than calligraphy.

2.2.2 Arabic calligraphy style recognition based on local approaches

For other works of Arabic calligraphy recognition based on a local approach. Azmi, Adam & Allaf [16-19] attempt to extract features using the text/letter shape. For Azmi and Adam [16-18], the main weakness for both of their works is manually segmenting letters, where some styles are so arduous or could be almost impossible. In [16] they used images from inscribed stone, then manually extracted twenty letters. And in [17] they used the same dataset with more images. The accuracies rates for each style individually were variate from 12.9% to 69.6%. Allaf [19] also uses the text geometrical shape, the density of text above and under the baseline, number of diacritics up and down the text, text orientation, the position of the text baseline, and the ration of black and white for the whole image on a small dataset of 3 styles, with a classification error rate equal to 8.02%. However, we argue that previous works suffer from certain weaknesses. First, regarding the letter segmentation due to the complexity of the Arabic text and the overwrought of the process, besides, the small datasets used for the study. For that, we try to keep away from text segmentation and use the whole sentence/word for feature extraction.

2.3 Related works for Arabic Font texts (non-Artistic)

Arabic text font recognition has been extensively studied, unlike Arabic calligraphy, owing to the availability of a huge number of resources that help the researchers to build new solutions. In the following table (Table 2.2) a group of research papers fall under Arabic Font recognition. In this part too, we will discuss them in two parts local approaches and global approaches.

Table 2.2 Arabic Font recognition related works

References	Script	Dataset	# Of Fonts	# Of Styles
Kellal & al (2016) [22]	Arabic	5000 text images Arabic Printed Text Image - Database (APTID)	10	2
F. Slimane & al (2013) [23]	Arabic	20000-word images		

		Arabic Printed Text Image (APTI)		
C. Tensmeyer & al (2012) [24]	English	3000-line images Classification of Latin Medieval Manuscripts (CLaMM)	12	
	Arabic	115,068 Scanned pages King Fahd University Arabic Font Database (KAFD)	40	
H. Luqman & al (2014) [25]	Arabic	2,576,024 Scanned pages	40	4
M. Lutf. & al (2014) [26]	Arabic	(private)	10	
Abuhaiba (2005) [27]	Arabic	108,000-word images (private)	10	3
Abuhaiba (2003) [28]	Arabic	185,839-word images (Private)	3	4
Kallel et al (2017) [29]	Arabic	500 block images	10	
Ben Moussa & al(2006) [30]	Arabic	450 block images	10	
Ben Moussa & al (2010) [31]	Arabic	1000 block images	10	
Zaghden et al (2006)[32]	Arabic	2500 block images	10	
Chaker & al (2010) [33]	Arabic	computer-generated dataset (size not indicated)	10	
Zahedi& Eslami (2011) [34]	Farisi-Arabic	1400 block images	20	
Nicolaou & al (2014) [35]	Arabic	APTI dataset		
Mousa & al (2015) [36]	Arabic		15	
Pourasad & al (2012) [37]	Arabic – Farisi	500 Document image	25	
Bozkurt & al (2015) [38]	Arabic-Farisi	60 Documents	5	
	Arabic	ALPH-REGIM dataset	10	
Izakian & al (2008) [39]	Arabic- Farisi			
Sakr & al (2019) [40]	Arabic	20000 images	50	

Ahmed Kawther Hussein (2020) [41]	Arabic	2766 text line image	8	

2.3.1 Printed Arabic text style recognition based on global approaches

For font recognition solutions that are built on a global approach, Kellal [22] has proposed a solution with two phases; image representation with a discrete curvelet transform, and backpropagation neural network for font recognition. The study was conducted on the APTID/Multi-Font dataset in three different cases of high resolution, noisy, and blurred images. He reported accuracies for the three cases as 99.72% for high-resolution images, 99.70% for noisy images, and 99.59% for blurred images. In Slimane [23], a new font and size identification method was proposed for ultra-low-resolution images of Arabic printed word images using a stochastic approach for feature extraction and GMM for font and size recognition. The conveyed accuracies for font recognition were 94.5%, for size recognition 96.2%, and 91.9% for font & size recognition. for that APTI dataset was used. Tensmeyer & al [24] have used two different deep learning models (ResNet/AlexNet) to compare their performance for text font recognition of Arabic printed text (KAFD dataset) and English text (CLaMM). Luqman & al [25] have created a large dataset named (KAFD) for Arabic printed text with different fonts and styles (pages and lines), they found that using Log-Gabor filters for image representation gives the best recognition rates of around 98%. Abuhaiba [27] Extracted 48 features from 100 fully connected words, which means words without separation character in the middle. These features include horizontal projections, Walsh coefficients, invariant moments, and geometrical attributes. He supposed that these words have existed in any Arabic text, he used a decision tree for font recognition, with an accuracy rate of over 90.8%. Kallel [29] used the steerable Pyramid texture descriptor for image processing and used a Backpropagation Artificial Neural Network (ANN) as a classifier. Ben Moussa & al [30] used fractal multi-dimensions for Feature extraction and a KNN classifier, with an accuracy rate of 98%. Another work for Ben Moussa & al [31] is based on fractal multi-dimension for image representation and RBF Neural network for image classification, with an accuracy rate of 96%. Ahmed Kawther Hussein [41] has tested several texture descriptors for font representation using Extreme Learning Machine (ELM) and Fast Learning Machine (FLN) to compare their performance in Arabic font recognition. The global approach turned out not to be enough for Arabic Calligraphy where features are extracted from the whole image without taking into consideration the characteristic of AC. Also, the researchers used their own datasets (private), there is no possibility to do fair comparisons.

2.3.2 Printed Arabic text style recognition based on local approaches

With respect to researches that are based on local approaches. we start with Lutf [26]. This paper presents a method for Arabic font recognition based on only text diacritics with ring and central projection (CCRP) for image representation and normalized cross-correlation for style classification. Another work for Font recognition is based on templates that are built from words after the letter's segmentation process. For Chaker & al [33], their system is based on the posterior approach that performs font recognition of isolated characters. It is based on the dissimilarity index computed on the polygonal approximation of the character. Nicolaou & al [35] used Reduced-Oriented-LBP with the nearest neighbor classifier for Arabic font recognition. Mousa & al [36] proposed algorithm used a scale-invariant detector, a gradient-based descriptor, and k-means clustering to recognize the Arabic font in the text image. The mean recognition rate was 99.2–99.5%. Bozkurt & al [38] had extract features using complex wavelet transform and use support vector machines for classification. In [40] Font recognition is based on letter levels using the Euclidean distance between spatial descriptors and gradient value in each boundary point.

2.4 Conclusion

In this chapter, we have reviewed some of the previous approaches used in the field of Arabic text style recognition for both printed and handwriting, focusing on Arabic calligraphy style recognition.

The choice of one approach over another is related to certain constraints such as the input representation, the size of the available training set. Its application on off-line automatic Arabic calligraphy style recognition gave interesting results.

The next chapters cover the proposed solutions for Arabic calligraphy style recognition.

Chapter 3

An efficient multiple-classifier system for Arabic calligraphy style recognition

Chapter 3 : An efficient multiple-classifier system for Arabic calligraphy style recognition

Contents

In this chapter we present an automatic system for Arabic-calligraphy style classification[42]. We put forward and make publicly available a new Arabic calligraphy style dataset that comprises 1685 images categorized into 9 styles. Additionally, we introduce a system for Arabic calligraphy style recognition based on the combination of multi-classifier decisions. The proposed system employs LPQ at the image representation level, whereas, it combines three different classifiers, namely SVM, MLP, KNN, at the decision level.

Published in: 2019 International Conference on Networking and Advanced Systems (ICNAS)
Date of Conference: 26-27 June 2019
Date Added to IEEE Xplore: 22 August 2019
ISBN Information:
INSPEC Accession Number: 18940227
DOI: 10.1109/ICNAS.2019.8807829
Publisher: IEEE
Conference Location: Annaba, Algeria

An efficient multiple-classifier system for Arabic calligraphy style recognition

Abstract

Arabic calligraphy is one of the most important inheritances especially for Arabic and Islamic civilization. Classifying Arabic calligraphy manuscripts manually is a tedious and time-consuming task. This work endeavors to present an automatic system for Arabic-calligraphy style classification. The main issue one faces to fulfill such a task is the unavailability of data due to the lack of related works. Therefore, we put forward and make publicly available, a new Arabic calligraphy style dataset that comprises1685 images categorized into 9 styles. Additionally, we introduce a system for Arabic calligraphy style recognition based on the combination of multi-classifier decisions. The proposed system employs Local Phase Quantization (LPQ) at the image representation level, whereas, it combines three different classifiers, namely support vector machine (SVM), multi-layer perceptron (MLP), and K-nearest neighbor (KNN), at the decision level. The Experimental evaluation with different combination techniques has demonstrated the efficacy of the proposed method and promising results compared to other methods.

Keywords— Arabic calligraphy handwriting, optical font recognition (OFR), Local Phase Quantization, Classifier combination.

3.1 Introduction

Arabic calligraphy handwriting is an artistic way for text writing, it contains several styles each one has its features. These styles were born over the epochs, e.g., Kufic, Diwani, Naskh, Thuluth, Roqa'a, Maghribi, etc. The font type and style give important information about the corresponding document such as content, parts, history, origins, etc. Some of these styles are illustrated in Figure 3.1. To differentiate Arabic autistic styles, one must hold a certain level of expertise. However, due to the huge amount of available digital documents in different fonts, manual classification became an extremely hard and tedious task. Therefore, computer vision and machine learning techniques should be involved in order to alleviate such a burden. Optical Font Recognition (OFR) is an approach that is widely used for automatic digital document processing and font recognition.

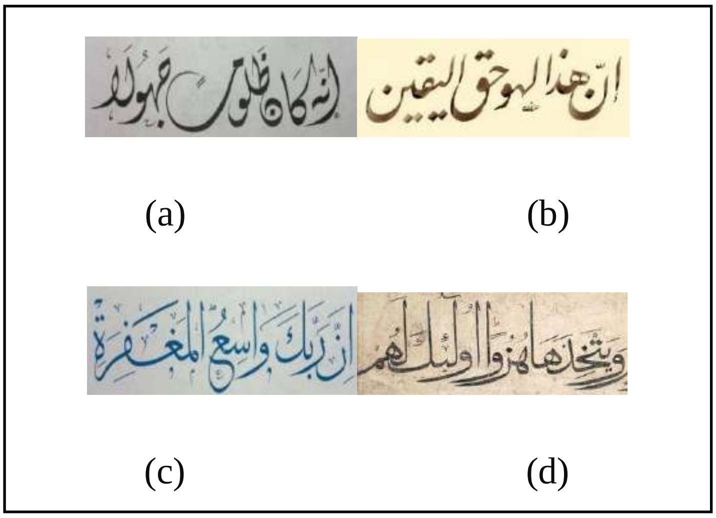


Figure 3.1 Illustrative examples of some Arabic calligraphy styles (a) Diwani (b) Parsi (c) Thuluth (d) Mohakek.

Owing to the lack of data and works on Arabic calligraphy handwriting, difficulties confront researchers working in such a field, which raises a serious challenge. Another problem when dealing with Arabic text is the complex shape of the Arabic letters, especially in artistic texts (Figure 3.1). The letters in Arabic have different shapes based on the appearance position (at first, middle, or end of the word) as is illustrated in Figure 3.2.

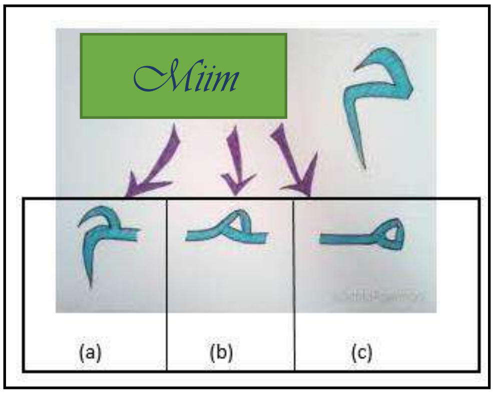


Figure 3.2 the Arabic letter "Miim" shape depends on its position (a) at the end (b) in the middle(c) at the beginning of the word.

Arabic handwriting style has a cursive shape, and its characteristics could be affected by the writer's style, or furthermore, the writer's emotions. All the aforementioned difficulties raise a serious challenge to propose a system for recognizing Arabic words or defining text styles. In Contrast to the Arabic language, a great amount of work has been done for other languages such as Latin [43], Hebrew [44], and Chinese [45] [46]. Therefore, and to the best of our knowledge, there is no publicly available benchmark for styles of Arabic handwritten text. In all the previous works [11] [47] [48] [49], the researchers have used their small, auto-collected, very simple, and personal image datasets. Some works in the literature have used the digital printed text [50] [51], as an alternative of handwritten text [46] [11] [48], for different tasks such as text extraction [52], segmentation [53], font recognition [54], word recognition[55], or writer identification [56]. However, opting for digitally printed text, as an alternative to handwritten text, does not reflect the real word problems.

Indeed, there exist some works in the literature where authors have tested their methods on real handwritten Arabic artistic text. In [11], a method based on statistical features, named Edges Direction Matrix (EDMS), for Arabic calligraphy recognition has been proposed where the images are represented by 22 statistical moments. Nevertheless, this method has been evaluated using a small dataset that contains 700 images categorized into seven styles. Another work in [48] has been proposed aiming to classify the JAWY (Malizian language) text styles. In this work, 20 letters must be manually segmented before being analyzed. The main weakness of this study is the difficulty (sometimes impossible in artistic texts) of manually isolating the different letters in a word. Adam & al [49] have adopted a similar technique based on letter isolation. Instead, they have used Local Binary Pattern (LBP) and Gabor Filters (GF) as image features and SVM as a classifier.

Although the aforementioned methods have proven their effectiveness, they suffer from the problem of results insufficiency due to the use of small and unpublicized datasets. Additionally, the used datasets lack complexity and hold images of words with easily-separable letters, which is not the case in most artistic calligraphy styles.

In this study, we put forward the richest handwritten Arabic calligraphy dataset that comprises 1685 images categorized into 9 categories. The introduced dataset offers real word problems in Arabic calligraphy styles. To meet real word cases, the data has been collected from heterogeneous sources including books, manuscripts, and the web. In addition, we have proposed a new system for automatic style classification of handwritten Arabic calligraphy using a combination of multiple classifiers. The proposed system has shown promising results as it will be shown in the experimental section.

The rest of the chapter is organized as follows: In section 2, we give a general overview of the proposed system and the different methods of classifier combination. Section 3 is devoted to present our collect dataset. In Section 4, experimentation will be conducted and results will be reported. Finally, we draw some conclusions.

3.2 The proposed system

Our main aim is to recognize Arabic calligraphy text style automatically. The proposed solution is based on a multi-classifier technique that has recently been widely utilized in different tasks such as object classification [57] and character recognition [58]. For image representation, we choose the LPQ texture descriptor. Texture-based methods have been established successfully in various applications of digital document analysis [59]. A general scheme of the proposed system is given in Figure 3.3.

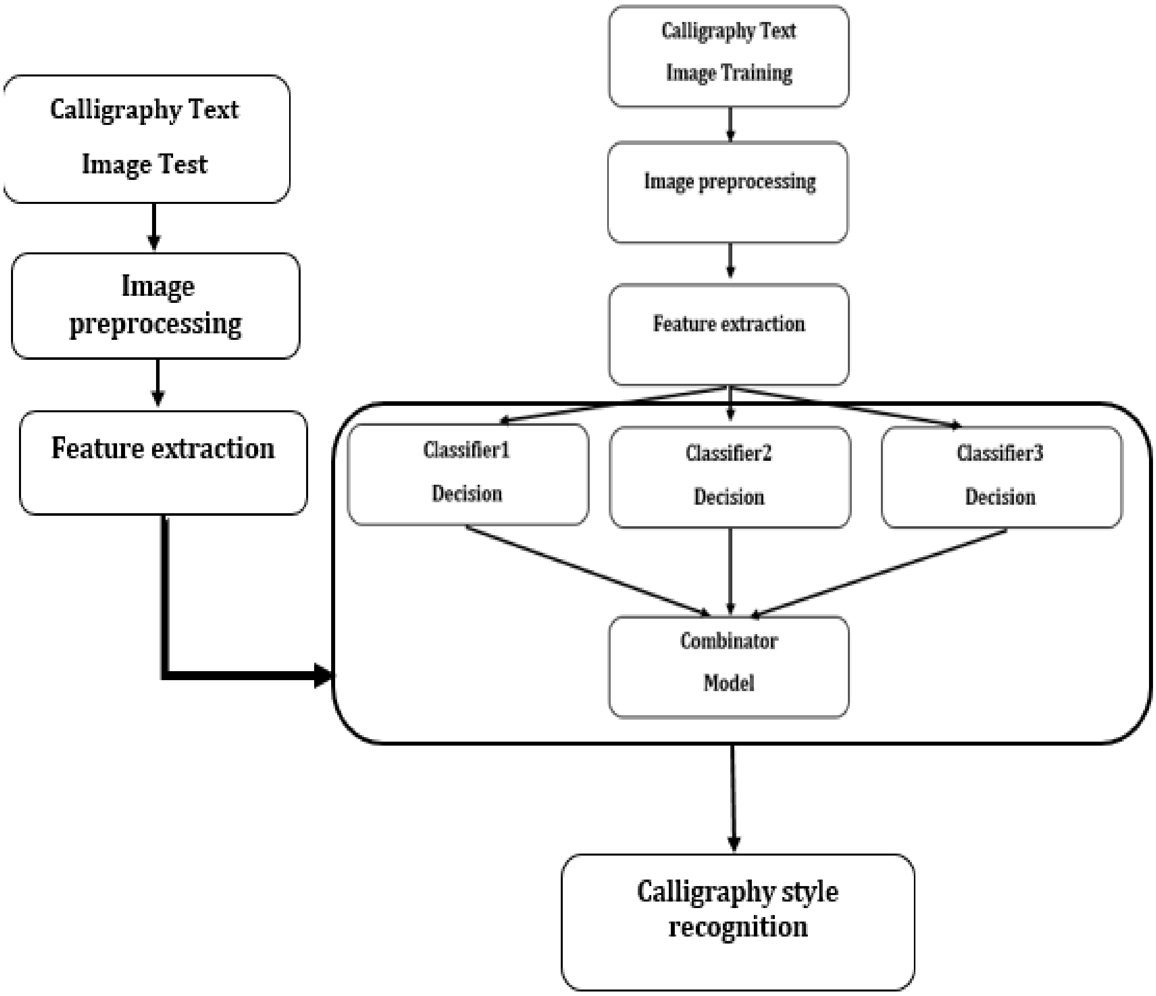
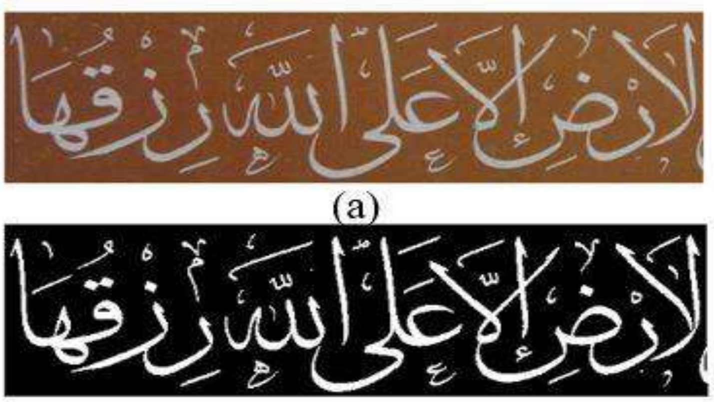


Figure 3.3 The scheme of the proposed system for the Arabic Calligraphy style recognition.

Firstly, the image is subjected to a preprocessing step to clean noise, binarize and crop the text. From each image in the training set, a Local Phase Quantization (LPQ) descriptor is extracted then fed to a multi-classifier module for training purposes. Likewise, test images are passed through the same process for identification purposes. The main reason behind using multiple classifiers instead of one is the intuition stating that multiple decisions are better than one.

3.2.1 Preprocessing

At this stage, we binarize all the images to get the clean text without extra information that could affect the recognition process. The adopted method is Otsu's [60], which establishes a threshold that minimizes the intra-class variance (Figure 3.4).



(b)

Figure 3.4 image binarization using Otsu's method (a) color image (b) binary image.

3.2.2 local phase quantization LPQ (feature extraction)

The local phase quantization is a blur insensitive texture descriptor, mainly invented to deal with image blurring. The LPQ method is based on the blur invariance property of the Fourier phase spectrum [61]. At each pixel position x of the image, the local phase information is computed over an M-by-M neighborhood using the 2-D DFT (Fig 3.5) or, more precisely, a short-term Fourier transform (STFT) using the following equation.

$$(u, x) = \sum{}_{y} \in N^{x} (x - y) -j^{2\pi u}\, Ty = wuTf$$

where wu is the basis vector of the 2-D DFT at the frequency u, and f is another vector containing all M2 image samples from Nx.

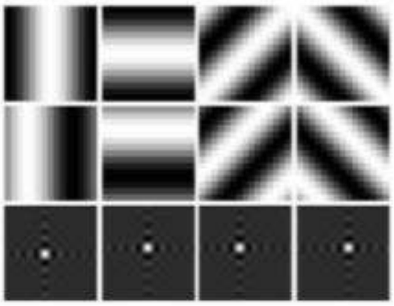


Figure 3.5 The short-term Fourier transform (STFT "uniform window").

3.2.3 Classification & combination

This step aims to assign each image to its corresponding class by employing, in a parallel manner, the decision of three different classifiers namely: a Multilayer Perceptron (MLP), a Support Vector Machine (SVM), and a K-Nearest Neighbors (KNN) classifier. These three classifiers have shown sufficient results on our dataset. It should be mentioned that there are several methods for combining multiple classifiers decisions' each with its pros and cons. In order to improve the classification results, we have tried multiple combination techniques namely majority vote, maximum, minimum, Sum. We, chose them for their effectiveness, ease of use, and because they do not require a complex calculation. As shown in this paper [57] the best classifier combinations technique is dependent on our data.

- Majority/Plurality vote means the correct class is the one chosen (voted) by most of the classifiers. Even if the sum of those votes did not exceed 50% of all votes obtained from the ensemble of classifiers. If all the classifiers indicate different classes, then the final class is the class with the overall maximum output score value.

- Maximum, the final score is the maximum between the classifier's output scores. That means if one classifier insists on a specific class for a given test sample, the final decision is what it assigns to that class, even if all other classifiers disagree.

- Minimum, the final assigned score is the maximum between the minimum classifiers output scores. That means for each classifier we take the minimum score for a given sample, then we choose the maximum between them.

- Sum, summing the output scores of each base classifiers and assigns the final class label with the maximum score to the given input sample.

3.2.4 The new dataset presentation

As mentioned earlier in this thesis, the literature lacks a public benchmark for handwritten Arabic calligraphy style classification. Although researchers have used some personal datasets, these datasets suffer from the smallness, homogeneity, and simplicity, which dispose them from being challenging. To tackle this issue, we propose and make publicly available a new dataset that holds the number of the height of images and categories among others. The proposed dataset holds the following advantages:

- Richness: Different categories represent the basis of calligraphy styles.
- Size: Sufficient number of images in each category.
- Heterogeneity (variation): Scale and sentence length variation for the same category.

The proposed dataset contains **9** artistic styles of calligraphy which are Naskh, Diwani, Parsi, Rekaa, Thuluth, Maghribi, Kufic, Mohakek, and Square-Kufic where each style represents a category /class as it is presented in Figure 3.6.

The number of images in each class is between 180 and 195 images as given in Table 3.1. For the capturing process, a camera with a resolution of 18 megapixels has been used to get images of sufficient quality. To grant variation, the pictures have been taken from different sources such as books, manuscripts, and from the web. The texts within the images have been carefully and manually segmented to formulate phrases with different lengths. Finally, the obtained images (i.e., 1685 images) have been categorized in their appropriate font styles.

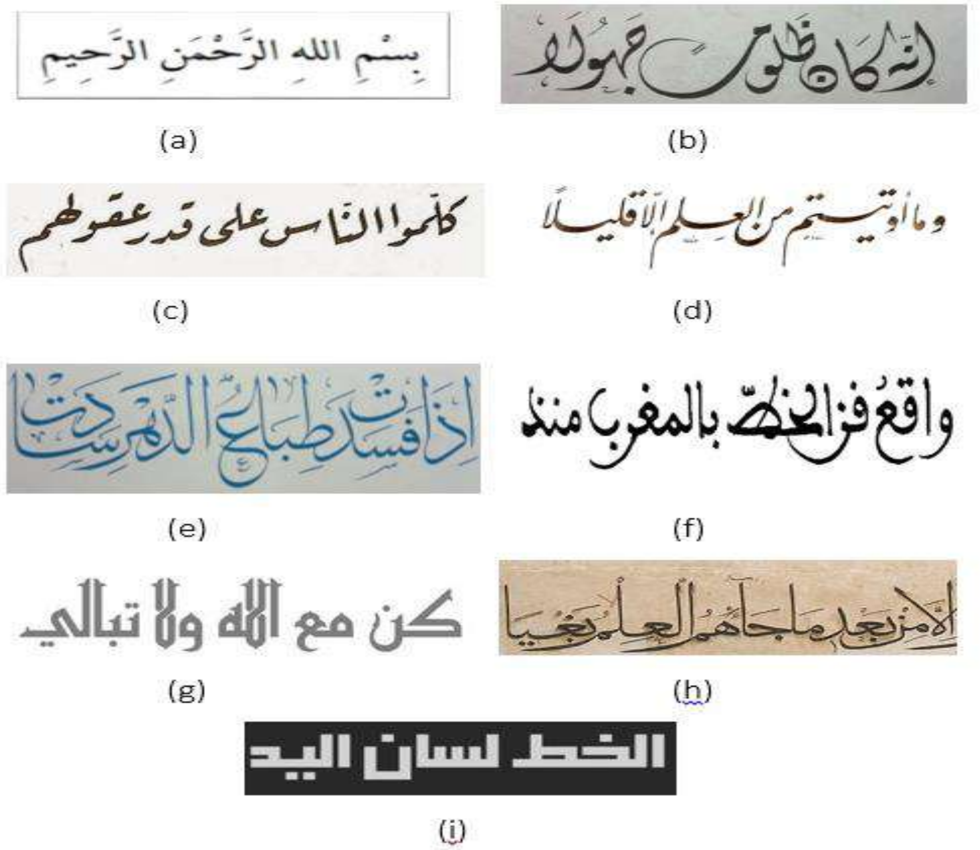


Figure 3.6 the dataset Arabic calligraphy styles.

Table 3.1 the different styles with their respective number of images in our dataset.

Category style	**# Of images**
Diwani	190
Naskh	190
Parsi	180
Rekaa	185
Thuluth	195
Maghribi	180
Kufi	185
Mohakek	190
Square-Kufi	190
Total images #	**1685**

3.3 The experimental results:

In this section, we accomplish two different experiments. The first one is devoted to separately evaluating each classifier whereas the second one is for evaluating the effect of decision combination. In order to avoid over-fitting, K-Fold (k=3) cross-validation has been used. The motivation to use cross-validation techniques is that when we fit a model, we are fitting it to a training dataset. Without cross-validation, we only have information on how does our model performs to our in-sample data. Ideally, we would like to see how the model performs when we have new data in terms of the accuracy of its predictions [62] [63]. The dataset is firstly divided into three equal sub-sets, and then, per each test sub-set, we use the other two for training. The average accuracy of the three folds, for each classifier, has been presented in Table 3.2.

Table 3.2 average accuracy for each single classifier

Classifier	**Average accuracy**
MLP	94.52%
KNN	93.21%
SVM	94.66%

We can see that the classifiers have yielded similar results with an error rate of 5.34%-6.79%. In order to reduce this error rate, the three classifiers have been combined and evaluated using the same configuration. We evaluated the performance of the proposed system by comparing it with the recent Edges Direction Matrix (EDMs) proposed by Batainah & al [64]. For faire comparison, we apply the EDMs method on our dataset. The EDMs statistically analyze the relationship between pixels of the boundary edges of a binary image. The final feature vector is calculated from the edges EDM1 and EDM2. The proposed method extracts 22 statistical moments (correlation, homogeneity, etc.). Table 3.3 presents the results obtained with our system using different combination methods.

Table 3.3 the average accuracy of multiple-classifiers system with different combination methods

Combination methods	**Avg accuracy**
Majority voting	95.32%
Maximum	93.02%
Sum	**96.31%**

Minimum	96.11%
EDMs+ Decision-Tree [11]	69.09%

As it is shown in Table 3.3, the proposed system has yielded the best result among all by reducing the error rate to 3.7% instead of 5.34% reported by SVM separately. This confirms the intuition stating that opting for multiple decisions instead of one improves results. Additionally, we can see that Batainah [11] have yielded poor results compared to the others. This could be attributed to the sensitivity of the latter method to the complex and heterogeneous artistic calligraphy texts. It, also, seems that the Sum combination methods have reported better results than the others have. This is because the Sum method involves all the classifiers in the decision-making in contrast to the others where the decision is made by the most (resp. least) dominant(s).

3.4 Conclusion

Despite the importance of Arabic calligraphy handwriting, intensive work should be carried out trying to alleviate the problems concern with text/style recognition. This chapter put forward a major twofold contribution to Arabic calligraphy font styles. On the one hand, it provides a rich and diverse image dataset that might be used for different tasks of text and style recognition. On the other hand, we provide an automatic system based on the LPQ texture descriptor and multiple classifiers combination for Arabic calligraphy style recognition. Overall, the obtained results demonstrate the effectiveness of combining multiple decisions by yielding the highest performance compared to a single decision or the work of Batainah & al [11].

Chapter 4

Arabic Artistic Script Style Identification Using Texture Descriptors.

Chapter 4 : Arabic Artistic Script Style Identification Using Texture Descriptors

Contents

This chapter deals with Arabic Artistic Script Style Identification Using Texture Descriptors. we design and evaluate a system for Arabic artistic style identification using numerous texture descriptors[65]. These descriptors have been chosen based on their effectiveness and the high performance they showed in other related works. To make it more comprehensive, the system has been evaluated using five different classifiers. Results indicated that texture descriptors appropriately fit the task and yielded almost perfect performances for most of the styles.

Published in: 2020 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP)
Date of Conference: 16-17 May 2020
Date Added to IEEE Xplore: 29 July 2020
ISBN Information:
INSPEC Accession Number: 19855521
DOI: 10.1109/CCSSP49278.2020.9151569
Publisher: IEEE
Conference Location: El Oued, Algeria

Arabic Artistic Script Style Identification Using Texture Descriptors

Abstract

Texture descriptors have been widely used for many computers vision tasks. Document Analysis (DA) is a hot topic of research, which aims at analyzing digitized documents containing different types of texts. In this chapter, we design and evaluate a system for Arabic artistic style identification using numerous texture descriptors. These descriptors have been chosen based on their effectiveness and the high performance they showed in other related works. To make it more comprehensive, the system has been evaluated using five different classifiers. Results indicated that texture descriptors appropriately fit the task and yielded almost perfect performances for most of the styles.

Keywords— Arabic calligraphy; texture descriptors; feature extraction; Optical Font Recognition (OFR).

4.1 Introduction

Digital Image Analysis (DIS) is the set of techniques that are used to represent the content of images appropriately. This representation can be used thereafter for recognizing, differentiating, and quantifying diverse types of images. DIS techniques have widely been used in several application areas such as robotics, handwriting recognition, document analysis, image indexing and retrieval, diagnostic assistance in medicine, object detection, pattern recognition, etc.

Document Analysis (DA) is a hot topic of research, which aims at analyzing digitized documents containing different types of texts such as numbers [66], letters [67] [68], words [69], sentences, or paragraphs [70]. Frequently, DA systems utilize both DIS and machine learning techniques to automatize their tasks. As illustrated in Figure 4.1, DA systems involve a set of majors including pre-processing, feature extraction, and classification. The purpose of each of these steps can be resumed as follow:

- **Pre-processing:** aims to de-noise or crop the image to remove useless information; threshold/binarize the image to highlight objects or separate foreground from the background.
- **Feature extraction:** is about extracting the most significant information from the image. The result of this step is a value, vector, or matrix that can be used later on to discriminate images.
- **Classification:** This step consists of two phases, a) training: in which we teach our model the common properties of each class of images by jointly providing the features and labels of each class. b) Classifying: this phase consists of feeding unlabeled images to the trained model to assign them to their appropriate classes (i.e., labeling them).

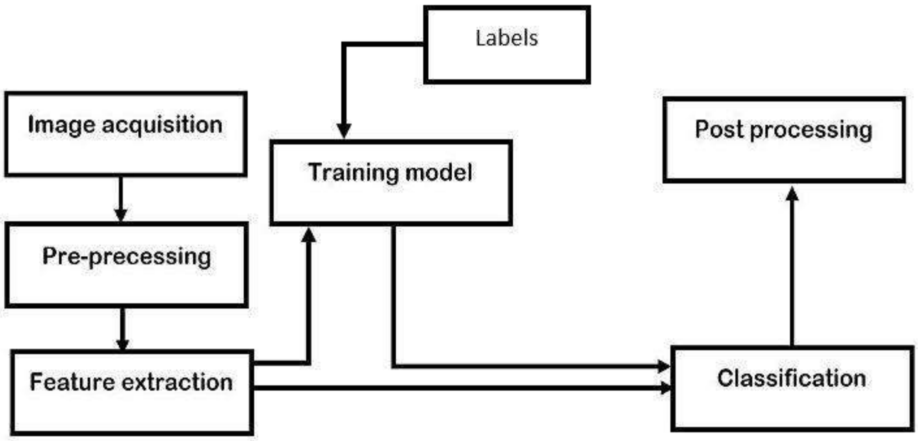


Figure 4.1 A common architecture of digital document analysis (DA) systems.

Feature extraction is a key step that is substantially addressed in many image c1assification systems including DA. Features can be defined as the individual measurable heuristic properties of the phenomena being observed [71]. A considerable number of image features have been proposed in the literature based on different approaches such as color, texture, shape, points, edges, or object-based features. The choice of the features to be extracted is guided by the image content and the needed type of information.

As for digitized DA systems, images contain a text composed of a finite number of letters (e.g., 28 letters in Arabic) and sometimes diacritics as in the Arabic language. Thus, a digitized document image will comprise repetitions of the same or similar letters and diacritics. Texture features have therefore widely been used in DA systems in which the image can be decomposed in patterns, either repeating or non-repeating [71].

In a former work [72], they have proven that texture is a powerful feature that may outperform even deep-learning-based features in several datasets. Numerous solutions for DA systems have been established based on texture descriptors. In chapter 3, we proposed a new system for Arabic calligraphy styles recognition based on Local Phase Quantization (LPQ) and a combination of three different classifiers namely K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP). Alaie & a1. performed a comparative study between twenty-six texture descriptors to demonstrate their performance on three public datasets [59]. A proposed solution from Ghosh & a1 [73] utilized LBP descriptor to isolate non-text from text components in handwritten documents. ln [74], authors have proposed a technique for writer identification using Histogram of Oriented Gradients (HOG) and Gray Level Run Length (GLRL) Matrices.

As we have mentioned, texture features have extensively been used, in literature, for DA systems. However, there is no former work that addresses the performance of texture descriptors in Arabic calligraphy/artistic documents classification.

This work aims to evaluate some of the most powerful texture descriptors for DA purposes. These descriptors are Gray Level Co-occurrence Matrix (GLCM) [74], Local Binary Pattern (LBP) [75], Weber Local Descriptor (WLD) [76], Histogram of Oriented Gradient (HOG) [77], Local Phase Quantization (LPQ) [61] and Binarized Statistical Image Features (BSIF) [78]. The choice of these descriptors has been performed based on the effectiveness they showed in different other application areas such as biometry [79] [80] [81]for LBP and BSIF, object detection for HOG, handwriting recognition [5] for LPQ, computer-aided diagnostics [82] for WLD and image retrieval[83] for GLCM. However, it should be mentioned that to the best of our knowledge none of the aforementioned descriptors have ever been considered for Arabic calligraphy document classification.

The rest of the paper is organized as follow: In section 2 we introduce the involved texture descriptors. Section 3 holds the experiment results and discussions. Finally, we draw some conclusions.

4.2 Texture descriptors

In this work, we study the effect of using texture descriptors in DA systems for Arabic calligraphy document classification. The focus has been on six of the most powerful and commonly used texture descriptors which are GLCM, LBP, WLD, HOG, LPQ and BSIF.

A. GLCM [74]

Among the well-known and widely used techniques for texture representation. It comprises a set of second-order statistics that measures the spatial dependency among gray-levels. Given a displacement vector $(\Delta x, \Delta y)$ and an image I with a size of $M \times N$, GLCM can be extracted using Eq. 1

$$M(p,q) = \sum_{i=1}^{N}\sum_{j=1}^{M}\begin{cases} 1 & \text{if } \mathrm{I}(i,j) = p \text{ and} \\ & I(i+\Delta x, j+\Delta y) = q \\ 0 & \text{otherwise} \end{cases} \quad ..(1)$$

Haralicks has introduced a set of statistical measurements that can be extracted from GLCM. In this work we extract only four statistical measurements of Haralicks namely contrast, correlation, energy, and homogeneity, these features are the least correlated together and these four sufficed to give good results in classification.

B. LBP [75]

LBP is a simple yet very efficient texture descriptor that consists in mapping each pixel of the image, based on its neighborhood, into its corresponding binary code as shown be Eq. 2. The value of the LBP code of a pixel (x_c, y_c) is calculated using the following formula:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s\left(g_p - g_c\right)2^p \quad s(x) = \begin{cases} 1, \text{ if } x \geq 0 \\ 0, \text{ otherwise} \end{cases} \quad (2)$$

Where P is the number of neighbor points on a circle of radius of R around the central pixel.

The appearance frequency of each binary code is thereafter accumulated into a histogram and considered as the descriptor of the image.

C. WLD [76]

Extracts at each pixel level in the image two measures namely, differential excitation and orientation. Differential excitation is the function of the ratio between the intensity of this pixel against its neighbors, whereas, the orientation is the gradient orientation of the corresponding pixel. These two components are respectively given by the following formulas:

$$\xi = \arctan\left(\frac{\Delta I}{I}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

$$\varphi_t = f_Q(\theta') = \frac{2t}{\pi}\pi, t = mod\left(\left\lfloor \frac{\theta'}{2\pi/T} + \frac{1}{2} \right\rfloor, T\right) \ldots (4)$$

Where in Eq. 3, ΔI represents the difference in intensity and I represent the intensity of the current pixel, to find the pixel excitation. Eq. 4 to get the orientation, where θ is the orientation angle at each pixel level.

The appearance frequency of one component given the other is then accumulated in a histogram and consider as the descriptor.

D. HOG [77]

Feature descriptor consists of calculating the appearance frequency of gradient orientation. To extract HOG from a patch image (i.e., $64 \times 128\, pixels$) the gradient orientation and magnitude are calculated for each pixel $p(x, y)$ using the following formulas respectively:

$$\varphi(x, y) = \arctan\left(\frac{f_y(x,y)}{f_x(x,y)}\right) \quad \ldots\ldots\ldots\ldots\ldots(5)$$

$$e(x, y) = \sqrt{f_x^2(x, y) + f_y^2(x, y)} \ldots\ldots\ldots\ldots\ldots(6)$$

Where fx(x, y) and fy(x, y) are the differences of brightness in the horizontal and vertical direction respectively.

Subsequently, a HOG histogram (9 × 1) is extracted from each cell, counting the number of occurrences of each orientation, and then concatenated to form the final HOG histogram.

E. LPQ [61]

LPQ is a blur insensitive texture descriptor, mainly invented to deal with image blurring LPQ method founded on the blur invariance property of the Fourier phase spectrum [61]. At each pixel position x of the image, the local phase information is computed over M-by-M neighborhood using the 2-D DFT or, more precisely, a short-term Fourier transform (STFT) using the following equation:

$$(u, x) = \sum y \in N_x\, (x - y) - j^{2\pi u\, Ty} = w^{uT} f \ldots\ldots\ldots\ldots\ldots(7)$$

Where w^u is the basis vector of the 2-D DFT at the frequency u, and f is another vector containing all M2 image samples from N_x.

F. BSIF [17]

Computes a binary code for each pixel by linearly projecting local image patches onto a subspace, whose basis vectors are learned from natural images via independent component analysis (ICA), and by binarizing the coordinates in this basis via thresholding. The number of basis vectors determines the length of the binary code string.

$$s_i = \sum_{u,v} W_i(u,v)X(u,v) = \mathbf{w}_i^{\top}\mathbf{x} \text{..........(8)}$$

Where W is the learned filters, X is the image window

4.3 Results and discussion

This section aims to evaluate the performance of GLCM, LBP, WLD, HOG, BSIF, and LPQ for calligraphy image classification. To this end, we use five different classifiers namely Random Forest (RF), Support Vectors Machines (SVM), Decision Tree (DT), K-Nearest Neighbor (KNN), and Local Discriminant Analysis (LDA). The following configurations, which found to be the best, have been set:

A. Classifiers parameter's

For some of the classifiers, we select the best parameters to fit out our data:

- For SVM: we choose 'polynomial' kernel, and 'onevsone'.
- KNN: we set k from "1 to 10" nearest neighbor.

- RF: we put 500 for the number of the trees.

B. Features parameter's

In this part we give the used parameters with all the studied descriptors:

- GLCM: we Define four offsets, with pixel distance (PD=1, 3, 5), and four orientations (0^o, 45^o, 90^o, 135^o, 180^o). Four statistical moments have been extracted namely contrast, correlation, energy, and entropy.
- LPQ: we use "Gaussian derivative quadrature filter pair", and a window size of 3*3.
- LBP: we use the uniform-LBP, with a 3*3 for the window size.
- WLD: we set the block size equals to 5*5, the window size 3*3.
- BSIF: we use 8 texture filters with size of 3*3.
- HOG: we set a 3*3 the number of HOG windows per bound box, and 19 the number of the histogram bins.

4.3.1 Dataset

Currently, there is no universal Arabic handwriting calligraphy-text dataset for font recognition. Therefore, we did create a public dataset in our previous work (chapter 3), that we will use in this work, it consists of 1685 image for 9 different styles, we use different techniques to collect the dataset images from different resources. Our dataset is a set of images for handwritten Arabic calligraphy texts, with different challenges, it contains different categories represent the basis of calligraphy styles, To the best of our knowledge, one of these styles is used for the first time, which is the Mohakik style. A sufficient number of images in each category. Variation means to scale and sentence length variation for the same category. Figure 4.2. Illustrate some images from the used Arabic calligraphy dataset.

Figure 4.2 . Arabic calligraphy handwritten styles.

4.3.2 Metrics

For performance measuring, we use cross-validation. That means the dataset is divided into 3 sub-sets, at each time one set will be for testing and the two others for training 3 times alternately. Then, we use three performance-measuring criteria, which are precision, recall, F1 score, and accuracy:

- Precision: Precision is defined as the number of true positives over the number of true positives plus the number of false positives.

$$\mathbf{P} = \frac{\textbf{true positive}}{\textbf{true positive}+\textbf{false positive}}$$

- Recall: Recall is defined as the number of true positives over the number of true positives plus the number of false negatives.

$$\mathbf{R} = \frac{\textbf{true positive}}{\textbf{true positive}+\textbf{false negative}}$$

- Accuracy: accuracy can tell us immediately whether a model is being trained correctly and how it may perform generally

$$\textbf{Accuracy} = \frac{\textbf{true positive}+\textbf{true negative}}{\textbf{true positive}+\textbf{false positive}+\textbf{true negative}+\textbf{false negative}}$$

- F1-score: is the harmonic mean of precision and recall, where an F1 score reaches its best value at one (1) (perfect precision and recall) and worst at zero (0).

$$\textbf{F1-score} = 2 \cdot \frac{\mathbf{R\,.P}}{\mathbf{R+P}}$$

Table 4.1 illustrate the obtained mean accuracy results from all the texture descriptors. From the classification results shown in Table 4.1 it is clear that the best texture descriptor that provided the highest accuracy are BSIF with 94.8% for the SVM classifier, LPQ with 93.8% for SVM classifier, LBP with 89.8% for RF classifier, HOG with 87.3% for 3%, WLD with 81.1% for RF classifier, GLCM with 60.4% for RF classifier, respectively. We did notice that the best classifiers were SVM then the Random-Forest classifier. In this step, the accuracy gives us how the overall classifier performed with our data for style classification. Based on the mean accuracy results the best descriptors for font recognition is BSIF descriptor with SVM classifier. In the next step we will establish a detailed study about these couple using Precision, recall, F1-score, and the confusion matrix, to see how they actually work with each class separately.

Table 4.1 Comparison of the mean accuracy results of all studied texture descriptors with five different classifiers.

	RF	**SVM**	**DT**	**KNN**	**LDA**
GLCM	**60.4%**	52.5%	55.2%	57.9%	55.6%
LBP	**89.8%**	84.6%	82%	84.2%	76.4%
WLD	**81.1%**	74.8%	75.4%	69.6%	61.3%
LPQ	93.1%	**93.8%**	85.5%	87.3%	73.8%

HOG	84.9%	**87.3%**	63%	78.3%	85.1
BSIF	93.9%	**94.7%**	85.2%	84.4%	85.9%

F1 score measure, precision and recall are shown individually for all the texture descriptors with all the classifiers in Figure 4.3/ Figure 4.4/ Figure 4.5.

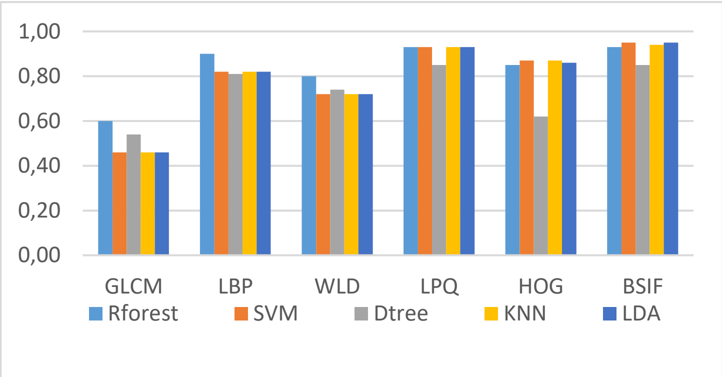


Figure 4.3 F1 score measure for all texture descriptors with all the classifiers.

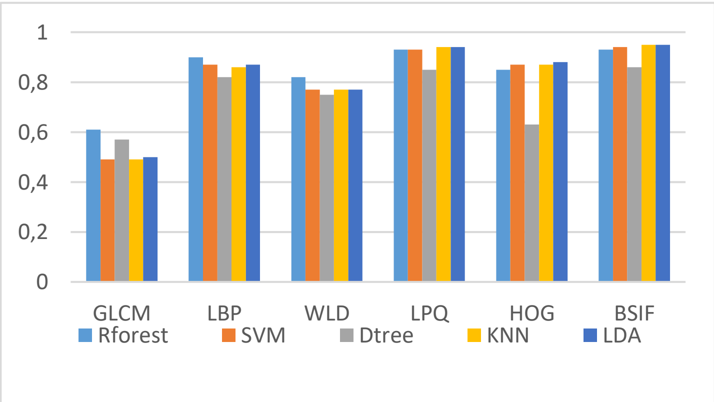


Figure 4.4 Precision.

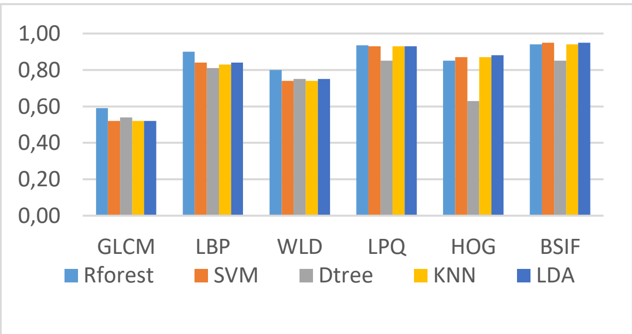


Figure 4.5 Recall.

In the present study, we found that the best couple (Descriptor/ Classifier) is (BSIF with SVM). Therefore, we give more detailing results for the last-mentioned in Table 4.2.

Table 4.2 the three measures of BSIF descriptor with SVM classifier (Precision/ Recall/ F1-Score).

	Precision	**Recall**	**F1-score**
Diwani	0.9948	1.0000	0.9974
Naskh	0.9845	1.0000	0.9922
Farisi	0.9714	0.9278	0.9490
Rekaa	0.9371	0.9727	0.9545
Thuluth	0.8919	0.8410	0.8624
Kufi	0.9677	0.9389	0.9503
Maghribi	0.9498	1.0000	0.9728
Mohakik	0.8486	0.8629	0.8525
Square-kufi	1.0000	0.9841	0.9920

Regarding the three measures each one individually, BSIF classify correctly all the images belongs to three styles (Diwani, Naskh, and Maghribi) in term of Recall. Mostly the same thing for precision, where Square-kufi and Diwani reached the highest precision with Naskh style, respectively. To have a deep look at this result, the confusion matrix has been shown in Table 4.3.

Table 4.3 Confusion matrix for BSIF descriptor and SVM classifier.

	Diwani	**Naskh**	**Farisi**	**Rekaa**	**Thuluth**	**Kufi**	**Maghribi**	**Mohakik**	**Square-Kufi**
Diwani	**1,00**	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Naskh	0,00	**1,00**	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Farisi	0,00	**0,01**	**0,93**	**0,07**	0,00	0,00	0,00	0,00	0,00
Rekaa	0,00	0,00	**0,03**	**0,97**	0,00	0,00	0,00	0,00	0,00
Thuluth	**0,01**	**0,01**	0,00	0,00	**0,84**	0,00	0,00	**0,15**	0,00
Kufi	0,00	0,00	0,00	0,00	0,00	**0,94**	**0,06**	0,00	0,00
Maghribi	0,00	0,00	0,00	0,00	0,00	0,00	**1,00**	0,00	0,00
Mohakik	0,00	**0,01**	0,00	0,00	**0,10**	**0,03**	0,00	**0,86**	0,00
Square-kufi	0,00	0,00	0,00	0,00	**0,01**	0,00	0,00	**0,01**	**0,98**

From the confusion matrix, it is obvious that the highest error rate is 15% for the missed classified images from Thuluth to Mohakik and the inverse 10% for the missed classified images from Mohakik to Thuluth. This could be referred to as the two style general characteristics that are much similar even for the human eye (Figure 4.6). Thuluth and Mohakik mainly have the same characteristics, like the writing style and the diacritics. Except in some cases for Thuluth style, they would exaggerate with the letter size and make them more overlapped like in (Figure 4.7).

Figure 4.6 The similarity between (a) Mohakik (b) Thuluth.

Figure 4.7 Thuluth style with extra overlapped letters.

4.4 Conclusion

The present study concentrates on comparing several texture descriptors that have been used widely in different classification problems. We compare their performance using five different classifiers, for each of the features and classifiers we set the best parameters. The experimental results proved that the transform-based texture feature extraction has performed well for Arabic calligraphy style recognition, especially for BSIF texture descriptor with SVM classifier.

Chapter 5

A new Computational method for Arabic calligraphy style representation and classification

Chapter 5 : A new Computational method for Arabic calligraphy style representation and classification

Contents

In this chapter we present a new machine learning-based tool for Arabic calligraphy style recognition. We propose a new tool for Arabic calligraphy style recognition (ACSR)[84]. This chapter aims to identify Arabic calligraphy style (ACS) from images where text images are captured by different tools from different resources. To this end, we were inspired by the indicators used by experts to distinguish different calligraphy styles. These indices were transformed into a descriptor that defines, for each calligraphy style, a set of specific features.

Journal: Applied Science. Appl. Sci. 2021, 11(11), 4852;
DOI: https://doi.org/10.3390/app11114852
Received: 16 April 2021 / Revised: 20 May 2021 / Accepted: 21 May 2021 / Published: 25 May 2021
Publisher: MDPI.
(This article belongs to the Topic Applied Computer Vision and Pattern Recognition)

A new Computational method for Arabic calligraphy style representation and classification

Abstract

Despite the importance of recognizing Arabic calligraphy styles and their potential usefulness for many applications, a very limited number of Arabic calligraphy style recognition works have been established. Thus, we propose a new tool for Arabic calligraphy style recognition (ACSR). The present work aims to identify Arabic calligraphy style (ACS) from images, where text images are captured by different tools from different resources. To this end, we were inspired by the indices used by human experts to distinguish different calligraphy styles. These indices were transformed into a descriptor that defines, for each calligraphy style, a set of specific features. Three scenarios have been considered in the experimental part to prove the effectiveness of the proposed tool. The results confirmed the outperformance of both individual and combine features coded by our descriptor. The proposed work demonstrated outstanding performance, even

with few training samples, compared to other related works for Arabic calligraphy recognition.

Keywords Arabic Calligraphy; feature extraction; machine learning; document image analyzing.

5.1 Introduction

Document Analysis (DA) is one of the computer vision applications. It is a discipline that combines image processing and machine learning techniques to process and extract information from document images of different text types such as digits or alphabet. DA attempts to extract the layout from a scanned document page and then reuse it to generate a new visually similar document with other content. Therefore, it must be capable to identify writing characteristics, such as the style used in writing, and employ it to reproduce similar documents with the same style.

DA techniques have been employed in many applications such as document layout analysis[85], signature verification[86], writer identification[87], Optical Character Recognition (OCR), and Optical Font recognition (OFR), etc. OFR for instance, aims to define the text writing style from a document image. Based on the purpose of the writing and the source of the written text, text writing can be categorized into three categories as shown in Fig 5.1. The first one is texts written by humans which in turn comprises two sub-categories namely ordinary handwriting and Calligraphy. The second one is machine-printed texts that may be written using common styles. Thus, text produced by machine or handwritten calligraphy uses well-known styles. Writing styles used by a machine are referred to as fonts whereas calligraphy refers to the set of styles that might be used in handwriting texts. Fonts and calligraphy also differ in the purpose of use, calligraphy is mostly used for decoration like in mosque walls, or in artist's paintings, whereas fonts are mostly used for documentation and communication.

Besides, Fonts are far less complicated than calligraphy. They may provide alternatives, ligatures, and some stylized versions, but fundamentally, they can only be as creative as type designers. When writing letters by hand, there are no limitations to how to write them. Every version of a letter can be written differently. letters may be merged or connected even though they are not adjacent. Consequently, Fonts does not offer flexibility compared to handwriting. Such flexibility makes it an extremely hard task for a DA system to recognize calligraphy or reproduce text using similar calligraphy styles. In this work, we aim to propose a tool that’s able to classify Arabic calligraphy styles.

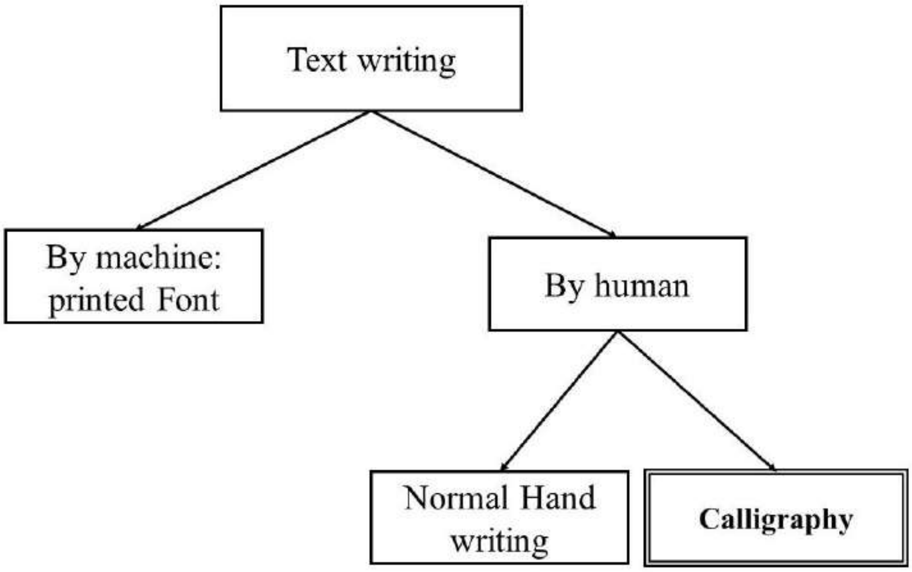


Figure 5.1 The different categories of text writing styles. Our work falls in the highlighted category of Calligraphy text.

Calligraphy Style Recognition (CSR) is considered an important research area for the following reasons:1) It helps to read the text content. In other words, knowing the used style reflects the rules used in the writing of different characters, which in turn ease the reading task. 2) It helps to recognize the different parts of a document. Some styles are defined to represent specific parts in the document such as titles, footers, paragraphs, etc. 3) It also helps to grasp the history of a document. In paleography, CSR is used to define the era in which the document has been written because styles have appeared in different eras.4) It also helps to define the origins (i.e., geographical area) in which a document has been written. 5) It can be used for tutor purposes. Calligraphy learners use CSR systems to judge the quality of their writings in cases of experts' absence.

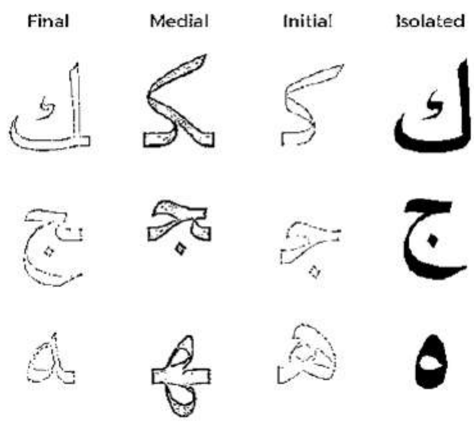
Arabic is the fourth most spoken language in the world and it is the first language of more than 200 million people across the world. Arabic script is the second most widely used script after Latin script. Like other languages, it has grammar, spelling, punctuation rules, pronunciation, slang, and idioms. Several characteristics beyond the mere differences between languages make Arabic distinctive, including the number of variations and the written form. Arabic handwritten text is divided into two main parts: artistic writing called calligraphy and non-artistic (handwritten or printed) scripts. Arabic calligraphy (AC) is one of the most significant arts in the world. It is written only by hand. The first style of calligraphy was developed at the end of the 7th century. By the time, the Muslim and Arab world had developed many other styles of calligraphy. Due to the complex shape of the Arabic text, one must hold a certain level of expertise to be able to write AC texts or recognize a writing style.

Before delving into Arabic calligraphy systems, we must shed a light on Arabic Font recognition (AOFR) and how they differ from calligraphy recognition. The AOFR domain has received a lot of attention for several reasons: a)The availability of public big datasets, b)the ease of dealing with Arabic Fonts due to their fixed shapes c) and the considerable amount of related works [41] [88]on the subject. On other hand, far less attention has been given to Arabic calligraphy style recognition (ACSR). For the other languages, however, several works have been done to this end such as the work [89] in which topological features have been proposed to recognize Hebrew handwriting styles. Nonetheless, the complexity of calligraphy styles makes it possible to use only two characters, the character Aleph and the character Lamed. Other attempts have been carried out for Chinese style recognition in [90] proposes a feature extraction method based on the regional guided filter (RGF) with reference images, which is generated by KNN matting and used as the input image for RGF, and in [91] using deep features with SVM and NN classifiers.

Despite the efforts in other languages to find a common solution for style recognition, these features are not efficient and applicable for Arabic due to their nature. Arabic calligraphy texts are extremely smooth, interconnected, and have a unique characteristic in their form. Arabic has 28 letters, each one has isolated and connected forms (at the beginning, center, and end) as illustrated in Figure 5.2. Another difficulty in Arabic calligraphy is the possibility of letters overlapping one above the other, which is not the matter with other languages. Additionally, the shapes of letters are affected by both the writing and the writer’s style, and also by the purpose of the writing itself (e.g., for decoration, documentation, or other purposes).

خ	ح	ج	ث	ت	ب	أ
khaa’	Ḥaa’	jiim	thaa’	taa’	baa’	alif
ص	ش	س	ز	ر	ذ	د
Saad	shiin	siin	zaay	raa’	dhaal	daal
ق	ف	غ	ع	ظ	ط	ض
qaaf	faa’	ghain	:ain	DHaa’	Taa’	Daad
ي	و	ه	ن	م	ل	ك
yaa’	waaw	haa’	nuun	miim	laam	kaaf

(a)



(b)

Figure 5.2 Components of the Arabic language. (a) The alphabet consisting of 28 characters, and (b) the different forms of the same character based on the position in the word.

In this chapter, we propose a robust solution for AC style recognition from images. Our proposal consists of two main phases: 1) feature extraction from AC images, and 2) classification through decision fusion. The extracted features, which represent the different morphologies of AC, have been used for the first time to the best of our knowledge in this work and have been inspired by the characteristics experts (calligraphers) use for recognizing AC styles. Since each feature is dedicated to distinguishing one or more styles, multiple classifiers have been engaged in the process of recognition in which the decisions are fused to obtain a final decision. It should be mentioned that none of the former works on AC recognition has implicated such a wide variety of styles (9 styles) as we did in this work. Our proposal achieved state-of-the-art and also provided stability against scale variation and learning from a few samples.

The remainder of this chapter is organized as follows. In Section 2, we clarify Arabic writing characteristics, styles, writing complexity of each style, and the reasons we consider 9 styles. Section 3 lists and discusses highly related works in the field of AC style Recognition. We thereafter introduce our proposal in Section 4 and experimentally evaluate it in Section 5. Finally, we conclude with a general conclusion and some perspectives.

5.2 Characteristics of Arabic handwritten calligraphy and its complexity

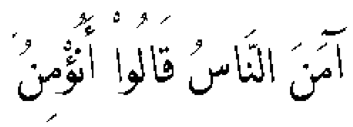
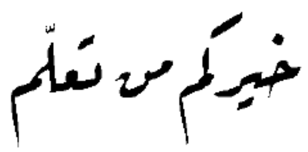
Arabic consists of 17 main character forms. With the addition of dots placed above or below certain of them, it provides a total of 28 letters. As shown in Figure 5.2 above, the same letter shape can form a "b" or “baa” sound when one dot is placed below (ب), a "t" or “taa” sound when two dots are placed above (ت), or a "th" or “thaa” sound when three dots are added above (ث). Short vowels are not included in the alphabet but instead indicated by signs (diacritics) placed above or below the consonant or long vowel that they follow. The Arabic writing system is a Right to Left (RTL), which means words are written and read in RTL way like Hebrew, Parisian, and Urdu. Texts are written in a cursive way where letters must be interconnected unlike other languages, such as English or French, where connecting letters is a matter of choice rather than obligation. One must also know that a letter in Arabic can be written in three different forms based on its location in the word (beginning, middle, or at the end) as shown in Fig 5.2.

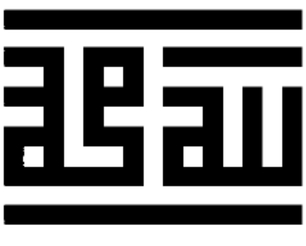
5.3 Arabic calligraphy styles

AC consists of two main styles (Kufi and Naskh), each style has several variants, as well as region-specific styles. In the history of Arabic calligraphy, there were six main styles namely Kufi, Diwani, Farisi, Naskh, Rekaa, and Thuluth each of which has its characteristics and writing rules. By the time, over 100 different styles have appeared and identified driven by the former main styles. Nine distinctive categories of cursive styles have evolved, becoming ever more refined and elegant, some of which are still used and some others have disappeared. The commonly used styles nowadays are the six basic styles (namely: Kufi, Diwani, Farisi, Naskh, Rekaa, and Thuluth) and three other styles that are driven from them which are Maghribi, Mohakik, and Square-Kufi. Unlike other related works which dealt with a few of these styles, in this work we consider the full list of the nine styles. Table 5.1 resumes the different characteristics and the cues that experts use in identifying each one of these styles. It also provides a representative sample from each style to show how it differs from others.

Table 5.1 *The nine most common AC styles with their characteristics and cues that used by experts to identify them.*

AC style	**Example image**	**General characteristics**	**Cues used by experts**	**Style source**
Kufi		Composed of geometrical forms like straight verticals/horizontals lines and distinguishable angles.	-It is angular. -Thick strokes. -long vertical extended strokes.	
Diwani		A cursive script, beautiful, and harmonious. It is characterized by the	- slant words. -written with a rounded shape.	

		rounded shape of letters. Words are written skewed descend from right to left. Lines start thick and then attenuate.	-doesn’t follow a straight line. -the style width changes from thick to thin.	
Farisi		Characterized by the letter's accuracy and extension, and by its ease and clarity and lack of complexity. Farisi (alt., Nasta'liq) is a cursive text.	-simple orientation. -doesn’t follow a straight line.	
Naskh		One of the simplest writing types of calligraphy. Its name means “the copy style” because it has been used for copying books. It is recognizable by its balance and its plain clear forms. Nowadays, this script is used primarily in print.	-simple orientation. -written on the base-line. -slight change in the text thickness.	
Rekaa		Mostly, used without diacritics. known with its clipped letters. Composed of short, straight lines and simple curves, as well as its straight and even lines. It is written above the baseline except for some specific letters.	-simple orientation. -written above the baseline. -thick stroke.	
Thuluth		It is an elegant, cursive script that has large,	-overlapping letters.	

		elongated, and elegant letters. It has certain roles, but it could be changed according to calligraphers' needs. It has big letters, which take a big space above the baseline; the big overlapped curves result in big holes that calligraphers fill with artistic diacritics.	-big rounded letters. -high number of diacritics. -it doesn't consider the baseline.	
Maghribi		It has special letterforms which provide it with a unique beauty and make it easy to read, even in long texts. It is marked by descending lines written with very large bowls.	-text orientation. -written on the baseline	-Kufi
Mohakik		Mohakik and Thuluth have almost the same characteristics. However, letters in Mohakik are longer under the baseline.	-special orientation under the baseline. -same diacritics as the Thuluth style. -written on the baseline.	-Thuluth
Square kufi		A unique style developed for decorating walls rather than papers. usually designed with mosaic faience, decorative glazed facing tiles, or simple	-strokes with equal thickness. -Square angles.	-Kufi

		bricks, rather than reed pens and ink. It has two strict rules: a) evenness of full and empty spaces, and b) square angles and strict lines**.**		

Due to the complexity of AC and the shared characteristics among its styles, none of the former related works has considered the full list of 9 styles. However, there were some serious efforts to come up with some specific techniques for recognizing AC styles. In the following section, we list and discuss works that aimed at recognizing Arabic calligraphy styles using computer vision techniques.

5.4 Related work

Arabic text font recognition has been extensively studied owing to the availability of a huge number of resources that help researchers to build new solutions. Works on Arabic font recognition can be grouped into two categories, namely: handcrafted- and deep learning-based categories. Starting with handcrafted-based solutions, Kellal [29] uses discrete curvelet transform (DCT) to convert images into descriptors, and then, a backpropagation neural network has been employed to predict the style of the image text based on the extracted descriptor. In [22] [92], the former method has been reinforced against scale changes by using a steerable pyramid texture. Hussein [41] has experimented and compare several texture descriptors for Arabic font recognition using Extreme Learning Machine (ELM) and Fast Learning Machine (FLN). On the other hand, Deep convolutional neural networks (DCNN) [40] [93] have been utilized in several occasions to perform Arabic font and font size recognition.

While there has been much research on Arabic Font, few researchers have dealt with Arabic calligraphy. It could be attributed to the lack of resources such as public datasets and insufficiently related works. However, the reader should know that dealing with calligraphy is much harder than fonts due to the absence of creativity in the latter. In other words, calligraphy texts are affected by the calligrapher impression which is not the case with texts generated using machine fonts. Therefore, we limit ourselves to works that tackle the issue of recognizing calligraphy styles rather than Fonts. Table 5.2 lists works conducted on AC style recognition with their respective datasets and the number of styles took into account.

Table 5.2 List of related works on ACSR.

Author (year)	Language	Dataset	Nº Styles
Batainah et al. (2012) [11]	Arabic JAWI	700 blocks image(private)	6
Batainah et al. (2013) [12]	Arabic-JAWI	700 block image (private)	7
Batainah et al. (2011) [13]	Arabic	14 documents images(private)	Unknown
Batainah et al. (2011) [14]	Arabic	100 line image(private)	6
Talab et al.(2011) [15]	Arabic	700 line images	7
Azmi et al. (2011) [16]	Arabic-JAWI	100 character image (private)	5
Azmi et al. (2011) [17]	Arabic-JAWI	1019 character images(private)	4
Adam et al. (2017) [18]	Arabic	330 character images (private)	6
Allaf et al.(2016) [19]	Arabic	267 line/word Images(private)	3
Elhmouz & al. (2020) [20]	Arabic	421 line/word Images(private)	3
M. Khayyat(2020) [21]	Arabic	2653-documents images(private)	6

Due to the lack of big datasets of AC and the fine-grained styles (e.g., Mohakik and Thuluth), most works in literature have utilized classical machine learning tools instead of deep learning. Moreover, they attempted to propose their descriptors for this issue instead of using other common descriptors such as Sift, LBP, etc.

In all of their works, Bataineh [11-14] and Talab [15] attempt to classify AC styles with a proposed statistical descriptor, named Edge Direction Matrix (EDMs). They firstly pass through two steps called EDM1 and EDM2 which produce 3x3 matrices counting edges (i.e., pixels' adjacency), and then extract22 statistical moments to constitute the final image descriptor. Unlike the aforementioned global approach, the works in [16-18] focused on extracting features using local shapes of letters. Allaf [19] for instance, uses the text geometrical shape, the density of text above and under the baseline, number of diacritics up and down the text, text orientation, the position of the text baseline, and the ratio of black and white for the whole image. However, the validity of this solution has been proven using a small dataset (260 images) comprising only three styles which are highly insufficient. To sum up, the aforementioned works can be categorized into two categories, holistic and local approaches. Holistic are concerning with recognizing styles using global descriptors. These approaches, however, are not dedicated to calligraphy, but they can be applied for heterogeneous images ignoring the very specific characteristics of AC. The local approaches on the other hand look into the characteristics of AC, exaggeratedly, via letter segmentation which makes it inapplicable especially for complicated styles or for big datasets. In this paper, we propose a solution that combines the advantages of both, describing the specific features of each style without resorting to letter segmentation. Although deep learning has proven itself as the best solution for many issues, it has not been widely explored for ACSR due to the fine-grained styles and absence of big datasets. We list the only two works that utilize deep learning nets for ACSR, which are [20] and [21]. In the former, a common deep model (stacked-auto encoder) has been trained and utilized for feature extraction and a SoftMax has been employed to perform the recognition process. In the latter, the full MobileNetV1 model has been fine-tuned and used to classify six AC styles.

5.5 Proposed method

In this chapter, a new tool has been proposed for the task of ACSR. The proposal is capable of recognizing nine different AC styles and might be extended to other styles. After analyzing Arabic calligraphy, we found out that the styles are geometrically different. Some styles have a unique shape, some share similar letter shapes, and some others have special diacritics. For the visually different styles, it is easy to separate them, whilst for the other fine-grained styles, we needed to extract one or more distinctive characteristics to separate them. The proposed solution was inspired by the calligraphy expert. After an in-depth study of each style, looking for the visual characteristics used by calligraphers to distinguish it, we came up with main distinctive features that need to be extracted for each style or set of styles. In the feature extraction section, we explain these visual features and the relationship with their corresponding styles.

Our proposed tool consists of three main steps, a) image preprocessing: perform several transformations to prepare the image for feature extraction, b) feature extraction: extract the proposed descriptor from each image, and c) classification: use the extracted features to predict the style of the input image. Figure 5.3 illustrates the general scheme of our proposed tool.

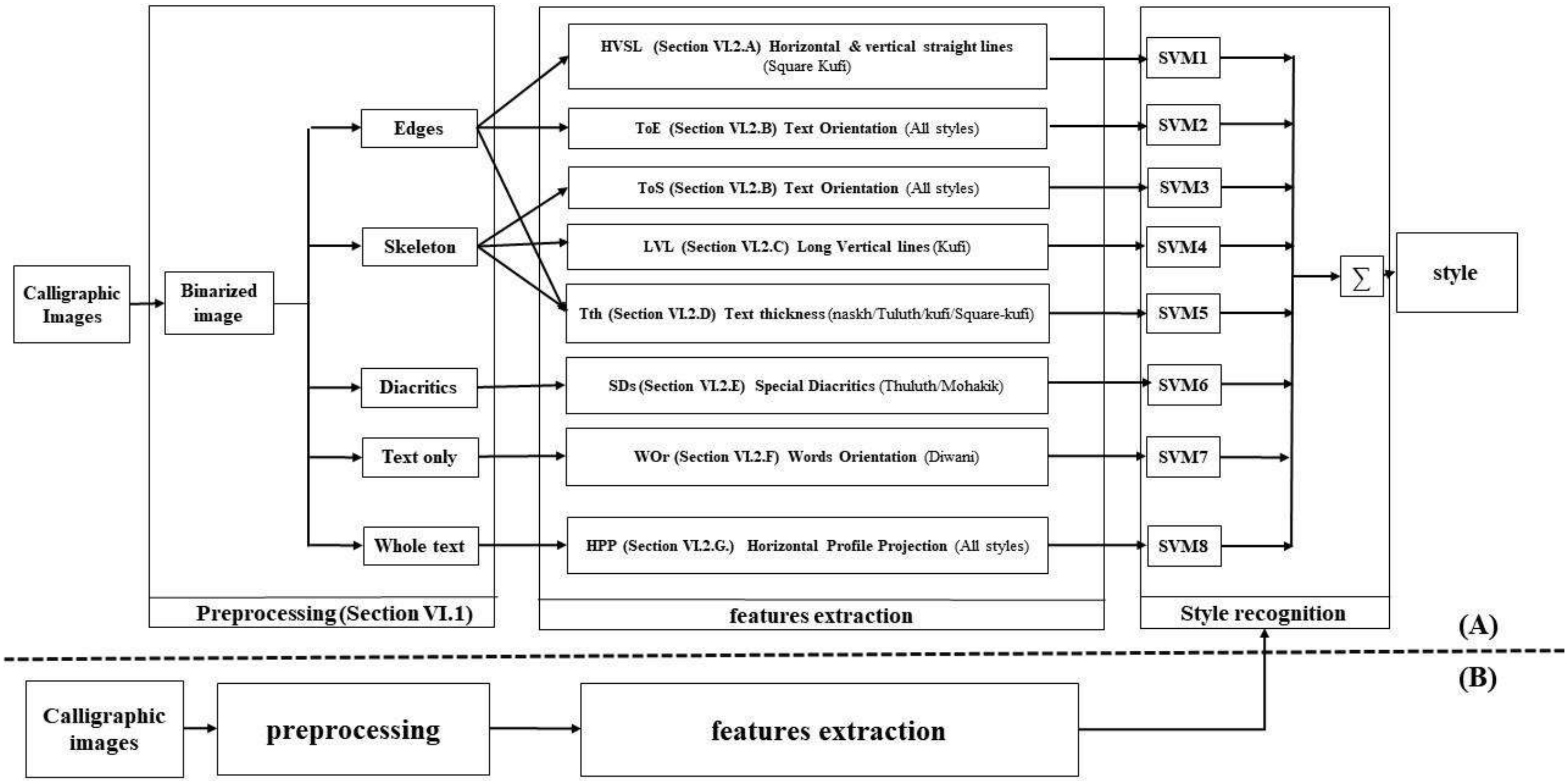


Figure 5.3 A general architecture for our proposed tool. The dotted line separates the (A) training from (B) test phases. The images pass through a processing sequence starting from preprocessing, features extraction, and finally classification. The final descriptor contains eight features, as shown, which will be explained in Section V.

5.5.1 Preprocessing

Since we are interested only in the morphology of text letters, all images are first converted into binary (i.e., black text on a white background) after having the texts separated from backgrounds [94]. It should be known that most images contain either meaningful texture which is a part of the decoration or a meaningless one that resulted from the noise while capturing. In either case, we illuminate the background so it does not affect the results. Thereafter, we apply the following micro-operations to extract, for each image, a set of images namely: edge Image, skeleton image, diacritics image, and no-diacritics image. Each of these generated images will thereafter be used to define the characteristics of one or more styles of AC. Figure 5.4 shows some images that resulted from these operations.

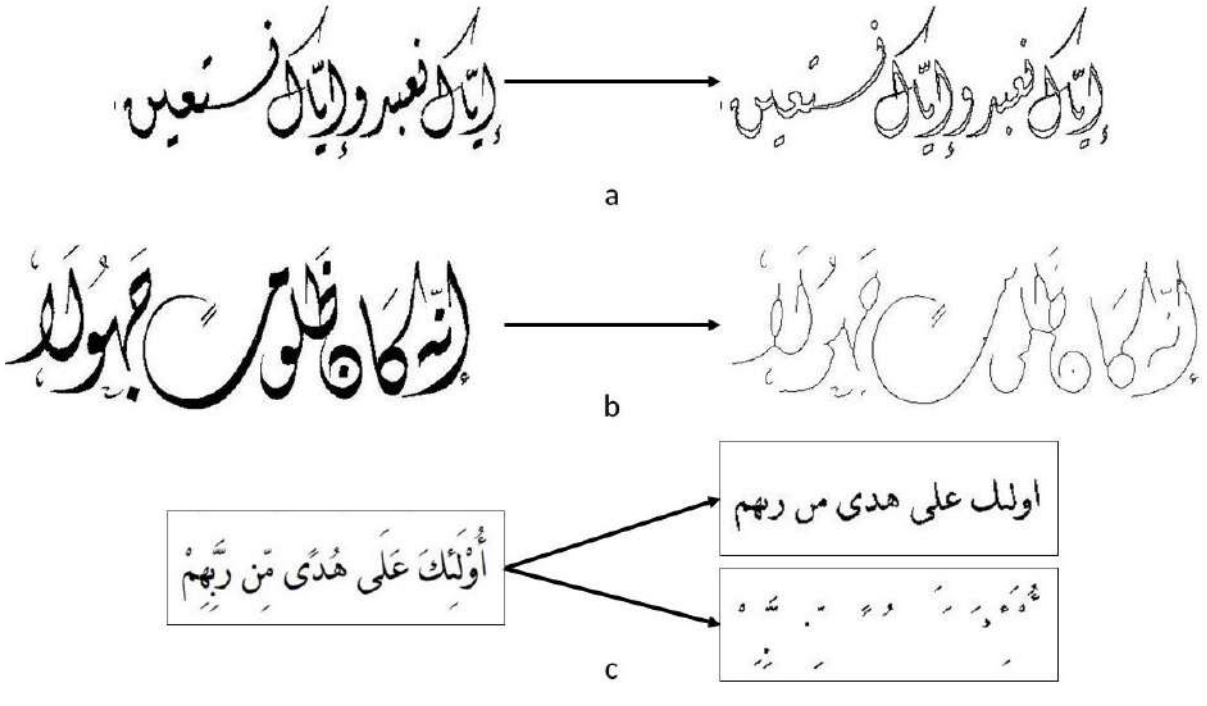


Figure 5.4 Operations executed in preprocessing step to generate a set of accompanying images for each image in the dataset. Each generated image will thereafter be used to extract one or a set of features. These operations are, (a)edge detection using a 3x3 Laplacian filter,(b) skeleton detection using 'Zhang-Suen Thinning Algorithm'[95], and (c) Separating diacritics from text using the flood-fill algorithm[26].

5.5.2 feature extraction

In most fields, it is a common practice to ask experts about cues they use to decide for a certain inquiry. Calligraphers use some specific features, which we will discuss hereafter, to find the appropriate style for a given text. The main aim of this step is to transform these features into values that can be fed to a machine so it can decide on the style of a given text image. It should be mentioned that a feature might specify one or more styles. Features designated for each style or set of styles might be used in a sequential manner (sequential decision) or parallel manner. In our work, however, we considered all the features to be equally important and, therefore, adopt a parallel approach. That is to say, each feature descriptor is fed to a classification machine, SVM in our case, and the final decision is the combination of all the decisions of these machines.

A. *Horizontal & Vertical Straight Lines (HVSL):*

This descriptor has been designed to separate Square Kufi from other styles. It mainly describes how frequently horizontal and vertical lines appear in a text image. Square Kufi differs from other styles by the high number of vertical and horizontal lines as shown in Figure 5.5.

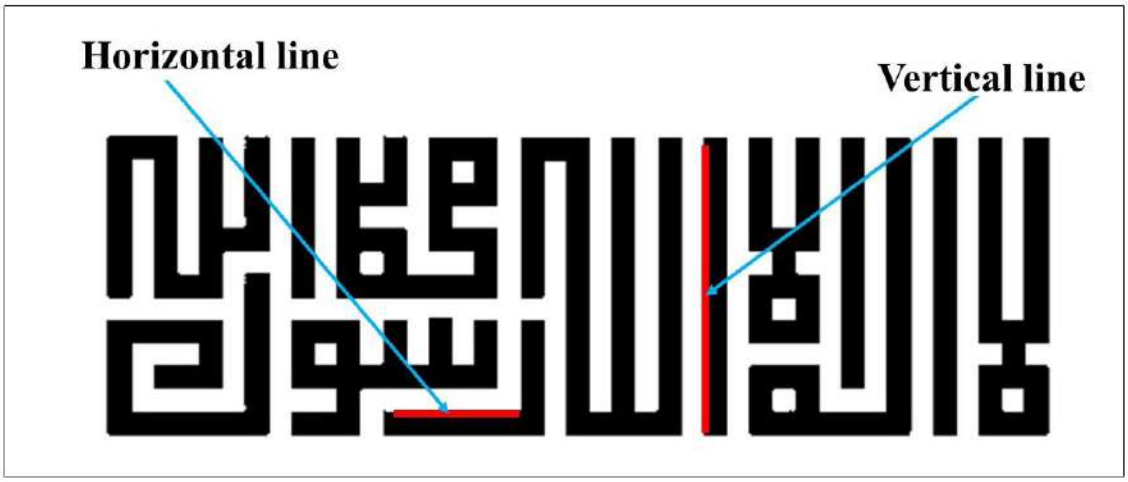


Figure 5.5 Vertical and Horizontal lines as basic features of Square Kufi.

HVSL are not extracted from the input image, but rather from the edge image resulted from the preprocessing step. The final HVSL descriptor holds the following features: a) The appearance frequency of vertical and horizontal (i.e., V/H) lines, which is the main cue used to distinguish Square-kufi, b) The difference ratio between the number of the pixels that constitute the texts edge and the sum of V/H lines appearance frequency.

B. Text orientation (ToE/ToS):

One of the main characteristics of all the Arabic calligraphy styles is text orientation. Each style is usually written in a specific direction that defines its unique shape. By direction, we mean the slope of the pen while writing words. For instance, the pen in kufi is used to write on the baseline without any slope, whereas, in Rekaa the writing is slightly sloping. Besides, the pen itself might be held by the calligrapher vertically or in a sloppily way based on the style intended to be adopted. A vertical grasp of the pen results in flat tails whereas a sloping grasp results in pointed ones as Figure 5.6 shows.

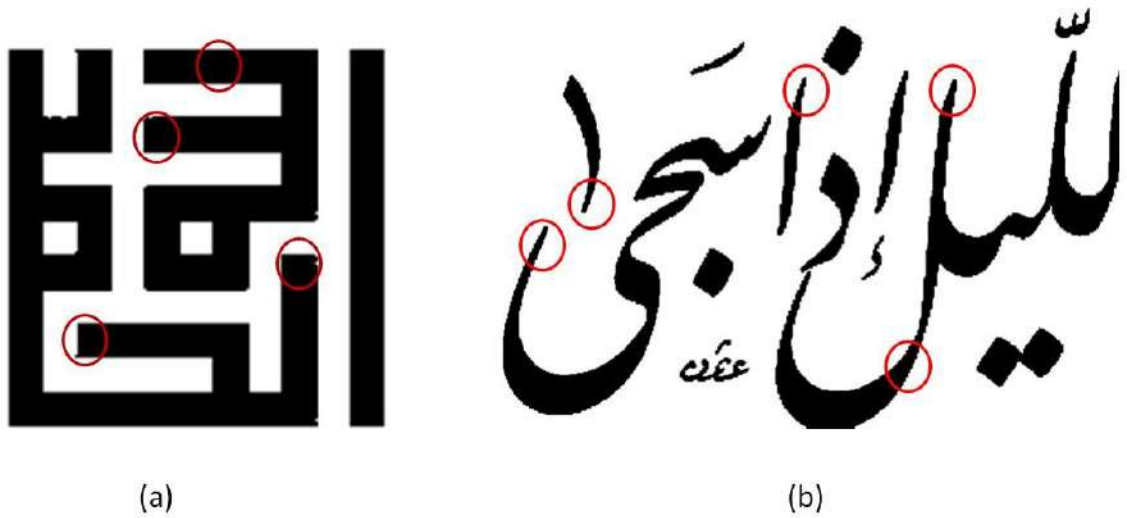
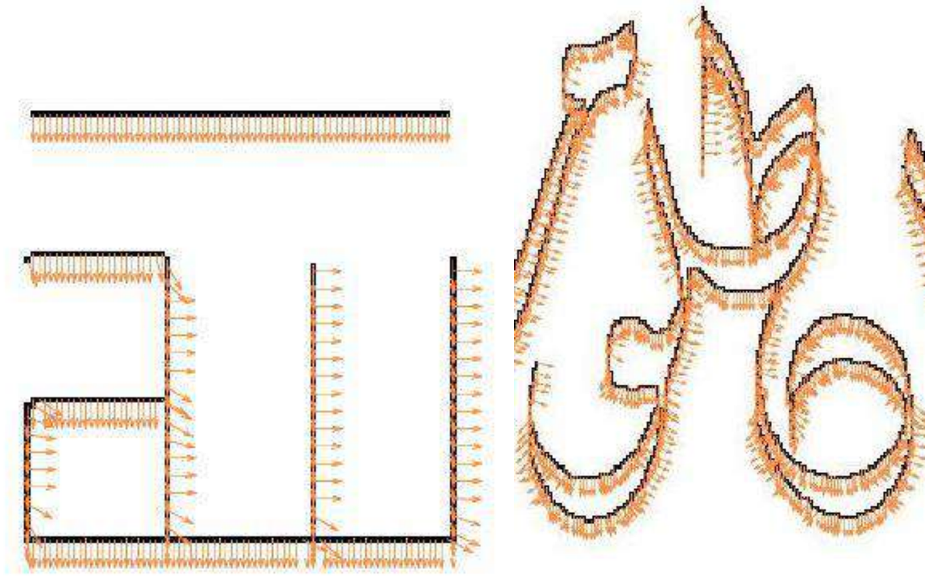


Figure 5.6 The effect of pen hold manner on the writing. In (a) the pen is held straight vertical whereas, in (b) the pen is held in a sloping manner.

To capture the orientation of words, we utilize the skeleton image, whereas, to capture the effect of pen sloping we use the text edge image. From each image, the orientation at each pixel level is extracted and then codified into a histogram (i.e., ToE for edge orientations and ToS for skeleton orientations). Figure 5.7 shows a representative example of the process.



(b) (c)

Figure 5.7 Extracting orientations to be used to generate ToE and ToS histograms. Orinantions extracted from (a) the skeleton image, and (b) the edge image.

C. Long vertical lines (LVL):

Since Kufi and Square Kufi share common features such as vertical and horizontal lines, HVSL will consider Kufi images as square kufi and separate them from other styles. However, Kufi has a distinctive characteristic, compared to Square kufi, which is the long vertical straight lines. Long Vertical Lines (LVL) descriptor has been designated to eliminate the conflict between texts written in kufi and the ones of square kufi. It mainly describes how vertical lines in a text vary.

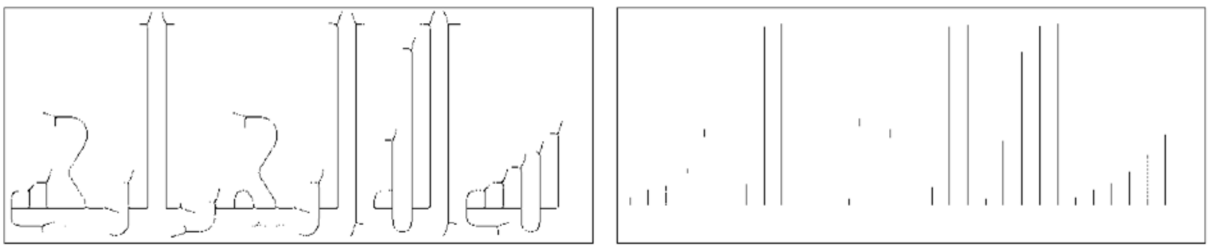


Figure 5.8 vertical straight lines detection.

After having the vertical straight lines extracted from the skeleton image as Figure 5.7 shows, the following five measurements are calculated: a) the text height from the bottom to top, b) the number of detected vertical lines, c) the length of the highest detected vertical line, d) the difference ratio between the text height and the highest vertical line, and e) the variance among the vertical lines.

D. Text thickness (Tth):

Stroke thickness plays an important role in defining the style. Some styles use a flat pen, whereas some others use a pointed one. In some styles, calligraphers alter the thickness while writing (via pushing down the pen or the opposite), whereas in others the thickness is always preserved. Modeling such a feature in form of a descriptor will help the machine to understand more specificities of each style. Text thickness (Tth) descriptor codifies the appearance frequency of different line thicknesses in a text image. To extract this descriptor, we employ both the skeleton and edge image. To find the thickness level at a pixel of the skeleton, we calculate the distance between the two points on a perpendicular line that passes through p.

E. Special diacritics (SDs):

Thuluth and Mohakik have a similar writing style that is decorated with diacritics having special shapes as shown in Figure 5.8. SDs descriptor will be used to inspect the existence of such diacritics in a given text image. To this end, each of the diacritics represented in Figure 5.8 is explicitly segmented and then represented with a vector of Hu moments[96]. Thereafter, the distance among these moments and those extracted from the input image's diacritics are calculated to decide whether the image is Thuluth/Mohakik or not.

Figure 5.9 Thuluth & Mohakik diacritics.

It is worth mentioning that SDs hs not been designed to identify Thuluth from Mohakik, but rather to separate both of them from other styles.

F. Word's orientation (WOr):

One of the salient features of the Diwani style is the words written in a slanted format as Figure 5.9 shows. WOr descriptor mainly specifies how, on average, words in text images are oriented. The flood-fill algorithm is used to detect words from no diacritics images that are generated in the processing step. After calculating each word's orientation, the mean orientation along with the number of detected words are combined to form the final WOr descriptor.

Figure 5.10 A slanted word from Diwani style, with its orientation measured, detected using the Flood-fill algorithm.

WOr algorithm will be used to distinguish Diwani from other styles. Diwani style will yield an orientation average of about 45 degrees compared to 0 degrees by other styles.

G. Horizontal profile projection (HPP):

In some AC styles, all words are written on a baseline, whereas in some other styles, words are not subjected to a baseline. Horizontal profile projection is a descriptor that specifies how words are vertically spread within a text image. First, a sub-image that exactly fits the text is cropped from the text image, and then all pixels are projected to a vertical histogram, using the following equation (1), to find how pixels are vertically spread.

$$HPP = \sum_N I(X,Y) \Big/ \max_M \left(\sum_N I(X,Y)\right) \quad (1)$$

Where *I* is the binarized image. (X, Y) are the image dimension. N indicate the actual image. M indicate the total number of the images. In an image in which words are written on a baseline, HPP will have only one bin with a high value, whilst, in an image in which words are vertically spread HPP will have more than one bin with high value.

5.6 Experimental results and Discussion

This section is dedicated to evaluating the performance of the proposed solution. Three scenarios have been carried out, the first one is meant to evaluate each descriptor individually to validate the role it has been proposed for in the first place. In the second scenario, the decisions yielded by all classifiers are combined to produce a final decision. We devote this final scenario to examine the effect of using few samples for training. All the experiments have been carried out under the following configuration:

- **Dataset:** we use the same dataset adopted in our previous works [65] [42]. It consists of nine (9) classes each of which represents an Arabic style. The total number of the dataset's images is 1685. Each style has several images that vary from 180 to 195.
- **Classification:** for image classification, we use the well-known classifier Support Vector Machine (SVM) with a Polynomial Kernel. This choice is a result of the outperformance SVM showed compared

to other classifiers. To avoid overfitting, 3-fold cross-validation has been adopted.

- **Metrics:** to validate our model, we use four metrics, namely: Recall (R), Precision (P), F1-score, and Accuracy given by the following formulas.

$$R = \frac{true\ positive}{true\ positive + false\ negative}$$

$$P = \frac{true\ positive}{true\ positive + false\ positive}$$

$$F1\text{-}score = 2 \cdot \frac{R\ .P}{R+P}$$

$$Accuracy = \frac{true\ positif + true\ negative}{true\ positive + false\ positive + true\ negative + false\ negative}$$

5.6.1 Scenario 1. experimenting with each descriptor individually

As we discussed beforehand, some descriptors have been mainly proposed to identify one single style, whereas some others are general and can be used to separate multiple styles. This scenario is dedicated to analyzing the results for each descriptor individually. It should be mentioned that each descriptor has been used to train a separated SVM. Table 5.3 presents the accuracies yielded by each feature descriptor for each style.

Table 5.3 The accuracies yielded by each descriptor. The first column holds descriptors with the respective style they have been proposed for.

	Diwani	Naskh	Farisi	Rekaa	Thuluth	Maghribi	Kufi	Mohakik	Square kufi
HVSL (Square kufi)	63%	62%	34%	37%	83%	62%	87%	74%	100%
ToE (General)	93%	88%	94%	87%	88%	81%	94%	86%	99%
ToS (General)	96%	88%	91%	83%	90%	80%	91%	86%	98%
LVL(Kufi)	54%	77%	28%	44%	49%	34%	89%	25%	91%
Tth (General)	47%	61%	51%	52%	66%	51%	76%	43%	82%
SDs (Thuluth + Mohakik)	13%	2%	0%	10%	86%	11%	17%	62%	18%
WOr (Diwani)	85%	24%	9%	21%	9%	6%	8%	18%	62%
HPP (General)	76%	97%	58%	76%	83%	92%	97%	78%	100%

From Table 5.3, we can see that the descriptors that have been proposed for one or two specific styles have performed adequately. HVSL and Wor as instances yielded accuracies of 100% and 85% respectively which are the highest ones compared to others. However, LVL yielded a similar performance for both Kufi and square kufi although it has been proposed for the former. This could be attributed to the similarity between these two styles. The Tth descriptor works well for several styles that are characterized by their thickness. Those styles are Naskh, Thuluth, Kufi, and Square-Kufi. The highest accuracy was for Square-kufi with a correct rate of 82%. In contrast, the lowest accuracy was for Mohakik style with 43%. SDs gives 0% with Farisi style this is because Farisi style is written with no diacritics only dots. Also, with the Naskh style it gives 2% because this style is written with the normal diacritics that we use even with printed text. HPP descriptor has worked well with most of the styles because it is a representation of a general characteristic. The lowest accuracy, which is 58%, yielded with the style Farisi because it can be written in different manners (some texts are written on the baseline; in some others, words are written one above another). In addition, that Maghrbi style yielded 6% with WOr because of the big curves under the base line that could effect on the text orientation measurements.

We can conclude that general descriptors have the highest accuracies. Nevertheless, by combining these general descriptors with the specific ones, even better results can be achieved. In the following scenario, we evaluate the impact of combining all the descriptors and compare the results to other proposed methods including deep learning.

5.6.2 Scenario 2: combine the descriptors

After having proved the efficiency of the individual descriptors, we combine them to get a powerful descriptor that can be used to distinguish the nine AC styles effectively. To this end, each descriptor is used to train a separated classifier (SVM in our case) and the overall decision will be the sum of the decisions generated by all the classifiers. In other words, the final decision will be the one with the highest votes. Figure 5.10 represents the precision and recall yielded by our combined descriptor compared to other methods from state of the art.

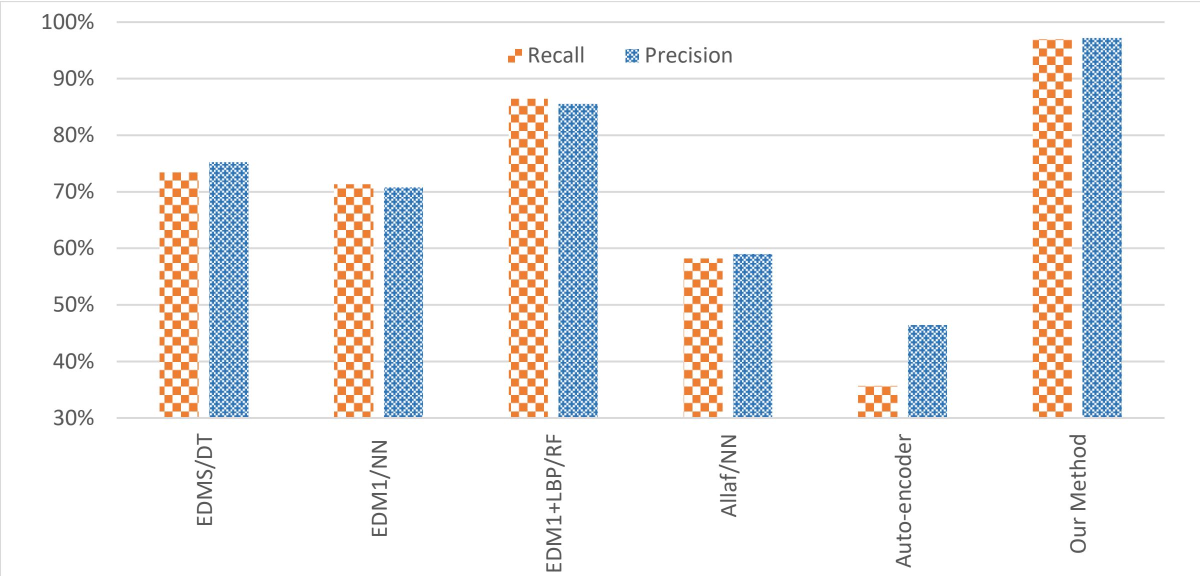


Figure 5.11 Recall and Precision yielded by the proposed method compared to other AC related works.

From Figure 5.11, it seems that our proposed method outperforms all other related works with a precision of 97%. This remarkable performance of ours is a result of combining all AC features used by calligraphers to recognize different styles in one descriptor. The method proposed in [97] (EDM1+LBP) has scored the second-best performance. This is because the latter deals with AC images as texture which is an approach, we prove its effectiveness in [66]. Although the work [19] has opted for morphological descriptors as we did, the results were too weak. This is because the authors have taken into account the general features only and did not consider the specificity of each AC style. Stacked auto-encoder, on the other hand, has scored the worst precision (resp. recall) among all. This is because using stacked auto-encoders needs tens of thousands of images to tune thousands of parameters that constitute the network, especially since the input of the network is the pixels of the image itself rather than a descriptor. Yet, no AC dataset contains this huge number of images. To further evaluate the performance of these methods, F1score at style-level has been estimated and listed in Table 5.4.

Table 5.4 F1-score at style-level yielded by our method and other related works.

	EDMS/Decision Tree	EDM1/NN	EDM1+LBP /Random Forest	Allaf/NN	Auto-encoder	Our Method
Diwani	76%	75%	90%	30%	35%	98%
Naskh	92%	87%	98%	81%	48%	99%
Farisi	56%	58%	80%	23%	2%	97%
Rekaa	54%	53%	77%	45%	12%	98%
Thuluth	63%	56%	78%	73%	43%	94%
Maghribi	66%	77%	75%	46%	15%	97%
Kufi	64%	72%	93%	73%	14%	98%
Mohakik	41%	60%	79%	45%	33%	94%
S-Kufi	96%	94%	97%	94%	80%	99%

At first glance, Table 5.4 shows that our proposed method outperforms other methods in all styles. S-kufi seems to be the most distinguishable style among the others due to its unique features. In contrast, Thuluth, Mohakik, and Rakaa seem to be harder to distinguish. This can be attributed to the common way of writing of Rakaa and the similarity between Mohakik and Thuluth. Nevertheless, our proposed method has yielded nearly perfect results because we took advantage of slight changes among styles such as the unique way of writing diacritics. With most of the styles, our proposed method has yielded more than 97% accuracy. However, with Thuluth and Mohakik our method yields the lowest accuracy which is 94%. To get an idea about how the styles Thuluth and Mohakik are misclassified, we generate and present the confusion matrix in Table 5.5. in addition, in this table we can see that the lowest results were for the auto encoder used on our data set. as we know that deep models need large training sets to achieve butter results, so, the common is to train a Deep Learning network on a very large data-set. but in our case the dataset is small, using it with the deep models could effect on its performance. we didn’t apply the data augmentation; where our main goal is to see how the existed solutions perform with our dataset.

Table 5.5 The confusion matrix, of the nine AC styles, generated by our proposed method.

Style	Diwani	Naskh	Farisi	Rekaa	Thuluth	Maghribi	Kufi	Mohakik	S-Kufi
Diwani	**100%**	0%	0%	0%	0%	0%	0%	0%	0%
Naskh	0%	**100%**	0%	0%	0%	0%	0%	0%	0%
Farisi	2%	0%	**96%**	2%	0%	0%	0%	0%	0%
Rekaa	0%	0%	1%	**98%**	0%	1%	0%	0%	0%
Thuluth	0%	0%	1%	0%	**95%**	0%	0%	3%	1%
Maghribi	0%	0%	1%	0%	1%	**99%**	0%	0%	0%
Kufi	0%	0%	0%	0%	0%	2%	**98%**	0%	0%
Mohakik	0%	0%	0%	0%	5%	0%	0%	**94%**	1%
S-Kufi	0%	0%	0%	1%	0%	0%	1%	0%	**99%**

From Table 5.5, we can see that our proposed method yields nearly perfect results with most of the styles. However, Mohakik and Thuluth have misclassified one as the other in some cases, which decreases the accuracy. This is because both styles Mohakik and Thuluth have similar shapes with less unique features that can be used to distinguish one from the other.

5.6.3 Scenario 3: Compare to other texture and Deep learning methods

In a previous work [66], we showed that AC is better to be dealt with as texture. Since AC text is relatively a homogeneous (i.e., stationary) texture with repeated patterns (characters, diacritics, etc.), using statistical feature descriptors yields better results than learning-based techniques including deep learning [72]. Furthermore, descriptors that are designed to be used with heterogenous images are harder to train (i.e., more images are needed) than descriptors dedicated to a specific family of images. That's to say, all the descriptors that we compare to our method will poorly perform in cases of a small training set. To prove these claims, an evaluation of statistical against learning-based (deep learning) methods in both normal and small training set size has been carried out. Fig. 11 represents precision obtained in both cases.

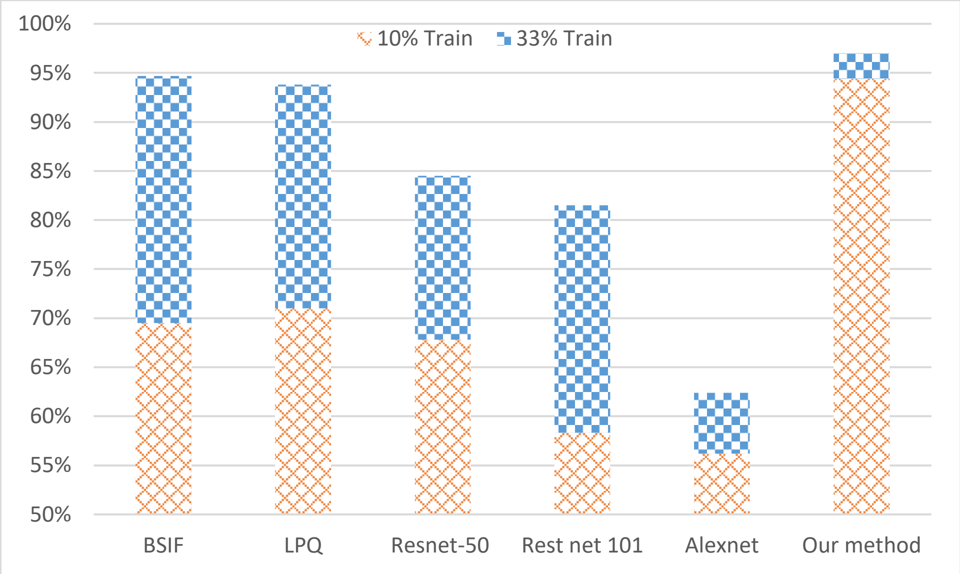


Figure 5.12 Related works evaluated and compared against our method using 33% and 10% images subset respectively for training.

From Fig 5.12, it appears that our method outperforms all other methods in both 33% and 10% train cases. As we have stated above, statistical descriptors (e.g., BSIF and LPQ) that are designed to be used with heterogeneous textures have performed well with AC style recognition. However, we have claimed that such descriptors lose performance in the case of using a few samples for training. The proof of this claim can easily be noted in the accuracy drop of all methods in the case of using a small image set for training. Our method on the other hand seems not to be greatly affected by reducing the size of the training subset to 10%. By this last scenario, we confirm that using dedicated descriptors for some specific datasets, which is the case with AC, is far more effective than general descriptors.

5.7 Conclusion

The main aim of this research was to investigate employing cues used by the calligrapher for AC style recognition. To this end, the features used by calligraphers have been codified into descriptors some of which are dedicated to distinguishing specific styles and some others are for general features. The proposed descriptor experimented as both individual and combined features forms using a rich dataset containing 1685 images categorized into nine styles. The results indicated the outperformance of the proposed method compared to other related works including deep learning. By these results, we have confirmed that exploiting the calligrapher's expertise rather than using general image features highly improves the performance of the AC recognition. Furthermore, the experiment outcomes proved that our descriptor was not highly affected by the small size of the training data, unlike other general-purpose descriptors. In this work, we have considered the most common features used by calligraphers in distinguishing styles. However, additional efforts should be spent to explore other less common features. Such features will surely further improve the results and extend the system's capability to other styles. Another aspect that needs to be tackled is text transformation, such as rotation, which is the case with most decorative texts.

Chapter 6

General Conclusion

Chapter 6 : General Conclusions

Contents

General conclusion

Arabic calligraphy style recognition is an important task due to its benefits to a lot of domains. But in reality ACSR has less attention from researchers regarding several reasons the unavailability of public datasets to work with, the luck of related works, that could help the researcher to start his new solution.

This thesis is about the analysis of Arabic calligraphy text images on a new dataset. Starting from a good understanding of the state of the art in Arabic text style recognition ACSR, we developed novel techniques and related algorithms to handle some of the current difficulties in the field.

Our analysis of the state-of-the art for Arabic-text style recognition systems lead us to the separation between Arabic text style for printed text and for handwritten text, and lead us to create new solutions specific to Arabic calligraphy because most of the existed works were based on general solutions that could be useful for other objects.

As there was no adequate dataset for printed Arabic text recognition research freely available at the time of starting our thesis, we have spent a considerable effort in building a new dataset. We provide a public Arabic calligraphy dataset, that is available for Arabic calligraphy text analyses and classification tasks. We could confirm that it is the largest and the most various dataset regarding text style numbers.

Contributions

Our main technical contributions can be summarized as follows:

- An efficient multiple-classifier system for Arabic calligraphy style recognition. This work attempts to present an automatic system for Arabic-calligraphy style classification. Therefore, we put forward and make publicly available, a new Arabic calligraphy style dataset that comprises 1685 images categorized into 9 styles. Additionally, we introduce a system for Arabic calligraphy style recognition based on the combination of multi-classifier decisions. The proposed system employs LPQ at the image representation level, whereas, it combines three different classifiers, namely SVM, MLP, KNN, at the decision level.
- Arabic Artistic Script Style Identification Using Texture Descriptors. we design and evaluate a system for Arabic artistic style identification using numerous texture descriptors. These descriptors have been chosen based on their effectiveness and the high performance they showed in other related works. To make it more comprehensive, the system has been evaluated using five different classifiers. Results indicated that texture descriptors appropriately fit the task and showed almost perfect performances for most of the styles.

- A new machine learning-based tool for Arabic calligraphy style recognition. we propose a new tool for Arabic calligraphy style recognition (ACSR). The present work aims to identify Arabic calligraphy style (ACS) from images, where text images are captured by different tools from different resources. To this end, we were inspired by the indices used by experts to distinguish different calligraphy styles. These indices were transformed into a descriptor that defines, for each calligraphy style, a set of specific features.

Detailed Conclusions

We provide here a detailed list of more specific conclusions that are summarizing the findings of this thesis work. but thanks to a thorough evaluation of the large dataset that we created; we are confident about the significance of these conclusions.

- In Arabic calligraphy, characters are represented using shapes that have different frequencies of use in the text. When it comes to building statistical models of such characters, one has to deal with very large datasets built with lexicon covering words/sentences where all characters are well represented in all different format.so our goal from creating the new dataset is to provide this diversity.
- The feature extractions used in this thesis have been partially built using classical features introduced in previous research and novel features dedicated to the Arabic calligraphy written by hand. we introduced two solutions based on the use of pre-existed features computed from the images to extract the information in an appropriate way to distinguish between the styles. We also introduced dedicated new features able to differentiate between the studied Arabic calligraphy styles.
- In all our proposed models we use machine learning for the style recognition step. After an in-depth study, we found that the classical solutions that are based on texture descriptors (pattern/ frequency/statistical based transformation) need a larger training dataset to achieve good results; in order to be able to recognize the text style in different situations. On the opposite, in our proposed model we propose a new descriptor, where we don't need a big training image set. our model was able to define the Arabic calligraphy style using a small set of training images.

- After several experiments, we found that deep learning has a weak performance with ACSR. We use the well-known pre-trained deep learning models. We use them for feature extraction with machine learning for style identification.

Future Works

We believe that there are several perspectives from this work that must be pursued.

1. **Extend the Dataset:** use more difficult text types like images from historical documents.

2. **Including more styles:** the styles of Arabic calligraphy are so many, in this study we use the six main styles, plus three extended styles that are derivate from the main styles (Maghribi/Mohakik/Square-Kufi). There are still more styles that no one has used yet, so it would be so interesting to investigate new styles.

3. **Model Adaptation:** the new proposed descriptor was inspired from the Arabic Calligraphy characteristics. Extending the dataset and adding more styles can effect on the recognition rate, for that we need to adapt our descriptor to the new styles to give a better performance.

4. **Use the deep learning:** the newest solutions today are mostly based on deep learning; only few works have used the deep learning for Arabic calligraphy. Our next step is to use a new model or adapt an existing model for the Arabic calligraphy style recognition.

Bibliography

Bibliography

[1] B. Khaldi and M. L. Kherfi, “Modified integrative color intensity co-occurrence matrix for texture image representation,” *JEI*, vol. 25, no. 5, p. 053007, Sep. 2016, doi: 10.1117/1.JEI.25.5.053007.

[2] B. Khaldi, O. Aiadi, and K. M. Lamine, “Image representation using complete multi-texton histogram,” *Multimed Tools Appl*, vol. 79, no. 11–12, pp. 8267–8285, Mar. 2020, doi: 10.1007/s11042-019-08350-1.

[3] L. Hamrouni, M. L. Kherfi, O. Aiadi, and A. Benbelghit, “Plant Leaves Recognition Based on a Hierarchical One-Class Learning Scheme with Convolutional Auto-Encoder and Siamese Neural Network,” *Symmetry*, vol. 13, no. 9, Art. no. 9, Sep. 2021, doi: 10.3390/sym13091705.

[4] L. Hamrouni, R. Bensaci, M. L. Kherfi, B. Khaldi, and O. Aiadi, “Automatic Recognition of Plant Leaves Using Parallel Combination of Classifiers,” in *Computational Intelligence and Its Applications*, Cham, 2018, pp. 597–606. doi: 10.1007/978-3-319-89743-1_51.

[5] A. Korichi, O. Aiadi, B. Khaldi, S. Slatnia, and M. L. Kherfi, “Off-line Arabic handwriting recognition system based on ML-LPQ and classifiers combination,” in *2018 International Conference on Signal, Image, Vision and their Applications (SIVA)*, Nov. 2018, pp. 1–6. doi: 10.1109/SIVA.2018.8661031.

[6] “ThesisFouadSlimane.pdf.” Accessed: Jun. 04, 2021. [Online]. Available: https://diuf.unifr.ch/main/diva/APTI/ThesisFouadSlimane.pdf

[7] S. Hamida, B. Cherradi, O. Terrada, A. Raihani, H. Ouajji, and S. Laghmati, “A Novel Feature Extraction System for Cursive Word Vocabulary Recognition using Local Features Descriptors and Gabor Filter,” in *2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet)*, Marrakech, Morocco, Sep. 2020, pp. 1–7. doi: 10.1109/CommNet49926.2020.9199642.

[8] A. Qaroush, A. Awad, M. Modallal, and M. Ziq, “Segmentation-based, omnifont printed Arabic character recognition without font identification,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, Part A, pp. 3025–3039, Jun. 2022, doi: 10.1016/j.jksuci.2020.10.001.

[9] A. A. Chandio, M. Leghari, M. Leghari, and A. H. Jalbani, “Multi-Font and Multi-Size Printed Sindhi Character Recognition using Convolutional Neural Networks,” *Pakistan Journal of Engineering and Applied Sciences*, Oct. 2019, Accessed: Jun. 21, 2022. [Online]. Available: https://journal.uet.edu.pk/ojs_old/index.php/pjeas/article/view/1635

[10] S. J. Attiya, “Calligraphy, Its historical roots and its philosophical and aesthetic impact: الخط العربي، جذورهُ التاريخية وأثرهُ الفلسفي والجمالي,” *AJSRP*, vol. 8, no. 1, Art. no. 1, Mar. 2022, doi: 10.26389/AJSRP.D190921.

[11] B. Bataineh, S. N. H. S. Abdullah, and K. Omar, “A novel statistical feature extraction method for textual images: Optical font recognition,” *Expert Systems with Applications*, vol. 39, no. 5, pp. 5470–5477, 2012, doi: 10.1016/j.eswa.2011.11.078.

[12] B. Bataineh, S. N. H. S. Abdullah, K. Omar, and A. Batayneh, “Arabic-Jawi Scripts Font Recognition Using First-Order Edge Direction Matrix,” in *International Multi-Conference on Artificial Intelligence Technology*, 2013, pp. 27–38.

[13] B. Bataineh, S. N. H. S. Abdullah, and K. Omar, “Arabic calligraphy recognition based on binarization methods and degraded images,” *Proceedings of the 2011 International Conference on Pattern Analysis and Intelligent Robotics, ICPAIR 2011*, vol. 1, no. JULY, pp. 65–70, 2011, doi: 10.1109/ICPAIR.2011.5976913.

[14] B. Bataineh, S. N. H. Sheikh Abdullah, and K. Omar, “Generating an Arabic Calligraphy Text Blocks for Global Texture Analysis,” *International Journal on Advanced Science, Engineering and Information Technology*, vol. 1, no. 2, p. 150, 2011, doi: 10.18517/ijaseit.1.2.33.

[15] M. A. Talab, S. N. H. S. Abdullah, and M. H. A. Razalan, “Edge direction matrixes-based local binary patterns descriptor for invariant pattern recognition,” *2013 International Conference on*

Soft Computing and Pattern Recognition, SoCPaR 2013, no. November, pp. 13–18, 2013, doi: 10.1109/SOCPAR.2013.7054123.
[16] M. S. Azmi, K. Omar, M. F. Nasrudin, K. Wan Mohd Ghazali, and A. Abdullah, "Arabic calligraphy identification for Digital Jawi Paleography using triangle blocks," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, Jul. 2011, pp. 1–5. doi: 10.1109/ICEEI.2011.6021785.
[17] M. S. Azmi, K. Omar, M. F. Nasrudin, A. K. Muda, and A. Abdullah, "Arabic calligraphy classification using triangle model for Digital Jawi Paleography analysis," *Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems, HIS 2011*, no. July, pp. 704–708, 2011, doi: 10.1109/HIS.2011.6122194.
[18] K. Adam, S. Al-Maadeed, and A. Bouridane, "based classification of Arabic scripts style in ancient Arabic manuscripts: Preliminary results," in *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, 2017, pp. 95–98.
[19] S. R. Allaf and R. Al-Hmouz, "Automatic Recognition of Artistic Arabic Calligraphy Types," *Journal of King Abdulaziz University*, vol. 27, no. 1, pp. 3–17, 2016.
[20] R. Al-Hmouz, "Deep learning autoencoder approach: Automatic recognition of artistic Arabic calligraphy types," *Kuwait Journal of Science*, vol. 47, no. 3, 2020.
[21] M. Khayyat and L. Elrefaei, "A Deep Learning Based Prediction of Arabic Manuscripts Handwriting Style," *IAJIT*, vol. 17, no. 5, pp. 702–712, Sep. 2020, doi: 10.34028/iajit/17/5/3.
[22] F. K. Jaiem and M. Kherallah, "A novel Arabic font recognition system based on texture feature and dynamic training," *IJISTA*, vol. 16, no. 4, p. 289, 2017, doi: 10.1504/IJISTA.2017.088053.
[23] F. Slimane, S. Kanoun, J. Hennebert, A. M. Alimi, and R. Ingold, "A study on font-family and font-size recognition applied to Arabic word images at ultra-low resolution," *Pattern Recognition Letters*, vol. 34, no. 2, pp. 209–218, 2013, doi: 10.1016/j.patrec.2012.09.012.
[24] C. Tensmeyer, D. Saunders, and T. Martinez, "Convolutional Neural Networks for Font Classification," *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 1, pp. 985–990, 2017, doi: 10.1109/ICDAR.2017.164.
[25] H. Luqman, S. A. Mahmoud, and S. Awaida, "KAFD Arabic font database," *Pattern Recognition*, vol. 47, no. 6, pp. 2231–2240, 2014, doi: 10.1016/j.patcog.2013.12.012.
[26] M. Lutf, X. You, Y. M. Cheung, and C. L. Philip Chen, "Arabic font recognition based on diacritics features," *Pattern Recognition*, vol. 47, no. 2, pp. 672–684, 2014, doi: 10.1016/j.patcog.2013.07.015.
[27] I. Abuhaiba, "Arabic font recognition based on templates," *The International Arab Journal of Information Technology*, vol. 1, no. 0, pp. 33–39, 2003.
[28] I. S. Abuhaiba, "Arabic font recognition using decision trees built from common words," *Journal of computing and information technology*, vol. 13, no. 3, pp. 211–224, 2005.
[29] F. Kallel, A. Mezghani, S. Kanoun, and M. Kherallah, "Arabic Font Recognition Based on Discret Curvelet Transform," in *International Afro-European Conference for Industrial Advancement*, 2016, pp. 360–369.
[30] S. B. Moussa *et al.*, "Utilisation de nouveaux paramètres à base de fractale pour la discrimination des fontes arabes To cite this version : HAL Id : hal-00112073 discrimination des fontes arabes," 2006.
[31] S. Ben Moussa, A. Zahour, A. Benabdelhafid, and A. M. Alimi, "New features using fractal multi-dimensions for generalized Arabic font recognition," *Pattern Recognition Letters*, vol. 31, no. 5, pp. 361–371, 2010, doi: 10.1016/j.patrec.2009.10.015.
[32] N. Zaghden, S. B. Moussa, and A. M. Alimi, "Reconnaissance des fontes arabes par l'utilisation des dimensions fractales et des ondelettes," 2006.
[33] I. Chaker, M. Harti, H. Qjidah, and R. Benslimane, "Recognition of Arabic characters and fonts," *IJEST*, vol. 2, no. 10, pp. 5959–5960, 2010.
[34] M. Zahedi and S. Eslami, "Farsi/Arabic optical font recognition using SIFT features," *Procedia Computer Science*, vol. 3, pp. 1055–1059, 2011.

[35] A. Nicolaou, F. Slimane, V. Maergner, and M. Liwicki, “Local Binary Patterns for Arabic Optical Font Recognition,” in *2014 11th IAPR International Workshop on Document Analysis Systems*, Tours, France, Apr. 2014, pp. 76–80. doi: 10.1109/DAS.2014.71.

[36] M. A. A. Mousa, M. S. Sayed, and M. I. Abdalla, “An efficient algorithm for Arabic optical font recognition using scale-invariant detector,” *IJDAR*, vol. 18, no. 3, pp. 263–270, Sep. 2015, doi: 10.1007/s10032-015-0248-9.

[37] Y. Pourasad, H. Hassibi, and A. Ghorbani, “Farsi Font Face Recognition in Letter Level,” *Procedia Technology*, vol. 1, pp. 378–384, 2012, doi: 10.1016/j.protcy.2012.02.078.

[38] A. Bozkurt, P. Duygulu, and A. E. Cetin, “Classifying fonts and calligraphy styles using complex wavelet transform,” *Signal, Image and Video Processing*, vol. 9, no. 1, pp. 225–234, 2015, doi: 10.1007/s11760-015-0795-z.

[39] H. Izakian, S. A. Monadjemi, B. T. Ladani, and K. Zamanifar, “Multi-Font Farsi/Arabic Isolated Character Recognition Using Chain Codes,” vol. 2, no. 7, p. 4, 2008.

[40] G. Sakr, A. Mhanna, and R. Demerjian, “Convolution Neural Networks for Arabic Font Recognition,” in *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, Sorrento, Italy, Nov. 2019, pp. 128–133. doi: 10.1109/SITIS.2019.00031.

[41] A. K. Hussein, “Fast learning neural network based on texture for Arabic calligraphy identification,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, Art. no. 3, Mar. 2021, doi: 10.11591/ijeecs.v21.i3.pp1794-1799.

[42] Z. Kaoudja, M. L. Kherfi, and B. Khaldi, “An efficient multiple-classifier system for Arabic calligraphy style recognition,” 2019. doi: 10.1109/ICNAS.2019.8807829.

[43] R. Ramanathan, K. P. Soman, L. Thaneshwaran, V. Viknesh, T. Arunkumar, and P. Yuvaraj, “A Novel Technique for English Font Recognition Using Support Vector Machines,” in *2009 International Conference on Advances in Recent Technologies in Communication and Computing*, Oct. 2009, pp. 766–769. doi: 10.1109/ARTCom.2009.89.

[44] I. B. Yosef, K. Kedem, I. Dinstein, M. Beit-Arie, and E. Engel, “Classification of Hebrew calligraphic handwriting styles: preliminary results,” in *First International Workshop on Document Image Analysis for Libraries, 2004. Proceedings.*, Jan. 2004, pp. 299–305. doi: 10.1109/DIAL.2004.1263258.

[45] Yong Zhu, Tieniu Tan, and Yunhong Wang, “Font recognition based on global texture analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1192–1200, Oct. 2001, doi: 10.1109/34.954608.

[46] X. Zhang and G. Nagy, “Computational method for calligraphic style representation and classification,” *Journal of Electronic Imaging*, vol. 24, no. 5, p. 053003, 2015, doi: 10.1117/1.jei.24.5.053003.

[47] S. Allaf and R. Al-Hmouz, “Automatic Recognition of Artistic Arabic Calligraphy Types,” *Journal of King Abdulaziz University-Engineering Sciences*, vol. 27, no. 1, 2016, doi: 10.4197/Eng.27-1.1.

[48] M. S. Azmi, K. Omar, M. F. Nasrudin, A. K. Muda, and A. Abdullah, “Arabic calligraphy classification using triangle model for Digital Jawi Paleography analysis,” *Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems, HIS 2011*, no. July, pp. 704–708, 2011, doi: 10.1109/HIS.2011.6122194.

[49] K. Adam, S. Al-Maadeed, and A. Bouridane, “Letter-based classification of Arabic scripts style in ancient Arabic manuscripts: Preliminary results,” in *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, Apr. 2017, pp. 95–98. doi: 10.1109/ASAR.2017.8067767.

[50] “Slimane et Vanoirbeek - 2018 - Multifont and Multisize Low-resolution Arabic Text.pdf.” Accessed: Apr. 05, 2021. [Online]. Available: https://www.researchgate.net/profile/Fouad-Slimane/publication/323514683_Multifont_and_Multisize_Low-resolution_Arabic_Text_Recognition_Using_APTI_Database_Benchmarking_State-of-the-Art_Systems/links/5b1e64d1a6fdcca67b69b16c/Multifont-and-Multisize-Low-resolution-Arabic-Text-Recognition-Using-APTI-Database-Benchmarking-State-of-the-Art-Systems.pdf

[51] H. Luqman, S. A. Mahmoud, and S. Awaida, “KAFD Arabic font database,” *Pattern Recognition*, vol. 47, no. 6, pp. 2231–2240, 2014, doi: 10.1016/j.patcog.2013.12.012.

[52] O. Zayene, J. Hennebert, S. M. Touj, R. Ingold, and N. E. B. Amara, “A dataset for Arabic text detection, tracking and recognition in news videos- AcTiV,” in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, Aug. 2015, pp. 996–1000. doi: 10.1109/ICDAR.2015.7333911.

[53] A. Al-Thubaity and A. Al-Subaie, “Effect of word segmentation on Arabic text classification,” in *2015 International Conference on Asian Language Processing (IALP)*, Oct. 2015, pp. 127–131. doi: 10.1109/IALP.2015.7451548.

[54] M. Lutf, X. You, Y. M. Cheung, and C. L. Philip Chen, “Arabic font recognition based on diacritics features,” *Pattern Recognition*, vol. 47, no. 2, pp. 672–684, 2014, doi: 10.1016/j.patcog.2013.07.015.

[55] M. Alghamdi and W. Teahan, “Printed Arabic script recognition: A survey,” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 9, pp. 415–428, 2018, doi: 10.14569/ijacsa.2018.090953.

[56] Y. Hannad, I. Siddiqi, and M. E. Y. El Kettani, “Writer identification using texture descriptors of handwritten fragments,” *Expert Systems with Applications*, vol. 47, pp. 14–22, Apr. 2016, doi: 10.1016/j.eswa.2015.11.002.

[57] M. Mohandes, M. Deriche, and S. O. Aliyu, “Classifiers Combination Techniques: A Comprehensive Review,” *IEEE Access*, vol. 6, pp. 19626–19639, 2018, doi: 10.1109/ACCESS.2018.2813079.

[58] J. Park, E. Lee, Y. Kim, I. Kang, H. I. Koo, and N. I. Cho, “Multi-Lingual Optical Character Recognition System Using the Reinforcement Learning of Character Segmenter,” *IEEE Access*, vol. 8, pp. 174437–174448, 2020, doi: 10.1109/ACCESS.2020.3025769.

[59] F. Alaei, A. Alaei, U. Pal, and M. Blumenstein, “A comparative study of different texture features for document image retrieval,” *Expert Systems with Applications*, vol. 121, pp. 97–114, 2019, doi: 10.1016/j.eswa.2018.12.007.

[60] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan. 1979, doi: 10.1109/TSMC.1979.4310076.

[61] V. Ojansivu and J. Heikkilä, “Blur insensitive texture classification using local phase quantization,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5099 LNCS, pp. 236–243, 2008, doi: 10.1007/978-3-540-69905-7_27.

[62] R. Andonie, “Extreme Data Mining: Inference from Small Datasets,” *INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL*, vol. 5, no. 3, Art. no. 3, Sep. 2010, doi: 10.15837/ijccc.2010.3.2481.

[63] “12_Inter_Ency_Soc_8446.pdf.” Accessed: Apr. 05, 2021. [Online]. Available: http://www.law.harvard.edu/faculty/shavell/pdf/12_Inter_Ency_Soc_8446.pdf

[64] B. Bataineh, S. N. H. S. Abdullah, and K. Omar, “A novel statistical feature extraction method for textual images: Optical font recognition,” *Expert Systems with Applications*, vol. 39, no. 5, pp. 5470–5477, Apr. 2012, doi: 10.1016/j.eswa.2011.11.078.

[65] Z. Kaoudja, B. Khaldi, and M. L. Kherfi, “Arabic Artistic Script Style Identification Using Texture Descriptors.,” in *020 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP)*, May 2020, pp. 113–118. doi: 10.1109/CCSSP49278.2020.9151569.

[66] H. Zhao and H. Liu, “Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition,” *Granular Computing*, vol. 5, no. 3, pp. 411–418, 2020, doi: 10.1007/s41066-019-00158-6.

[67] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, “Online and offline handwritten Chinese character recognition: Benchmarking on new databases,” *Pattern Recognition*, vol. 46, no. 1, pp. 155–162, Jan. 2013, doi: 10.1016/j.patcog.2012.06.021.

[68] A. Lawgali, “A Survey on Arabic Character Recognition,” *IJSIP*, vol. 8, no. 2, pp. 401–426, Feb. 2015, doi: 10.14257/ijsip.2015.8.2.37.

[69] M. R. Hashemi, O. Fatemi, and R. Safavi, “Persian cursive script recognition,” in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, Aug. 1995, vol. 2, pp. 869–873 vol.2. doi: 10.1109/ICDAR.1995.602039.

[70] U. G. Mangai, S. Samanta, S. Das, and P. R. Chowdhury, “A Survey of Decision Fusion and Feature Fusion Strategies for Pattern Classification,” *IETE Technical Review*, vol. 27, no. 4, pp. 293–307, Jul. 2010, doi: 10.4103/0256-4602.64604.

[71] “Version soumise.” Accessed: May 01, 2021. [Online]. Available: https://http.cs.berkeley.edu/projects/vision/papers/efros-iccv99.pdf

[72] B. Khaldi, O. Aiadi, and M. L. Kherfi, “Combining colour and grey-level co-occurrence matrix features: a comparative study,” *IET Image Processing*, vol. 13, no. 9, pp. 1401–1410, May 2019, doi: 10.1049/iet-ipr.2018.6440.

[73] S. Ghosh, D. Lahiri, S. Bhowmik, E. Kavallieratou, and R. Sarkar, “Text/Non-Text Separation from Handwritten Document Images Using LBP Based Features: An Empirical Study,” *Journal of Imaging*, vol. 4, no. 4, Art. no. 4, Apr. 2018, doi: 10.3390/jimaging4040057.

[74] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural Features for Image Classification,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973, doi: 10.1109/TSMC.1973.4309314.

[75] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Gray scale and rotation invariant texture classification with local binary patterns,” in *Proceedings of the Sixth European Conference on Computer Vision (eccv2000*, 2000, pp. 404–420.

[76] J. Chen *et al.*, “WLD: A Robust Local Image Descriptor,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1705–1720, Sep. 2010, doi: 10.1109/TPAMI.2009.155.

[77] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, Jun. 2005, vol. 1, pp. 886–893 vol. 1. doi: 10.1109/CVPR.2005.177.

[78] J. Kannala and E. Rahtu, “BSIF: Binarized statistical image features,” *Proceedings - International Conference on Pattern Recognition*, pp. 1363–1366, 2012.

[79] R. P. Sharma and S. Dey, “Quality Analysis of Fingerprint Images Using Local Phase Quantization,” in *Computer Analysis of Images and Patterns*, Cham, 2019, pp. 648–658. doi: 10.1007/978-3-030-29888-3_53.

[80] S. R. Arashloo, J. Kittler, and W. Christmas, “Face Spoofing Detection Based on Multiple Descriptor Fusion Using Multiscale Dynamic Binarized Statistical Image Features,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2396–2407, Nov. 2015, doi: 10.1109/TIFS.2015.2458700.

[81] U. Scherhag, D. Budhrani, M. Gomez-Barrero, and C. Busch, “Detecting Morphed Face Images Using Facial Landmarks,” in *Image and Signal Processing*, Cham, 2018, pp. 444–452. doi: 10.1007/978-3-319-94211-7_48.

[82] Y. Wang *et al.*, “Polyp classification by Weber’s Law as texture descriptor for clinical colonoscopy,” in *Medical Imaging 2019: Computer-Aided Diagnosis*, Mar. 2019, vol. 10950, p. 109502V. doi: 10.1117/12.2512756.

[83] D. Srivastava, B. Rajitha, S. Agarwal, and S. Singh, “Pattern-based image retrieval using GLCM,” *Neural Comput & Applic*, vol. 32, no. 15, pp. 10819–10832, Aug. 2020, doi: 10.1007/s00521-018-3611-1.

[84] Z. Kaoudja, M. L. Kherfi, and B. Khaldi, “A New Computational Method for Arabic Calligraphy Style Representation and Classification,” *Applied Sciences*, vol. 11, no. 11, Art. no. 11, Jan. 2021, doi: 10.3390/app11114852.

[85] G. M. Binmakhashen and S. A. Mahmoud, “Document Layout Analysis: A Comprehensive Survey,” *ACM Comput. Surv.*, vol. 52, no. 6, p. 109:1-109:36, Oct. 2019, doi: 10.1145/3355610.

[86] M. Stauffer, P. Maergner, A. Fischer, and K. Riesen, "A Survey of State of the Art Methods Employed in the Offline Signature Verification Process," in *New Trends in Business Information Systems and Technology: Digital Innovation and Digital Business Transformation*, R. Dornberger, Ed. Cham: Springer International Publishing, 2021, pp. 17–30. doi: 10.1007/978-3-030-48332-6_2.

[87] A. Rehman, S. Naz, and M. I. Razzak, "Writer identification using machine learning approaches: a comprehensive review," *Multimed Tools Appl*, vol. 78, no. 8, pp. 10889–10931, Apr. 2019, doi: 10.1007/s11042-018-6577-1.

[88] H. Luqman, S. A. Mahmoud, and S. Awaida, "Arabic and Farsi Font Recognition: Survey," *International Journal of Pattern Recognition and Artificial Intelligence*, Jan. 2015, doi: 10.1142/S021800141553002X.

[89] I. B. Yosef, K. Kedem, I. Dinstein, M. Beit-Arie, and E. Engel, "Classification of Hebrew calligraphic handwriting styles: preliminary results," in *First International Workshop on Document Image Analysis for Libraries, 2004. Proceedings.*, Jan. 2004, pp. 299–305. doi: 10.1109/DIAL.2004.1263258.

[90] L. Wang *et al.*, "Artistic features extraction from chinese calligraphy works via regional guided filter with reference image," *Multimed Tools Appl*, vol. 77, no. 3, pp. 2973–2990, Feb. 2018, doi: 10.1007/s11042-017-4688-8.

[91] G. Pengcheng, G. Gang, W. Jiangqin, and W. Baogang, "Chinese calligraphic style representation for recognition," *IJDAR*, vol. 20, no. 1, pp. 59–68, Mar. 2017, doi: 10.1007/s10032-016-0277-z.

[92] F. K. Jaiem, F. Slimane, and M. Kherallah, "Arabic font recognition system applied to different text entity level analysis," *2017 International Conference on Smart, Monitored and Controlled Cities, SM2C 2017*, pp. 36–40, 2017, doi: 10.1109/SM2C.2017.8071847.

[93] I. M., S. Hamdy, and M. G., "Deep Arabic Font Family and Font Size Recognition," *IJCA*, vol. 176, no. 4, pp. 1–6, Oct. 2017, doi: 10.5120/ijca2017915589.

[94] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[95] M. Cheriet, N. Kharma, C.-L. Liu, and C. Suen, *Character recognition systems: a guide for students and practitioners*. John Wiley & Sons, 2007.

[96] Ming-Kuei Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, Feb. 1962, doi: 10.1109/TIT.1962.1057692.

[97] M. A. Talab, S. N. H. S. Abdullah, and M. H. A. Razalan, "Edge direction matrixes-based local binary patterns descriptor for invariant pattern recognition," *2013 International Conference on Soft Computing and Pattern Recognition, SoCPaR 2013*, no. November, pp. 13–18, 2013, doi: 10.1109/SOCPAR.2013.7054123.