KUVEMPU UNIVERSITY

MACHINE LEARNING APPROACHES FOR HANDWRITTEN CHARACTER RECOGNITION SYSTEM: A CASE STUDY FOR ARABIC LANGUAGE

A Thesis
Submitted for the degree of

DOCTOR OF PHILOSOPHY

In the faculty of Science and Technology

By

OMAR ALI BORAIK SALIM BORAIK

Research Supervisor

Dr. RAVIKUMAR. M

DEPARTMENT OF P.G STUDIES AND RESEARCH IN COMPUTER SCIENCE
KUVEMPU UNIVERSITY, SHANKARAGHATTA-577 451,
SHIVAMOGGA, KARNATAKA, INDIA

January - 2023

KUVEMPU UNIVERSITY

MACHINE LEARNING APPROACHES FOR HANDWRITTEN CHARACTER RECOGNITION SYSTEM: A CASE STUDY FOR ARABIC LANGUAGE

A Thesis
Submitted for the degree of

DOCTOR OF PHILOSOPHY

In the faculty of Science and Technology

By

OMAR ALI BORAIK SALIM BORAIK

Research Supervisor

Dr. RAVIKUMAR. M

DEPARTMENT OF P.G STUDIES AND RESEARCH IN COMPUTER SCIENCE
KUVEMPU UNIVERSITY, SHANKARAGHATTA-577 451,
SHIVAMOGGA, KARNATAKA, INDIA

January - 2023

CERTIFICATE

This is to certify that **Mr. Omar Ali Boraik Salim Boraik** has worked under my supervision for his Ph.D. thesis entitled **“Machine Learning Approaches for Handwritten Character Recognition System: A Case Study for Arabic Language”.** I also certify that the work is original and has not been submitted to any other university wholly or in part for any other degree.

11/01/23

Dr. Ravikumar M

Research Supervisor
Department of P. G. Studies and Research in Computer Science
Kuvempu University, Shankaraghatta – 577 451, Shivamogga, INDIA

DECLARATION

I hereby declare that the work embodied in this Doctoral thesis has been carried out by me at the Department of P. G. Studies in Computer Science, Kuvempu University, Shankaraghatta, under the supervision of **Dr. Ravikumar M.** This thesis has not been submitted in part or full for the award of any diploma or degree of this any other University.

Omar Ali Boraik Salim Boraik

Research Scholar
Department of P. G. Studies and Research in Computer Science
Kuvempu University, Shankaraghatta – 577 451, Shivamogga, INDIA

DEDICATED TO

My parents, family members, friends, and all my teachers for their encouragement love, and support.....

Acknowledgments

First, I thank the Almighty Allah who is being with me and showering his blessings and grace towards me in all walks of my life. Later, I am extremely grateful to my Parents, Jameela Faraj Ali Saeed and Ali Boraik Salim Boraik, for their love, prayers, caring and sacrifices for educating and preparing me for my future. I am very much thankful to my brother Hussein Ali Boraik, nephews Abdullah and Miqdad, nieces Fatimah and Retaj, and uncles Saeed, Hadi, Obaid, Salim, Abdullah, Saleh and Ali for their direct or indirect encouragement, continuing support and prayers towards achieving my goals. I am forever indebted to my wife for understanding my tough goals and difficulties in reaching this stage. Her support towards my research has made me in-valuable in my life. Also, my lovely two sons, Mohammad and Mustafa, whom I missed so much during my Ph.D.

Then, I would like to thank those who made this thesis possible particularly my supervisor, **Dr. Ravikumar M.** for his valuable guidance at every stage in this thesis. I must acknowledge my indebtedness to him for his untiring support, insights, motivation, time, trust to be a part of his Ph.D. Scholars team under his guidance, and last for his humane and fatherly approach which helped me greatly with the research work. He has been such a candid Guide to me. His methodical acumen and the ability to give importance to the right things at the right time would always be helpful to me in my life. His constant support, affection and encouragement at every stage made my task much more manageable. I owe my most tremendous gratitude to him.

I wish to express my profound gratitude, devotion and respectful regards to **Sri. Yogish Naik G.R**, Chairperson, Department of Computer Science, Kuvempu University, Shankaraghatta. He has been such a humble assistant to me. His constant support, affection and encouragement made my task much easier.

I take immense pleasure to express my sincere and deep sense of gratitude to **Dr. Prabhakar C.J., and Dr. Suresha M.** for their kindness, cooperation, encouragement, care and concern shown during my research at the time of their presence in the department.

I am highly grateful to **Prof. Riaz Mahmood,** Director of the International Students' Cell for his immense help and sincere support during the tenure of my research work. I will remain indebted to him for his unremitting care, concern, encouragement, and kindness.

I would express my deep sense of gratitude to faculty members of the Department of Computer Science, Kuvempu University, for extending their continuous help and support in the department.

My special words of thanks to **Dr. Hasib Daowd Esmail Al-ariki**, associated Professors Department of computer Science Sana’a Community College, Sana’a, Republic of Yemen, for his support, help, advice and encouragement during the tenure of my research work.

I would like to remember with special love and highly gratitude, the pleasant moment I had with my research group **Mr. G. Shivakumar, Mr. M.C. Prashanth, and Mr. S Sampath, Rachana P G** for their timely help, memorable moments spent with them and support throughout my research work. I also thank all my research fiends **Raghukumar D S., Kuppa S, Divyarani R., shrinivasa S R, Vidyasagar K B., Vidyasagar S D and Narendra R**. A special thanks to my partner in lab, registration, synopsis and final synopsis,

Dr. Shivaprasad B J. I will never forget his state, "we came together, studied together, then we left," and that is what happened.

I would also like to thank the Staff members of Examination, SC/ST Development cell, Academic and Finance Section of Kuvempu University for their help.

At the outset of my research, I faced many problems, but my brother and very close friend to me **Dr. Mohammed Ali Bladram** tamed the difficulties, and his humanitarian stance played a major role in completing the admission procedures, not only in the academic situation even in my personal life. He was always my backup and support in any matter I face, without him, I could not start my Ph. D. Programme. I offer my deep thanks, praise, love, and appreciation. I am deeply indebted to my dear friend **Dr. Ghassan Qaid Al-Maqtari** for his assistance, care, concern, and cooperation given to me with love, not for me alone but to my friends too.

I extend my sincere thanks to all Yemeni friends in India who left a fingerprint in my memory including**, Dr. Mohammed Abdullah Bajiri, Mr. Hussein Al-Aidaros, Dr. BakeeL Rezeq Battah, Dr. Hesham Abdo Aqlan**, **Mr. Mugaahed Abdu Kiad**, **Dr**. **Mufeed Ahmed Naji Saif and Abdullah Moaath.**

And all who are responsible for making this dream come true and for their blessing, understanding, patience and moral support and encouragement. Without them, my work would not have seen the light of day.

Omar Ali Boraik Salim Boraik

List of Figures

List of Tables

Contents

Abstract

Recognition of handwritten Arabic characters is gradually becoming popular in the field of research due to its many critical applications like historical document processing, digital signature verification, postal address reading, check verification in banks and helping visually impaired people. Concerning automation, written character detection becomes difficult, especially in cursive writing. Its recognition is a challenging deliberating task due to several reasons like the shape resemblance of characters, different handwriting styles, different ways of writing the same character, the existence of dots or diacritical marks above or below the letters, and variations in strokes and complexity of compound characters. The writing style is also impacted by the writers' mental and physical fitness.

In this research work, we constructed a new database. This database is a handwritten Arabic text collected from native writers, in addition to historical manuscripts, text documents and letters. Then, the collected images are scanned and stored as images in different formats (JPG, PNG). In addition to other available databases, this database, such as KHAAT and HACD datasets, has been fed into the proposed system for Arabic handwritten character recognition. This research work was divided into three stages: pre-processing stage in order to enhance the poor images. Segmentation stage, the segmentation algorithm of this research work divides the segmentation stage into three levels: segment a text page into lines, find the word boundaries of each text line, and then find and segment each cursive word into characters. The algorithm produces a novel matrix consisting of segmentation points and their position status for each text line. Characters separate the output of the segmentation stage with their position status. Finally, the outcomes of the segmentation stage are fed character by character to the recognition stage. Finally, the classification and recognition stage.

We proposed novel stage-wised efficient models to classify and recognize Handwritten Arabic Characters. In the first stage, Convolutional Neural Network (CNN) model is implemented to identify Arabic handwritten characters. The model consists of a stack of Five Convolution2D-BatchNormalization-ReLU-MaxPooling2D layers. The transfer learning methodology is used in the next stage and the model is trained using the bottleneck characteristics of a pre-trained network of Inception v3 and ResNet 50 methods. Finally, the models are fine-tuned on top of the pre-trained network to achieve better performance. We achieved promising results with the fine-tuned model. Several Convolutional Neural Network models like VGG16, ResNet 50, Inception v3 and DenseNet 121 are also explored on our newly created dataset as well as standard datasets. Isolated Handwritten Arabic Characters collected from AHCD and our proposed database containing Arabic characters in their four shapes has been built to provide normalized data used for training systems to recognize unconstrained handwritten Arabic text. The success rate of the proposed approach is 98.94% for the Arabic handwritten character classification and 98% for recognition.

Chapter 1

Prologue

1.1 Preamble

Nowadays, people have become very dependent on computer and modern technology in processing huge amounts of data in order to manage and organize their work and life in preserving their heritage, assistance in education and transfer of knowledge, and uses in the medical, economic, political and military fields in the first place. In the past, mankind used to read and write because reading and writing are the basis of civilization and their Live's affairs. Culture, civilization, literature, heritage and history of previous nations are only a result of what they recorded and reached us as manuscription, books and documents, etc. By which they left behind them. With huge and unexpected changes in the development of technology, it has become a great demand for the rapid entry of huge amounts of information, whether printed, carved or handwritten. This data was entered into computers by human operators using traditional methods at the end of the 20th century. These input methods took a long time and effort with more errors. The discovery of artificial intelligence, pattern recognition and the development of optical character recognition (OCR) systems which read a text through recognition techniques help to convert this text into digital data in a short time with more accuracy and few errors.

Despite the rapid and significant progress of optical character recognition systems in converting printed and handwritten texts into digital ones, these systems are still uncompleted, especially in recognizing handwriting in various languages (Mohamed

Cheriet, 2007). There is also an urgent need to do a lot of research works to develop and enable the machine to read printed and handwritten documents. To process and recognize them faster, more accurately and comprehensively for types of fonts and different languages. The process of converting printed or handwritten texts into digital which are subject to edit, modify and use in other applications called optical character recognition (OCR) [Somaya Al-ma'adeed, 2004, Fakhr, M. W, 2011].

OCR is a general term used to describe technologies that have the ability to recognize text in scanned documents and images to convert them into a digital format. OCR technology is also used to convert all types of images that contain typed and handwritten text into machine-readable text data. Over the last two decades, OCR technology has become one of the main areas of interest in the implementation of projects related to the digitization of old documents such as newspapers, manuscripts, bills, letters, constitutional documents, etc. The significance of OCR techniques has become more widespread with the advent of the Internet, which is a source of multilingual information based on digital text data.

The advancements in technology and artificial intelligence in the world in recent decades, the OCR system of different languages has become an interest of studies in the field of document image processing. On the other hand, Arabic optical character recognition system is one of these fields which is based on two different types: On-line and Off-line. (Figure 1.1) Automatic off-line recognition of text can be divided into recognition of printed and handwritten characters. Off-line Arabic handwriting recognition still faces big challenges. Even though there are a lot of studies that investigated the recognition of handwritten languages such as Latin through the use of various techniques. Little works have been conducted to shed light on Arabic handwritten recognition. However, no specific technique of these studies has worked as a significant output with high accuracy in the aspect of practical application to recognize Arabic handwriting characters. It is challenging to recognize Arabic letters of scanned documents due to the difference in people's handwriting and other factors such as the different shapes of Arabic letters, ligature between letters as well as components overlap. Arabic language is highly distinguished from Roman-based languages with many characteristics like the way of writing, from right to left, regardless of the similarity with them in writing numbers' shapes (e.g., from left to right).

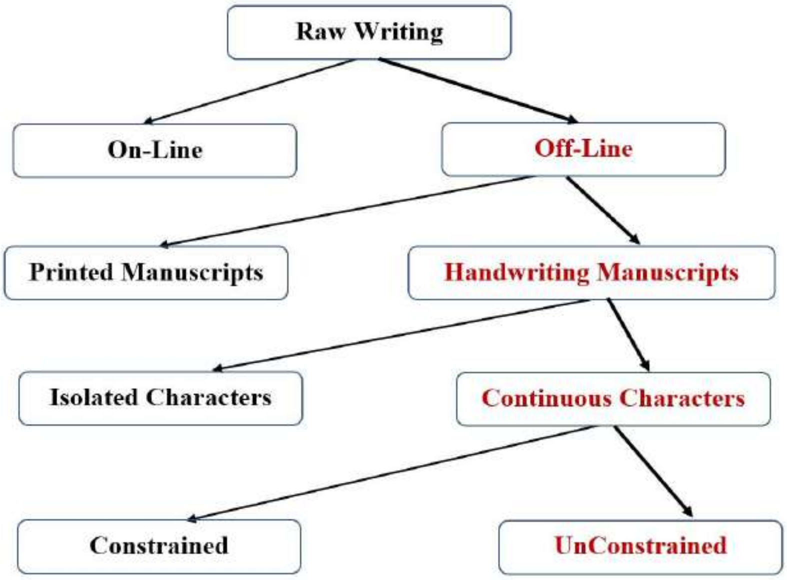


Figure 1.1: Types of Character Recognition.

The study of Arabic OCR started in the 1970s. In 1975 the first research work on Arabic OCR was published to start a new direction of works in Arabic OCR. In the 1990s they developed a new system of Arabic OCR and were practically ready to use (Chen, J et al., 2010). The recognition of Arabic handwriting improvement is depending on solving the challenges. Regardless of the efforts that have been made in the past, there is still a lack of outcomes in Arabic Handwritten Character Recognition system compared to OCR of texts of other scripts [Tharwat et al., 2015, Chen, J et al., 2010].

When researchers reach a technique that gives better accuracy in Arabic OCR, it would provide a good opportunity for using Arabic digital alphabet in other applications as well as other languages which use Arabic alphabet or similarities to its alphabet characteristics (Tharwat et al., 2015).

However, when the process of recognizing scripts, in Arabic particularly, many challenges are hard to overcome such as different handwriting of the same writer at different times. Also, unlimited differences in human handwriting styles, similarities of distinct character shapes, character overlaps and interconnections of neighbouring characters. In addition, the style of writing can be affected by the mood of the writer.

Handwriting recognition approaches are divided into two categories: the first category is the handcrafted approach and the second one is the unsupervised/supervised learning approach. The first one uses methods widely methods such as Scale-Invariant Feature Transforms (SIFT) [Elzobi et al., 2012, Elleuch et al., 2015, Zaiz, F. et al., 2016] and Gabor features [Pechwitz, M. et al., 2002, Amrouch, M. et al., 2017]. the second catenary, known as the type of deep learning approaches, has obtained its reputation for solving many problems of computer vision. the applications of these approaches show better outcomes and notable results than traditional methods in the field of handwriting recognition (Al-Huri, Ibrahim, 2015). Many studies worked on many methods to conduct the process of recognizing Arabic handwriting documents, these documents contain words, letters, punctuation, shapes or numbers, which can be classified into groups by using Support Vector Machine classifier (SVM), Naive Bayes, K-Nearest Neighbours, (KNN) Decision Tree, Random Forest, Hidden

Markov Models [Robert A. Cote, 2009, Mahmood Abbas, 2000]. The classification stage is performed after pre-processing stage which is involving greyscale, various filters, Binarization, Morphological Operations and Normalization. The next stage is line/ word/ character segmentation which is including many methods like Projection Profile, Contour Tracing, Morphological Operation, Template Matching, Hidden Markov Models (HMM), implicit segmentation [Qaroush, Aziz, et al., 2019, Choudhary, Amit.2014]. These studies are applied to many Databases such as IFN/ENIT, KHATT, AHDB, HFT.

In addition, (Amrouch, et al., 2017) have two different studies focused on learning features using the Convolutional Neural Network (CNN) and handcrafted features. (Amrouch, M. et al., 2017) their experiments were applied on IFN/ENIT database and compared the results between these studies. This study concluded that the obtained results by using the CNN features excelled over those achieved results obtained by using the handcrafted features. This chapter introduces an overview of Arabic handwriting and challenges which researchers faced in Arabic OCR field. It also mentions remarkable reasons behind the backwardness of Arabic Handwritten Character Recognition system. In other words, this chapter is a brief history of the development of Arabic handwriting throughout phases. It will also state the objectives, motivation and publications related to this research work/ this chapter also will give a brief summary of the entire chapters of the thesis.

1.2 Arabic Language

Arabic language is one of the Semitic language family which such including Aramaic, Phoenician, Amharic, Hebrew, etc. Arabic language associated with Islam and Arabic is the language Glorious of Qura’an. It is considered one of the most widely spoken language in the world. According to United Nations classification of World Languages Rank, Arabic language occupies the sixth rank (Al-Huri, Ibrahim, 2015). It is the official language in all Arab countries. In addition, some African countries such as Mali, Chad, Eritrea and Senegal use Arabic language as a second language. According to the University of Arizona report, more than 400 million speak Arabic language (Robert A. Cote, 2009).

Arabic alphabetical order contains 28 letters and each letter has numerous characteristics. Arabic writing style, unlike English language, is from right to left and the letters of each word are cursive, not separated. Each letter has more than two shapes based on its position in the word. It has one shape when it is written isolating and other shapes when it is written at the beginning, middle or end of the word. So, some letters have four or more shapes. Based on that Arabic alphabetical order can expand to 84 structural forms. Table 1.1 shows the different forms of Arabic letters depending on their place in the word.

1.2.1 Characteristics of Arabic Alphabet

- Advantages of Arabic letters can be formed in any geometrical shape with keeping their origin and essence, So Arabic style has a lot of variant fonts such as Nasekh, Roqa’a, Qoufi and so on. (Figure. 1.2)
- The names of the Arabic letters depend on their pronunciation and exit from the speaker’s throat or mouth. For example, the letter "أ " is pronounced with a’a. “ب” is ‘Baa’, etc. while in English, the letter “X” is pronounced with e first then ‘X’, “H” too, “L” and so on.

- Each Arabic letter gives only one sound, while in English letter may give two sounds like ‘C’ and some two English letters give one sound only such ‘C, K,’ ‘f, ph’.
- Arabic letters are written based on their pronunciation, no silent letter in Arabic, like the silent letter ‘و’ in ‘عمرو’. On the other hand, some English letters are written but silent like ‘Bought’ two letters ‘gh’ are silent ‘Neighborhood’ [Mahmood Abbas, 2000, Adel Al-Aloosi, 2008, Salah Al-Deen Al-Mongid, 1979].
- The letters of each word are cursive in Arabic language, but some of them cannot be cursive such as “ـا، ـو، ـد، ـذ، ـو،ـر، ـز” (Adel Al-Aloosi, 2008).
- Each Arabic letter has 2-4 shapes depending on its location in the word, it has one shape when it comes alone, and has another shape/s when it comes at the beginning, middle or end of the word. Another notice in some letters such as ‘هـ’ when it comes at the end of the word, it may come ‘ـه’ or it may change into another letter ‘ـة’ as shown in the words ‘فاطمة’ and ‘ماله’ (Adel Al-Aloosi, 2008).
- Diacritical Marks: Diacritics are small symbols that are placed above or below Arabic letters to indicate vowels, helping the readers to pronounce a word correctly, i.e., the tone of the voice change when they pronounce the letter because the sound level gives the word completely different meaning. For example, the word “جَـنَّـةَ” “Jannta”, “جُـنَّـةَ” “Jonnta” and “جِـنَّـةَ” “Jinnta”, the first word means “Heaven”, the second one means “Prevention” and the last means “Fairy”. Usually, Diacritical Marks are not used in informal writing because they have clear meanings throughout the context of the sentences. Table 1.2 shows the Diacritical Marks and their locations.

- Ligature is a character formed by combining two or more letters in an accepted manner. (table 1.3 shows that).
- Some letters have similar shapes with a little different which is in number and position of dots like ‘ب ، ت ، ث’ , ‘حـ ، خـ جـ’ , ‘د ، ذ’ , ‘ر ، ز’ , ‘س ، ش’, ‘ص ، ض’ , ‘ط ، ظ’ , ‘عـ ، غـ’, ‘فـ ، قـ’ .
- Some letters start with a closed-loop when they are written, such as the letters ‘ص ، ض ، ط ظ ، ف ، قـ ، و’ and the type of style writing (fonts) has a role in showing or concealing these circles.

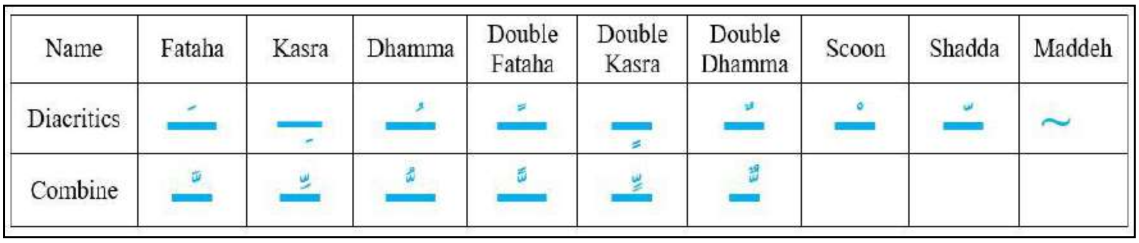
Moreover, there are many non-Arabic-speaking languages that use the Arabic alphabet in writing, including Kurdish, Tajik, Persian, and Azeri. More than 20 languages use the Arabic alphabet.

Table 1.1: Arabic Alphabet with their shapes based on the position.

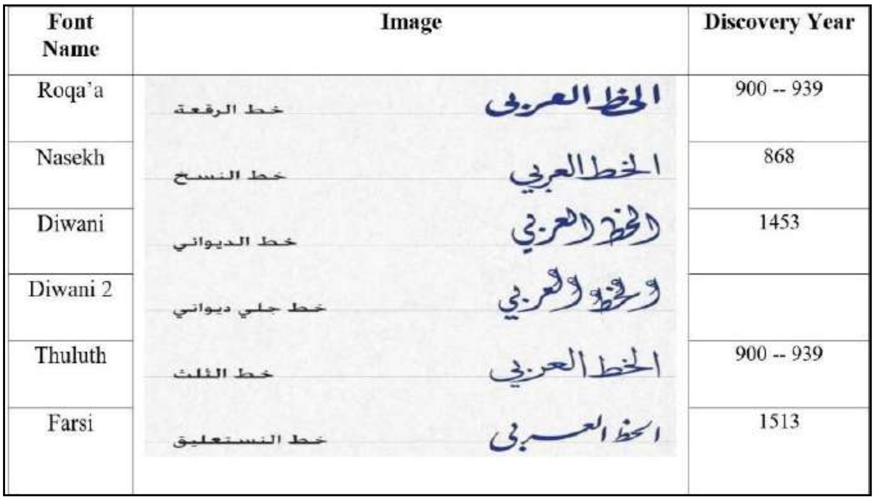
No	Name	Isolated	Start	Middle	End
1	Alif	ا			ـا
2	Ba	ب	بـ	ـبـ	ـب
3	Ta	ت	تـ	ـتـ	ـت / ـة
4	Tha	ث	ثـ	ـثـ	ـث
5	Jeem	ج	جـ	ـجـ	ـج
6	Hha	ح	حـ	ـحـ	ـح
7	Kha	خ	خـ	ـخـ	ـخ
8	Dal	د			
9	Thal	ذ			
10	Raa	ر			
11	Zay	ز			
12	Seen	س	سـ	ـسـ	ـس
13	Sheen	ش	شـ	ـشـ	ـش
14	Saad	ص	صـ	ـصـ	ـص
15	Dhad	ض	ضـ	ـضـ	ـض
16	Tta	ط	طـ	ـطـ	ـط
17	Zha	ظ	ظـ	ـظـ	ـظ
18	Ain	ع	عـ	ـعـ	ـع
19	Gain	غ	غـ	ـغـ	ـغ
20	Faa	ف	فـ	ـفـ	ـف
21	Qaaf	ق	قـ	ـقـ	ـق

22	Kaaf	ك	كـ	ـكـ	ـك
23	Lam	ل	لـ	ـلـ	ـل
24	Meem	م	مـ	ـمـ	ـم
25	Noon	ن	نـ	ـنـ	ـن
26	Haa	هـ	هـ	ـهـ	ـه
27	Waw	و		ـو	
28	Yaa	ي	يـ	ـيـ	ـي

Table 1.2: Arabic Diacritical Markings



Name	Fataha	Kasra	Dhamma	Double Fataha	Double Kasra	Double Dhamma	Scoon	Shadda	Maddeh
Diacritics	ـَ	ـِ	ـُ	ـً	ـٍ	ـٌ	ـْ	ـّ	~
Combine	ـَّ	ـِّ	ـُّ	ـًّ	ـٍّ	ـٌّ			



Font Name	Image	Discovery Year
Roqa'a	خط الرقعة الخط العربى	900 -- 939
Nasekh	خط النسخ الخط العربي	868
Diwani	خط الديواني الخط العربي	1453
Diwani 2	خط جلي ديواني والخط العربي	
Thuluth	خط الثلث الخط العربي	900 -- 939
Farsi	خط النستعليق الخط العربى	1513

Figure 1.2: Various Common of Arabic Fonts

Table 1.3: Example of Arabic Ligatures

Ligature	**No ligature**
محمود	محمود
الحليم	الحليم
كجم	كجم

1.3 Brief history on Arabic Handwriting

Arabic script had gone through many stages of developments, evolution, progression and improvements. Linguists present many views about the evolution of Arabic handwriting (Saleh Ibrahim AL-Hassan, 2003). There are several points of view in this regard (Adel Al-Aloosi, 2008), but from the stone inscriptions and monuments discovered by Archaeologists and linguists, it was found that Arabic alphabet has appeared since 250 AD (Anno Domini) (Mahmood Abbas, 2000), where the letters were written without dots and diacritical marks and were written separately too [Al-Jabori Kamel Suliman, 2000, Al-Mongid Salah Al-Deen, 1979].

The original Arabic alphabet was a derivative of the Nabataean alphabets where they both have similar characteristics and some features as well. The Nabataeans are Arabic tribes who inhabited Syria, Jordan, Northern Arabia, and the Sinai Peninsula. Due to trade movements and commercial exchange, the Arabic script spread in Arabic regions that overlapped with foreigners because of the relationship between Arabic language and the Islamic religion, the differences in accent and tone appeared in Arabic language, then linguists were afraid of the damage that could deteriorate Arabic language, and because many characters have similar forms like "Ba" "Ta" "Tha".etc., making confusion to distinguish the difference among of them, so they set the language grammars and writing standards, and put small dots on or under the letters to distinguish between similar letters (Al-Mongid Salah Al-Deen, 1979), it was done primarily by Zeyad (a governor of Iraq) who directed Abu Alasouad Aldoaly at 686 AD, Al-Jahedh in "Al_Amsaar" (Countries) 649, mentioned that Naser Bin Asem is the first person who did that. Alkhaleel Bin Ahmad (718 – 786), created diacritical marks to help pronounce the Arabic words correctly (Ahmed Abdelmajid Khalifa, 2015).

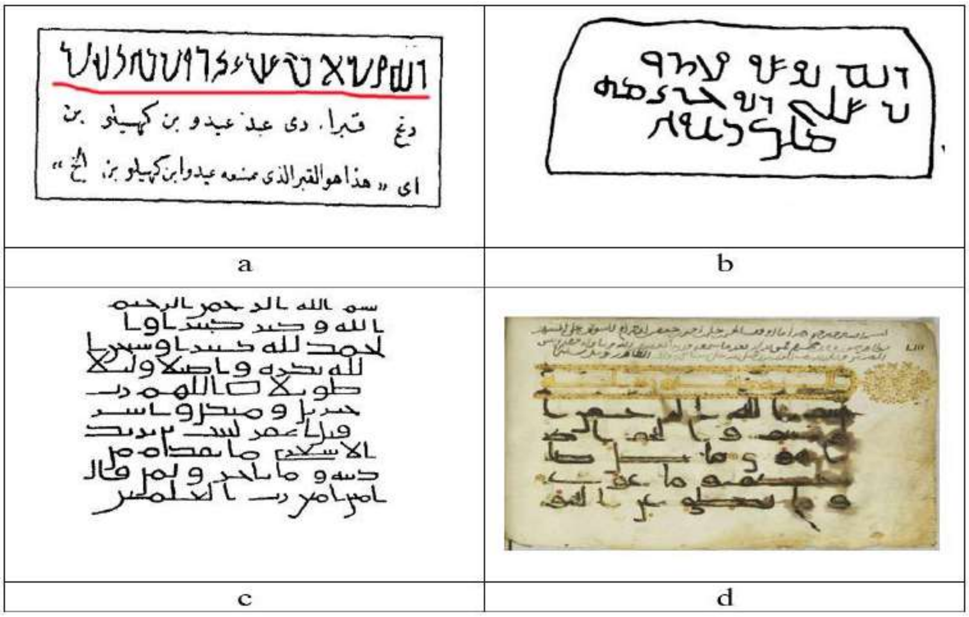


Figure 1.3: (a) A sample of Nabataean manuscript engraved on stone at 9 BC (Before Christ).

(b): Nabataean stone Inscription at 250 AD, found in Jourdan.

(c): Arabic stone Inscription in Karbala, Iraq at 368 AD, the carving was before discover the dots.

(d): Arabic manuscript after discovered the dots at 686 AD. It is kept in Metropolitan Museum, USA.

1.4 Literature Review of Arabic OCR

Since the mid-1940s, there has been extensive research and publications on optical character recognition, most of the published works were on Latin characters, while research on the recognition of Japanese, Chinese and various Indian languages published in the mid-sixties, in spite of nearly a billion people in the world use the Arabic alphabets in writing (such as Urdu, Farsi, Kurdish, etc.), but the research on Arabic optical characters recognition began in the seventies of last century, where the first published work was in 1975, and the first system in Arabic OCR was available in 1990. Research and studies in Arabic OCR especially in handwritten are still considered few and limited [Fakhr, 2011), Patel & Jha, 2015], due to the several following causes, are: 1- Insufficient journals, books, conferences, funding and interaction between researchers (there is a small percentage). - 2 - Lack of resources and tools such as datasets for Arabic scripts, lexicons and programming tools. - 3- The delayed start of text identification in Arabic language. - 4 - The techniques were improved for other scripts could not be successfully applied to the recognition of Arabic script because of the unique characteristics of Arabic script. -5- most of the researchers move away from Arabic recognition field because of the extremely complicated features of Arabic letters, which are represented in the different shapes of each letter based on its place in the word. The letters are written cursively, dots are above or below some of letters, which is considered a part of the letter’s body. Diacritical marks.

1.4.1 Arabic Printed Text Recognition

Previous works and studies have achieved great success in recognizing the printed Arabic scripts with the errors rate being very low because the printed scripts are characterized in the stability of the gaps between lines and words, there is no overlapping, touching or wavy in the lines. Skewness is global of the scripts in existence, if any, the font size and font style are uniform. The document images are uncomplicated and unambiguous compared to document images for handwriting. Among the most used techniques and achieved success, such as projection profile, Hidden Markov models, Artificial Neural Networks, and various convolutional neural networks. Here, the most important Previous work on text recognition in Arabic has been the most significant. will be reviewed.

Study (Yamina et al., 2017) " Printed Arabic Optical Character Recognition using Support vector machine ", has discussed the accuracy of classification based on Features Extraction where it depends on segmentation. The results showed in isolated Arabic characters, the Character Recognition Rate was 99.08%, and in machine-printed Arabic text was 95.03%.

(Batawi & Abulnaja, 2012) made an experimental Study in their article entitled "*Accuracy Evaluation of Arabic Optical Character Recognition Voting Technique*" using a new technique called voting technique to evaluate four accuracy types: accuracy all, accuracy ~ , accuracy !, accuracy ~!.

Study (Al-Muhtaseb et al., 2008) "*Recognition of printed Arabic text using Hidden Markov Models*" in which they provided a system to recognize printed Arabic text using Hidden Markov Models with the aid of the algorithm that segment the text lines into connected parts then into characters. Therefore, the experimental results have been achieved a rate of recognition accuracy reached (94%).

(Naz et al., 2014) presented in their thesis "Challenges *in Baseline Detection of Arabic Script Based Languages"* baseline detection challenges for Arabic script based on languages and targeted Nastaliq and Naskh writing style. Baseline is an important step in the OCR as it directly affects the rest of the steps and increases the performance and efficiency of character segmentation and feature extraction in OCR process. Also, they provided a comprehensive review of baseline detection methods for Urdu language.

(Besbas et al.; "Improved Method for Sliding Window Printed Arabic OCR," 2015) "*Improved Method for Sliding Window Printed Arabic OCR*" have helped to optimize the Method for recognizing printed Arabic characters. It is separation-free character recognition. Reference characters are selected for recognition by sliding a window with the size of the reference characters. In its study, they developed a database for Arabic characters with images that they use for reference. Five font styles and nine font sizes were used, as well as their different positions in a word (isolated, beginning, middle, and end). were accomplished.

(Slim KANOUN, 2000) "Script Identification for Arabic and Latin Printed and Handwritten Documents" Based on the difference in nature of Arabic and Latin documents; printed or handwritten, it is proposed an approach to discriminate between them. In their approach, two steps are taken: one is morphological, which is applied at the block level of text, and a second is geometric, which is applied at the line level of text. They found 88.5 % of the times that they correctly identified 400 big blocks scanned text and 92 % of the times when dealing with 335 small blocks of scanned text. They have been still optimizing and improving their system to make it along the following principal axes:

1- Provide a combination of features as needed to ensure that printed and handwritten texts don't become confused and that Arabic and handwritten Latin don't make sense to each other.
2- By employing neural classifiers along with K-nearest neighbour (KNN) algorithm, better results should be obtained.
3- Build Test on large datasets of representative samples.

(Mousa et al., 2017) "*Arabic Character Segmentation Using Projection-Based Approach with Profile's Amplitude Filter*" discussed the challenges in the segmentation process which must be carried out to determine the character's start and end, so they presented Arabic Character Recognition and proposed algorithm using projection-based approach to identify separate line, words, characters with profile’s amplitude filters and a line, word, and character segmentation using their proposed algorithm yielded promising results.

(Shamsan et al., 2017) "Off-Line Arabic Handwritten Character Using Neural Network" discussed some distinctions Arabic handwrote from various styles of writing in Asian and Latin scripts, which complicate the recognition and classification processes. As a result, neural networks were proved to be a viable concept and were considered to be the most successful method of handwriting recognition, particularly in Arabic or any other complex writing.

(Sahlol et al., 2017) “*Arabic Handwritten Characters Recognition System, Towards Improving its Accuracy”*, Based on the GREY WOLF optimization system, a system has been proposed for the recognition of Arabic handwritten characters. Preprocessing, feature extraction, and classification were the steps of the OCR process. Their proposed system utilizes well-known optimization algorithms such as Bat (BAT), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Grey Wolf Optimization (GWO).

(Qaroush et al., 2022) proposed a method for indirect words and characters segmentation from different independent fonts. The proposed method was carried out in two levels. The first level was based on the baseline method to locate the lines to separation, then save the separated lines in independent images as an output. Then take theses output individually words segmentation by using vertical projection profile technique in addition to Interquartile Range (IQR) technique in order to distinguish the gaps between words and gaps in sub-words. The second level is to recognize the correct separation points from all the prospect points.

1.4.2 Arabic Handwritten Text Recognition

(Fadeel, 2016) “*An Efficient segmentation algorithms for Arabic handwritten characters recognition systems"* proposed a solution, When a handwritten text is not segmented correctly, poor recognition will certainly occur., depending on diacritics removal, segmentation of (word/ sub word, ascender characters, descenders characters and embedded characters) techniques.

(Sahlol & Suen, 2014) studied in their paper "*A Novel Method for the Recognition of Isolated Handwritten Arabic Characters*", The commonalities of different letter shapes, the interconnections of surrounding characters, and their position in the word are all examples of the infinite variation in human handwriting. They achieved accurate results reached to 99%.

(Benjelil et al., 2009) discussed in their research paper “*Arabic and Latin script identification in printed and handwritten types Based on Steerable Pyramid Features*”, there are still confusion issues between Arabic and Latin Handwritten scripts that must be resolved, using *K* nearest neighbor classifier technique.

(Mezghani et al., 2014) “*Identification of Arabic/French Handwritten/Printed Words using GMM - Based system”*, they have made a comparison between Arabic and French languages to identify their text either handwritten, printed and discrimination problems between them in Automatic document text recognition is a challenge that should be solved. The evaluation of script category likelihoods is depending on Gaussian Mixture Models (GMM).

(Aburas & Gumah, 2008), have a research paper entitled "*Arabic Handwriting Recognition: Challenges and Solutions*". They have been showing the difficulties and challenges facing on recognition to Arabic character recognition and suggested many solutions.

1- Depending on Characters are pre-segmented.

2- depending on segmenting words into elementary words.

3- relied on segmenting words into isolated characters.

4- Identified words before segmentation.

5- Requires no segmentation and is based on recognition.

(Haraty, 2004) "*Arabic Text Recognition* ", presented a new division for the classification process of handwritten Arabic text. There are three main components: a heuristic algorithm for extracting character features from an image, two generalized feed-forward networks to discover the optimal output, and a classifier algorithm to determine which character corresponds to each index value.

(Fadeel, 2016) "*An Efficient Segmentation Algorithm for Arabic Handwritten Characters Recognition System*" has developed and provided techniques in segmentation stage. The segmentation stage is considered the most important among the process stages where he introduced a dependable segmentation method for Arabic handwritten documents.

The study (Kessentini et al., 2010) have discussed "Multi-*Stream Markov Models for Arabic"* the unconstrained off-line Arabic handwritten word recognition system based on multi-stream segmentation free HMM. The proposed approach uses low-level feature sets and does not explicitly split handwriting into graphemes.

(Sari et al., 2002), entitled *"Off-line Handwritten Arabic Character Segmentation Algorithm: ACSA*," in which they showed the difficulties and challenges that occur in many OCR systems that require pre-processing to perform character recognition. In character segmentation, precisely as a result of improperly segmented characters, they cannot be recognized accurately because they are poorly segmented. Cursive script poses the most incredible difficulty in character segmentation.

The study (Charfi et al., 2012) *"A New Approach for Arabic Handwritten Postal Addresses Recognition*". They suggested an analytical approach for automatically recognizing Arabic handwritten postal addresses using the beta elliptical model. The system was separated into three stages: analysis, pre-processing, and classification.

In the study (Farooq et al., 2005) entitled "*Pre-processing Methods for Handwritten Arabic Documents"* provided pre-processing methods for images of Arabic handwritten documents: Pre-processing methods are used to increase manuscripts' readability and automate recognition. In addition to the standard noise reduction and filtering processes, these procedures include text normalizing techniques such as baseline correction, slant normalization, and skew correction.

(Abuhaiba et al., 1994), "*Recognition of Handwritten Cursive Arabic Characters" Presented* an offline character recognition system for handwritten cursive Arabic characters. They created a noise-independent method that produces skeletons that reflect the structural relationships of the character components. The character skeleton is transformed into a tree structure appropriate for recognition.

Deep Learning Techniques (Convolution Neural Network (CNN)) by (El-Sawy et al., 2017), (AlJarrah et al., 2021), in Arabic Handwriting Character Recognition. Both studies used AHCD database, which contains 16800 images of hand letters. The database was divided into 13,440 images for training the proposed model (80%) and

3360 images for testing (20%), the success rate reached 97.7% for classifying Arabic handwritten characters.

Moreover, in study (Shams et al., 2020), Deep Convolution Neural Networks (DCNN) were applied along with Support Vector Machine (SVM) for recognizing the Arabic Handwritten Character. The characters were extracted and classified using the DCNN architecture.

Most of the problems and difficulties that still concern researchers in solving them as much as possible in the scope of Optical Character recognition systems for Arabic are the overlapping, touching challenges. The researcher (Ullah et al., 2019) proposed a system for touching character segmentation from Arabic Handwritten text which includes these challenges. Overlapping set theory and contour tracing methods are used to find out the touching point. Morphological Operations are also used to detect the endpoints of Arabic words. After locating all touching points, the trace boundaries of separation are applied to trace the boundaries of each character till the touching point to extract it.

Characters are used through the segmentation method to separate the characters which are touching, (Ullah et al., 2019). The proposed system achieved the success of good and promising results, which were tested on 220 images of Arabic handwritten text, 214 images were segmented correctly and well-segmented, the error rate was 2.17%. While, (Aouadi et al., 2013), proposed a method to segment touched lines or words in Arabic handwritten text, they utilized a shape context and thin-plate spline to find the segmentation through transformation and recognition stages.

From the literature review in the recognition systems for either printed or Handwritten Arabic text, it is noticed that the systems for recognizing a text have made wider progress in achieving the purpose of recognition [Qaroush et al., 2022, Alghamdi & Teahan, 2018, Alginahi, 2010, Lawgali, 2015a, Qaroush et al., 2020, Ravi and Khan, 2013, Padmavathy and Priya, 2018, Ahmad, 2013, Boukharouba, 2017, Mohammed and Soora, 2018, Al-Shatnawi and Omar, 2009, Dave, 2015].

1.5 Arabic Handwritten Character Recognition system

Text of Arabic handwritten is complex for machinery recognition. In the research field, a very large amount of research efforts was invested to increase the rate of handwritten character recognition in Arabic using various methods and techniques. In 2002, they developed a system to recognize Arabic characters which/ while using the neural network for classification. They used a set of unchanging equations and symbols (Awel et al., 2019). In a study (Lamghari, N, et al., 2019) performed their methods on a database that contains 34000 images of isolated characters of Arabic handwritten. By using Feed-Forward Neural Network, 70% of these images used if they were allocated for training and 15% for testing while 15% for validation. The accuracy rate reached 98%. In (El-Sawy, et al., 2017), the authors used the Convolution Neural Network (CNN) method to recognize Arabic handwritten characters. They created a new special database which includes 16800 images of Arabic handwriting texts. This database was divided into two sections; 80% for training and 20% for testing. The result of their experiment achieved 5.1% misclassification. This gave an impression that the recognition of the handwritten character process is in progress. According to (Altwaijry, Najwa and Isra Al-Turaiki, 2021) the author used the same method which led to 97% accuracy and 88% by applying CNN method on AHCD database.

As usual, any Optical Character Recognition system goes through several stages, starting with the pre-processing stage to improve the input document. Next, the segmentation stage which is considered the heart of the OCR system. Finally, recognition of the Arabic handwritten characters (post-processing).

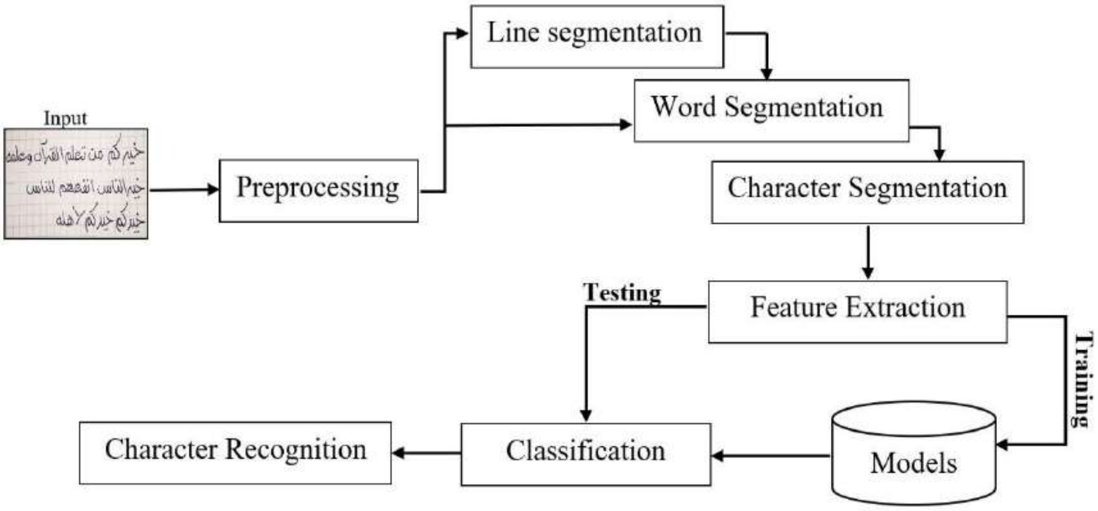


Figure 1.4: Stages of Handwritten Character Recognition

1.5.1 Preprocessing Stage

The input image to the recognition system is almost accompanied by poor quality, noises and global or partial slant problems. These problems reduced the chances of success in the post-processing stage. This stage is the first step for enhancing and preparing the target image for the next stages in order to reach satisfactory results. The more successful of the stage, the more successful the recognition system. The techniques used in this stage depend on the purpose of classification. Handwriting recognition differs from the techniques in recognizing printed text. In this work/ process/ technique, preprocessing techniques of improving images (recognition) for handwriting will be discussed, starting with binarization technique (convert the input image from RGP to Grayscale then to Black/ white) by using Otsu (Ahmed, et al., 2016). Then, it removes the noises, small dots and diacritical marks. In these research works [Ahmed, et al., 2016, Boukerma, et al., 2012] Arabic diacritical marks are returned after the recognition process to their position. While, in character recognition these marks will be removed. The dots above or below the characters must be kept because they are considered part of the character's body. Using some filters to improve the image pixel values such as Gaussian Filter, Median filter, Low Pass, also Gabor filters. Then, correcting the slope of the document either global or partial. Dilation and erosion methods are used to expand the objects of image to help in the line/word segmentation individually and then return them to the original structure after the segmentation process. Due to Arabic writing being cursive, it is better to use the thinning, open and close methods to find a gap among the connected characters to segment and extract these characters (Boukerma, et al., 2012).

1.5.2 Segmentation stage

"Segmentation is a process of partitioning a digital image into multiple regions and extracting the meaningful region which is known as Region of Interest" (Arunachalam, T, 2017). This stage is considered the most important stage in the OCR system through which it achieves a higher success of rate. It is also related to the preprocessing stage. The input images are different in contents, as they contain single or several lines, or even one word. In this step, the role of segmentation is to process the image and split the entire document into lines or words individually. Then, it focuses on the main goal of split the word into characters. The segmentation process of handwriting is more complicated than the printed one because it contains several challenges such as the overlap between the lines and the inclination of the line. Often, it occurs outside the basic horizontal line, and in the spaces between words are usually not equal in Arabic handwriting. The type of font (style) in handwriting is not restricted too. However, the research works in the OCR field led to a notable success of the segmentation process at high rates and the continuous attempts helped to overcome challenges. Meanwhile, the machine learning and deep learning approaches have achieved great success along with their application with a very large database [Qaroush, Aziz, et al., 2019, Choudhary, Amit. 2014]. This stage goes through three parts as shown in figure 1.5.

1.5.2.1 Line Segmentation

In this stage, most of the segmentation methods have been investigated. It showed satisfactory success and it overcame the challenges such as overlap among lines, correcting the slant of each line by increasing the thickness of the line. It also segmented the lines by using the Neighbour Connected Components Method (Khandelwal, et al. 2009), but the boundary box around the line was applied by Brodic et al., 2013, while (Bahaghighat et al., 2012) used the base line for each line. In (Arivazhagan et al., 2007), they suggested the statistical method. They used horizontal/ vertical projection methods for the line segmentation, and Hough transform technique (Louloudis, Georgios, et al., 2009).

1.5.2.2 Word Segmentation

After the success of extracting the lines separately, the next part is word segmentation. The most important feature for separating and extracting words is the gap among the words. There are many challenges in Arabic handwriting facing word segmentation process such as overlapping in one word, and some Arabic letters cannot connect with the next character. Therefore, in some cases, the space between the characters of the same word is greater than the space between two words which cause errors in the segmentation and thus getting misclassification.

(El-Hajj, et al., 2005) have proposed a system to recognize Arabic handwritten documents proposed with no need for lines segmentation, as the method of identifying hidden Markov models was adopted in all stages of the training and testing process, extracting features based on the baseline, then detecting the two lower and upper sub lines of the words. The words were extracted using the horizontal projection method. Their experiment achieved a success rate of 86.51%. (Al-Hajj et al., 2007) also presented an analytical method for recognizing Arabic handwritten characters by using two stages for character recognition without the segmentation.

(Pak, et al., 2018), the authors used bounding box technique for word segmentation directly and extract after removing the diacritical marks to prevent (avoid) errors in word region detection to draw its borders. In (Elzobi, et al., 2011) applied word/ sub-word segmentation based on connected components approach after applying morphological operations and also through which characters were recognized individually. It was noticed that it is better for the techniques to be hybrid for word and character segmentation because many techniques succeed with a high accuracy rate when they are performed on a particular database, Moreover, machine learning algorithms can segment words without line segmentation from the whole document as shown in Figure 1.6.

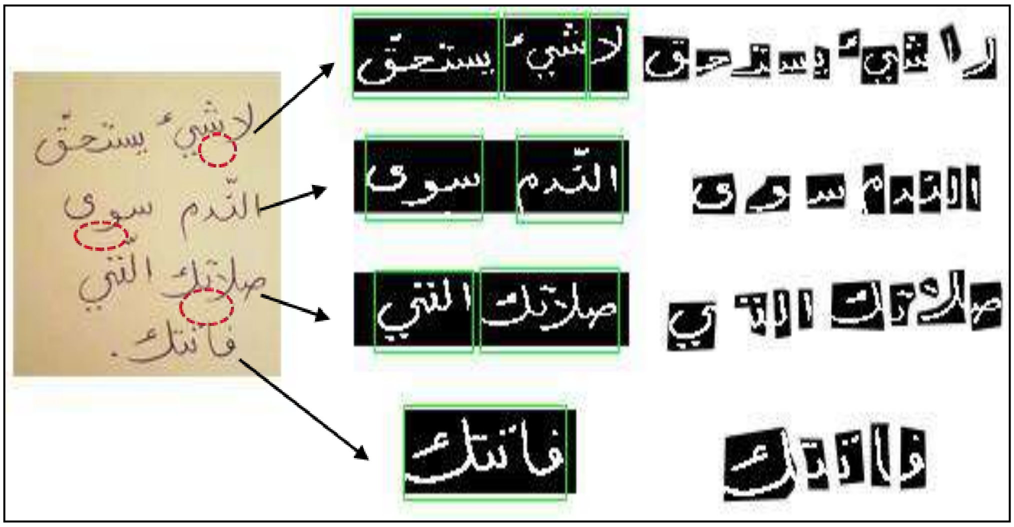


Figure 1.5: Line, Word or Character segmentation. Red circles are overlapping

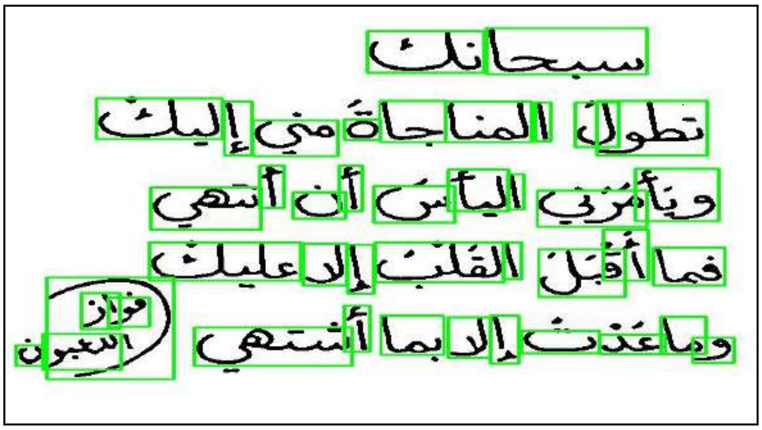


Figure 1.6: Word or Sub-word segmentation without line Segmentation

1.5.2.3 Character segmentation

This part is the most important and the main aim of the segmentation process and extracting characters from a word. The most difficult challenge is to determine the critical points for each character, where its beginning and its end are (Shaikh, et al., 2008). One of the characteristics of Arabic characters is that some of them are connected (joint) with only the previous letter, but the others are joint with the previous and next letter as shown in Table 1.1.

1.5.3 Extraction of the Features

This process raises the success and accuracy of the learning process where it reduces the dimensions of the raw data by removing unimportant information and reducing its repetition and thus effectively reducing the amount of irrelevant data. Features Selection chooses sub-set of features from the relevant original features for model construction. It speeds up the training process, streamlining models, prevent overfitting as it improves generalization [Alghamdi and Teahan, 2018, Trier et al., 1996, Alghamdi and Teahan, 2018, Trier et al., 1996].

1.5.4 Classification

The task of classification is to understand, identify, and classify objects into (labelled) categories that have been labelled in supervised learning. In unsupervised learning, clustering is used for dividing the data points into unlabeled groups. This division is based on similar data in characteristics to be in one group, and the dissimilar data should be in different groups.

1.6 Motivation

Technological development and advances in machine learning to Optical Character Recognition in many languages have encouraged the application of study techniques to Arabic scripts. As it is well known, Arabic language is spoken by millions of peoples, it has its place among world common language, it is also important in defining many others' cultures. Through previous studies, the text recognition system in other languages achieved advanced success, in comparison to the Arabic handwritten, the techniques did not reach unified results that are difficult to generalize on the other handwriting document images, because of the unlimited challenges that the most interested of them are summarized as (El-Hajj. 2005): 1- Arabic words are overlapped. 2- The dots which are above or below the characters and diacritical marks as well make the segmentation process more complex. 3- The different forms of each letter according to its position in a word (in the beginning - middle - end or isolated) lead to errors in classification. 4- The handwriting style is, it is usually written with mixed fonts. 5- Ligatures: Two or more letters are joined together appearing as one character. 6- the writing is curved and touching, and 7- Extracting the texts from input script images which include shapes, signatures, horizontal or vertical lines.

1.7 Objectives

The aim of this work is to address the problems and extend the recognition system for handwritten Arabic characters by applying Machine and Deep Learning approaches to improve the recognition of Arabic handwriting. The major objectives are:

1. Develop a robust model for pre-processing Arabic handwritten characters.
2. Construct an algorithm for segmentation of Arabic handwritten characters.
3. Design an efficient algorithm for recognition of handwritten Arabic characters.

1.8 Structure of the thesis

Chapter 1: presenting an introduction to Arabic Optical Character recognition (AOCR) system, describe the available methodologies for recognizing the Arabic text, reviewing machine learning and deep learning techniques in extracting the features to classify, providing a historical overview of the development of Arabic writing throughout history, and reviewing Challenges and the characteristics of the Arabic alphabets, showing the objectives and motivation of this research, Chapter 2: An introduction to the literature review and the stages of improvements of the OCR system for recognizing printed and Handwriting Arabic text, an overview of the research methodology.

Chapter 3: Provides a detailed description of the available Arabic databases. Steps for collecting data and methods for creating a database for this work, in addition to using an available database such as KHAAT, AHCDB databases.

Chapter 4: Suggested Preprocessing techniques for enhancing the input Arabic document images and preparing them for the next stages of AOCR system.

Chapter 5: The different and suggested methods for segmenting the Arabic Handwritten documents into lines, words, and letters.

Chapter 6: Application of machine learning and deep learning algorithms in extracting the features and classifying the Arabic characters, comparative analysis between deep learning architectures and recognizing the classified characters.

Chapter 7: providing the Conclusion, summary and contribution of this work, also suggestions for future work.

Chapter 2

Datasets

2.1 Preamble

An important basic component of OCR is to provide a database for the identification system. Over the past decades, several databases of various Arabic handwritten, printed data and images of Arabic numbers have been created and developed, such as IFN/ENIT and MNIST (Arabic numeral and English letters) databases. These databases are common and the most widely used among researchers (Amara et al., 2005). In the early years of the development of OCR systems, Arabic text recognition system was the last interest of the researchers (Abdalkafor, 2018), it was due to the lack of databases and the complexities of Arabic writing, even though, Arabic language is one of the official languages spread in the world, also, along with the wide use of Arabic alphabets in other languages. The unavailability of databases for these languages in that time such as Urdu, Kurdish, Persian, etc. was an obstacle to improving Arabic text recognition system. But with the first success of the Arabic handwritten recognition system in 1990 [Abdalkafor, 2018, Amara et al., 2005], the interest of researchers increased in applying the system to languages that use the Arabic alphabets, for example in Persian letters (Parhami and Taraghi, 1981), Jawi (Manaf, 1998), Urdu (Pathan et al., 2012) because the characteristics of the letters are similar to some extent Large with Arabic letters, but with a slight difference, such as in the number of letters and the number of dots as shown in figure 2.1, as well as the number of different letters classifications in relation to the distribution of letters to the categories to which it belongs.

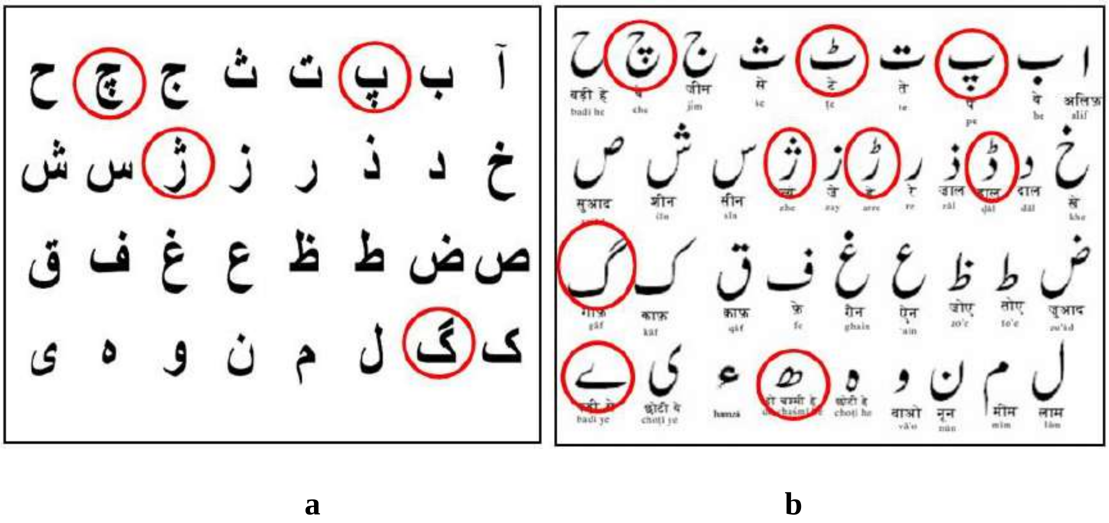


Figure 2.1: (a) Persian Alphabet. (b) Urdu Alphabet. The red circles are marks around the unique letters.

In this chapter, the available Arabic databases are discussed and a review of how to create a database for this research and how to collect random images of Arabic Handwritten documents for test the search on them, as well as using other databases such as AHDB, IFN/ENIT. Then compare the results and find out the success of the proposed system on the randomly entered images.

Many languages where there are few or no databases. Limited like Urdu, Arabic, some Indian languages and many other world languages. Various Arabic databases, some of which are images of printed documents; for instance, the study (Trenkle et al.,1995) used 700 digital pages from 45 printed documents. The study (Amara et al., 2005) used an Arabic database that includes images of printed documents for text phrases, words, independent sub-words and isolated letters. The study (Kharma et al., 1999) a description of an Arabic database which includes words, numbers and signatures. The authors (Soleymani and Razzazi, 2003) presented a database of isolated Arabic and Persian letters. Hamid (Hamid and Haraty, 2001) collected pictures of students' mail addresses, which amounted to 360 addresses; the total words were almost 4000 words. The authors (Al-Ohali et al., 2003) collected mixed images of Arabic words and sub-words. They also collected sample images of Arabic bank checks were written using Indian numbers, as shown in Figure 2.2. The most significant and common Arabic databases used recently will be presented. The most important databases are:

IFN/ENIT database: (IFN: **I**nstitute **F**lur **N**achrichtentechnik. **E**NIT: **E**cole **N**ationale d'**I**ngenicursed **T**unis (Amara et al., 2005; Memon et al., 2020). This dataset is one of the most famous and widely used in the Arabic OCR systems. It was developed in 2002 in cooperation between the Tunisian Laboratory for Signals Systems and the German Institute for Wireless Communications, where they collected images data about Arabic handwriting, the text of the image are the names of Tunisian cities and villages, 411 writers of different age categories participated in writing. This database contains 26,459 images of words and 210,000 letters (Pechwitz et al., 2002). The images were entered by a scanner with a resolution of 300 dpi. Then the input images are converted to the binary system. This database is free to provide the researchers in Arabic OCR area.

CENPARMI Database: This database is images of a bank's checks. It was developed by the CENPARMI Institute in cooperation with "Al Rajhi Banking and Investment Corporation" (one of the famous banking institutions in Saudi Arabia). Almost 7000 images of handwritten checks were collected using Indian numerals as in Figure 2.2 (these numbers are frequently used in the eastern part of the Arab world, while Arabic numerals are used in the western part of the Arab world, such as Libya, Algeria, Tunisia, Morocco). This database is unfree and is marketed for a profit.

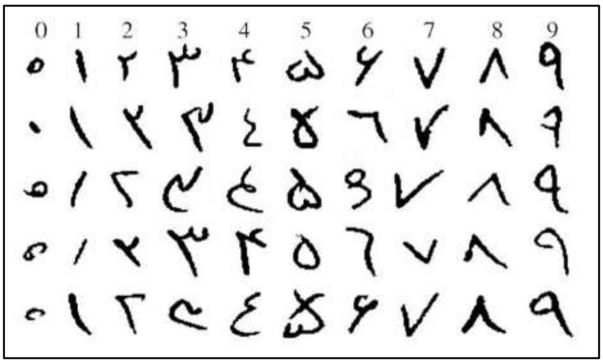


Figure 2.2: Indian-Farsi Digits.

There is another database called by the same name as the previous database called CENPARMI database. It was used by (Suen et al., 1992), it was containing around 17000 samples. the author in (Solimanpour et al., 2006) presented a database where it includes words and isolated letters from the Persian language and also Indian numbers [Pathan et al., 2012, Pechwitz et al., 2002, Soleymani and Razzazi, 2003] and it contains 18,000 images of documents. It was expanded in (Khosravi and Kabir, 2007). In 2009, CENPARMI issued an updated version which included a large number of data images of the Persian text, amounting to 432,357 images (Haghighi et al., 2009).

KHATT Database: abbreviation of the phrase (**K**FUPM **H**andwritten **A**rabic **T**ext) (Mahmoud et al., 2012). This database includes approximately 1,000 samples of Arabic Handwritten documents, it is written by 1,000 randomly selected writers from different age categories and also from different countries. The images were entered by a scanner with a change in resolution 200, 300, 600 dpi. It is developed in 2012 by (Mahmoud et al., 2012). It is free for researchers to access and use in applications.

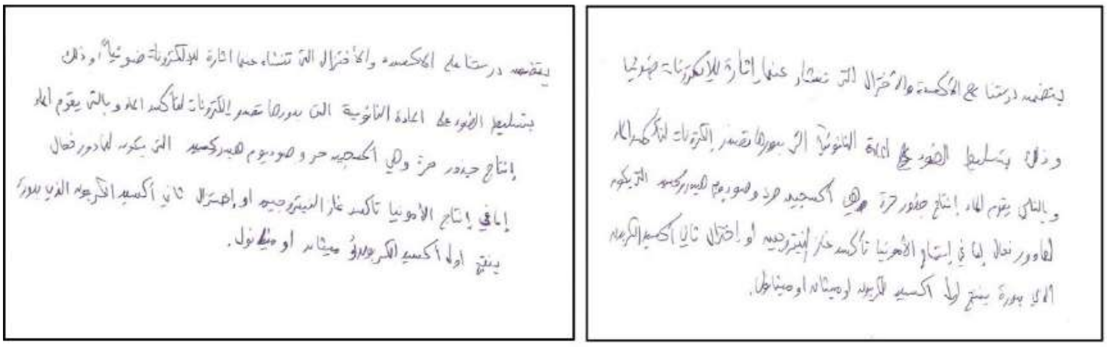
AHDB database: This database includes Arabic words and text sentences, collected from the most common and used words in writing bank checks. It is a data set for 105 volumes which include 305 documents and was shared written by 105 from different age categories. It is also free and available to researchers.

Table 2.1 presents the rest of the other databases, some of them private

Name	Language	No. of Documents	Writers	Accessible
BBN (Saleem et al., 2009)	Arabic	39500	259	proprietary
LDC	Arabic	7447	70	Private, only for registered members
Al Isra (Kharma et al., 1999)	Arabic	37000 words 10,000 Digits 2500 Signatures 500 Sentences	500	Public
Alamri and others (Alamri et al., 2008)	Arabic	11375 Words 21426 Letters 46800 Digits 1640 Symbols 13439 Numerals	328	Upon Request
FHT (Ziaratban et al., 2009)	Persian	1000	250	Upon Request
MADCAT (Al-Hajj et al., 2007)	Arabic	9693 Pages	400	Public
QUWI (Al Maadeed et al., 2012)	Arabic and English	5085	1017	
AD/MAD (El-Sherif and Abdelazeem, 2007)	Arabic	700,000 Digits	700	Upon Request
AWIC2011	Arabic	161	45	Public
PD100 (Helli and Moghaddam, 2008)	Persian	----------	100	Upon Request
AHTD (Abdalkafor et al., 2020)	Arabic and English	1500	100	--------
AHDB/FTR (Ramdan et al., 2013)	Arabic	497 Words images	100	Freely

2.2. Collecting dataset and Database Designing

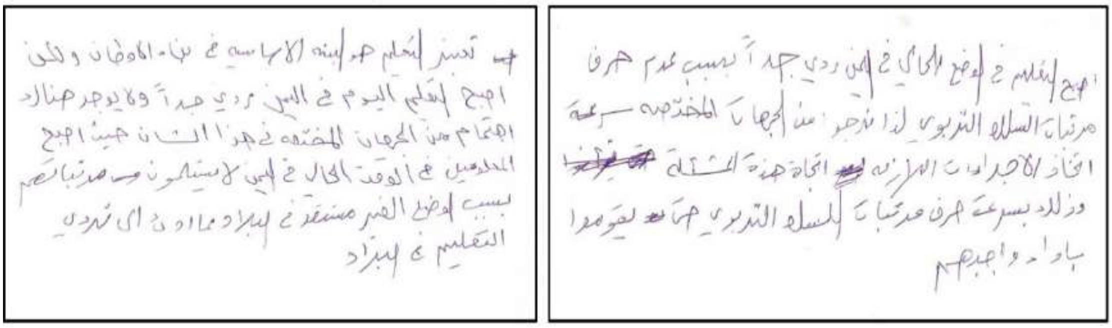
The database (handwriting of the Arabic text) was collected in two ways: the first way, 450 people of different age groups were selected, 250 of whom were university students and the rest were of different qualifications. Those people were requested to write a random Arabic text in the form of paragraphs, and write on A4 sheets, each paragraph consists of 3-5 lines. many of them were asked to rewrite the same text more than once in different time periods, we noticed that the writer's psychological state affects the aesthetics of the text and the quality of the font as shown in Figure 2.3. A comparison between writing for the same writer in two different time periods (quickly writing differs on careful writing), all the documents we have obtained are without diacritical marks. Furthermore, some of the documents entered are handwritten books. The collected documents were scanned by a scanner HP Deskjet F2120, 600 dpi resolution. the input document images were stored in JPG format.



يتضمن درستنا على الكسدة والاختزال التي تنشأ حينما اثارة للالكترونات ضوئياً وذلك
بتسليط الضوء على المادة الثانوية التي بدورها تعمل الكترونات لتأكسد الماء وبالتالي يقوم الماء
إنتاج جذور حرة وهي الكسجين حر وصوديوم هيدروكسيد التي يكون لها دور فعال
إما في إنتاج الأمونيا تأكسد غاز النيتروجين او اختزال ثاني أكسيد الكربون الذي بدوره
ينتج اول أكسيد الكربون ميثانو او ميثانول.

يتضمن درستنا على الأكسدة والاختزال التي تحدث عنها إثارة للالكترونات ضوئيا
وذلك بتسليط الضوء على المادة الثانوية التي بدورها تعمل الكترونات لتأكسد الماء
وبالتالي يقوم الماء إنتاج جذور حرة وهي أكسجين حر وصوديوم هيدروكسيد التي يكون
لها دور فعال إما في إنتاج الأمونيا تأكسد غاز النيتروجين أو اختزال ثاني أكسيد الكربون
الذي بدورة ينتج أول أكسيد الكربون او ميثانو او ميثانول.

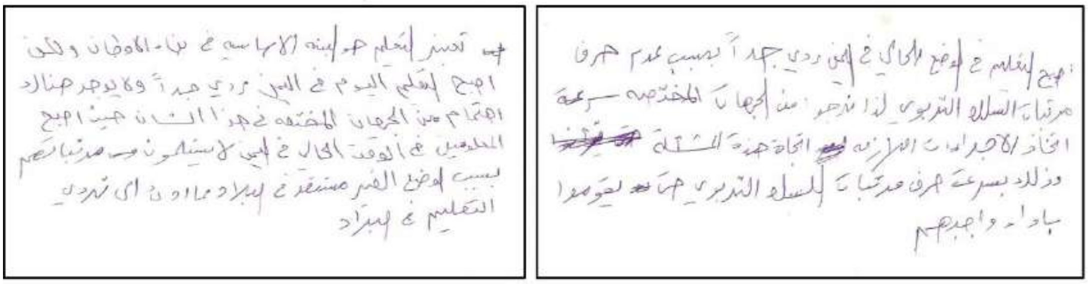
a) Sample 1



تعتبر التعليم هو البنية الاساسية في بناء الاوطان ولكن
اصبح التعليم اليوم في اليمن رديء جداً ولا يوجد هناك
اهتمام من الجهات المختصة في هذا الشأن حيث اصبح
المدرسين في الوقت الحالي في اليمن لا يستلمون مرتباتهم
بسبب الوضع الغير مستقر في البلاد مما ادى الى تردي
التعليم في البلاد

اصبح التعليم في الوضع الحالي في اليمن رديء جداً بسبب عدم صرف
مرتبات السلك التربوي لذا نرجو من الجهات المختصة سرعة
اتخاذ الاجراءات اللازمة اتجاه هذه المشكلة
وذلك بسرعة صرف مرتبات السلك التربوي حتى يقوموا
باداء واجبهم

b) Sample 2



c) Sample 3

Figure 2.3: Sample (a) was written twice by a writer himself at different times. As well as the samples (b) and (c).

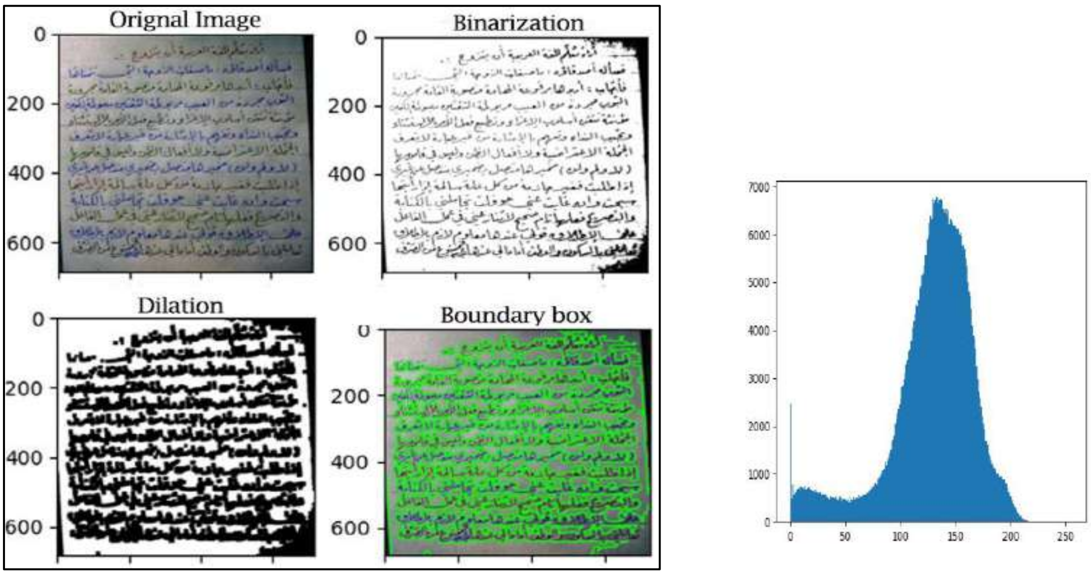
Then, since the images were scanned are good and clear, noise and distortion were not observed unless they were from the document itself, the brightness and contrast were constant, no global skewness, so after applying the pre-processing techniques, we got better outputs without problems and errors and the process was smooth for enhancements. Color images have been converted to binary.

The second way to collect images of Arabic handwriting is through members registered to social media applications, as Figure 2.4 shows a sample of these collected images. Some of the images we got are of poor quality, bright and contrast. We faced difficulty in solving the challenges of brightness and weakness when applying pre-processing techniques (the reason for weakness and poor quality is due to the poor resolution of the captured device, as well as the effect of the communication application on the quality of the transmitted images). Techniques in the pre-processing stage failed in 450 images sent, and their quality was improved, lighting and contrast adjusted by Photoshop. There are 140 images contain signatures. Due to the objective of this research work being character recognition, signatures have been removed (as will be explained later).

2.2.1 Pre-processing

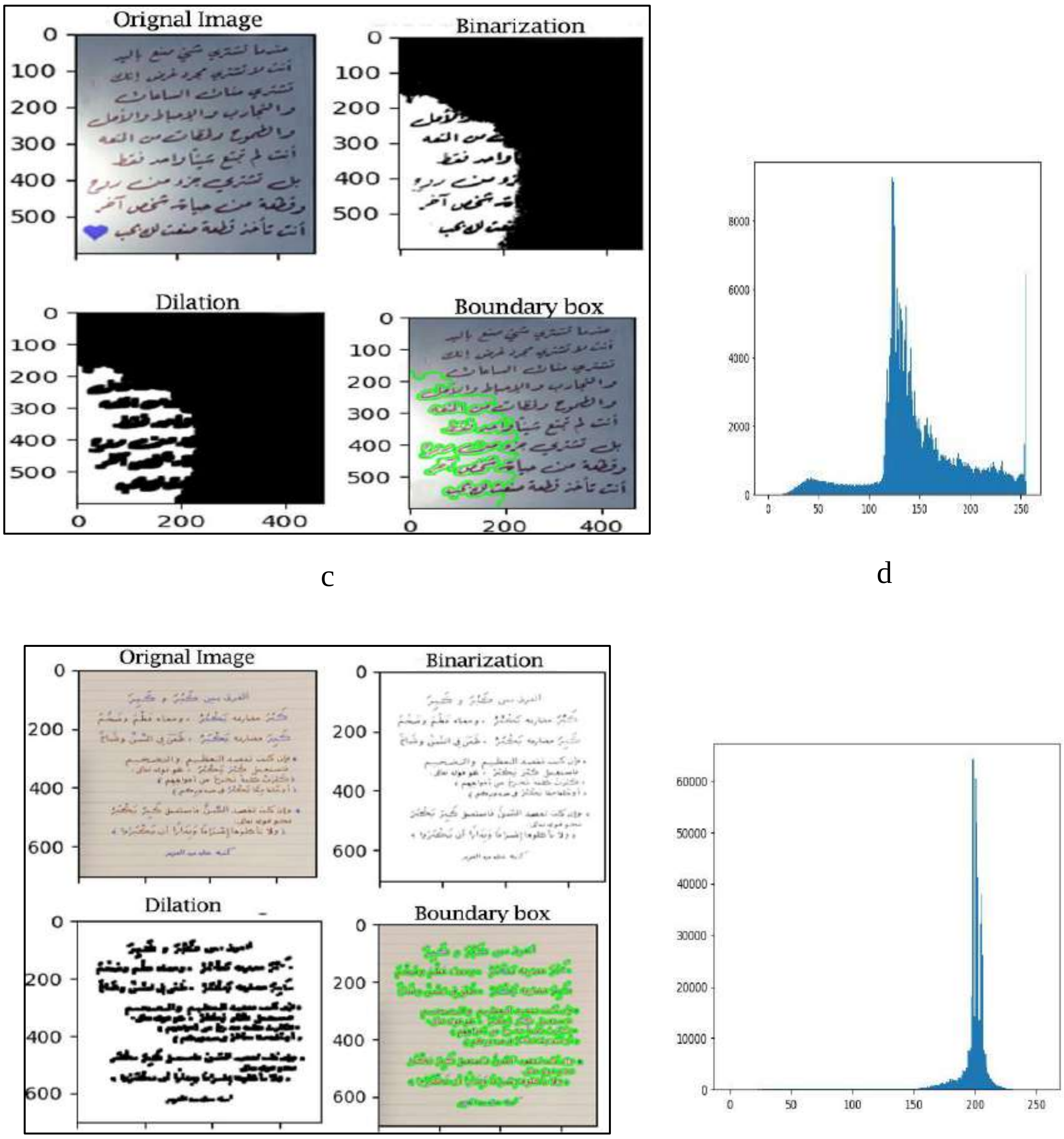
As mentioned earlier, the quality of the scanned images is clean and no noise or distortion is observed, so applied the pre-processing techniques were smooth and the methods are used less than pre-processing in the images collected through social media apps, techniques used such as salt and pepper, Gaussian and median filter to remove noise and small dots in the original document, and morphological operations to fill in the holes or gaps in the word. Otsu method was used converting to Binarization.

Pre-processing techniques were applied to the images collected through social media, these techniques succeeded very hard in some images and failed in the others as shown in Figure 2.4, so these bad images were enhanced by Photoshop CS6 and prepared to apply the recognition system. The problem of weak and unstable lighting in the whole document is a big obstacle which made the Preprocessing stage complicated, as there are different lighting points in the same (one) document, which leads to a lack of performance of the segmentation process.



a

b



c d

e f

Figure 2.4: Different lighting and Histogram of Intensity Distribution. (a), (c) and (e) are the input document images, while (b), (d) and (f) are the original images' Intensity Distribution Histogram for Showing the intuition about the Contrast Brightness Intensity Distribution.

One of the most challenges this study faced is the image quality where several document images which were collected, have poor quality and resolution as shown in Figure 2.4. The image (a) is darker than others. Its histogram in (b) placed at 0 - 200 to represent (the) pixel value in x-axis takes the image (a) pixel value and the value placed at 0 to 200 that indicated the values grouped at black values, this represents the image (a) consists of more black pixels compare to other gray level value. the image (c) is a low contrast image, its histogram (d) is placed at almost the center towards white. The image (e) is a bright image where its histogram (f) values are placed at 170 towards white 255, which indicated the image(e) has more white pixels. although the Preprocessing approaches were applied to these images the results were not good.

2.2.2 Signatures Elimination

Many images of documents contain signatures, so before shifting to the next stages of Arabic OCR stages, it should eliminate these signatures to help to word/ character segmentation. the method is used to remove these signatures depending on Connected Components and find out the thresholding average. By using image processing, the regions of connected pixels were recognized by this algorithm. It commonly gives the same result. In other words, the given input image is scanned by these connected components along with this attached signature.

The next step is gathering the pixels into new components connectivity i.e. the elements of the image connected to the same intensity values of each pixel and showing the link with other values. So, the components will be recognized, and every pixel will be highlighted with a specific color (color labeling). Each pixel may highlight with a grayscale according to the component which it has located. Nowadays, classification of each connected component along with the assorted dissociates are essential to many analysis applications of image's machine-driven. In this process, the whole image, from left to right and top to bottom, is scanned to recognize the region of each pixel which is connected to the image's component.

In other words, it can be said the adjacent pixels of each component share constant value V. it can be applied to binary or grayscale images, and it measures connectivity differently. here, before applying the mask technique move over, the input image should be a binary, 8-connectivity where is a mask created and each pixel and its surroundings are checked using this 8-connectivity. The operator moves over the image to scan rows individually until it arrives at ‘p’ point, it examines the remaining eight neighbors of the labeled pixel (at any stage, ‘p’ is the labeled pixel) for which V = [1]. Then it examines the four neighbored labeled pixels from right to left and from the upper diagonal direction, which were already encountered in the scan. According to previous details of the scan, the term ‘p’ is classified when the process finds an adjacent value equal to ‘1’. Then the label is assigned to ‘p’. At the same time, if other neighbors have the same value, all of the labels will be assigned. The outcome values from the equivalencies; if all pixels are 0, a new label will be given to ‘p.’ This process is followed by the initial scan of the label pair area units and sorted into equivalent categories (classes and distinct labels).

The next step is the second image scan, where every label is replaced by the equivalent category, even though the labels may not be identical. A Scikit image library provides an exciting feature for identifying and labeling connected components. We used this library to check the scanned input image documents and find these connected components with their labels in addition to grey and color labels. It turns out that the largest connected component is the signature compared with word components. Therefore, if we can extract the largest component of the whole document, we can recognize the signature.

However, using large connected components can extract unwanted words, lines, or other shapes. Therefore, a threshold value is used to solve this problem. We use the threshold value to detect outliers, i.e., any lines, structures, and texts that do not belong to signatures are calculated after a series of experiments which have been performed. In terms of a mathematical formula which is obtained based on experimental results. It gives quite effective recognition of signature's regions in dealing with most A4 size scanned documents. Because all collected document images were written on A4 sheets. The threshold value for removing attached pixels is relatively smaller than the variable that is formatted to A4 scanned documents.

$$\text{Cons\_A4} = [100 + (250*(\text{average}/84))] \qquad (2.1)$$

Table 2.2: Features of Signature' regions

Sample Images	Biggest Components	Average	Small Components	Big Components
A	924	162.508772	583.657059	10505.82707
B	7682	793.734940	2462.30634	44321.51463
C	1158	128.257353	481.718312	8670.929621
D	1029	38.2051282	213.705739	3846.703298
E	1156	77.5617021	330.838392	5955.091185
F	5888	123.899471	468.748425	8437.471655
G	566	85.7945946	355.341055	6396.138996
H	1294	106.838709	417.972350	7523.502304
I	2286	157.333333	755.555555	13600.00
J	624	118.603238	594.180161	1069.242914

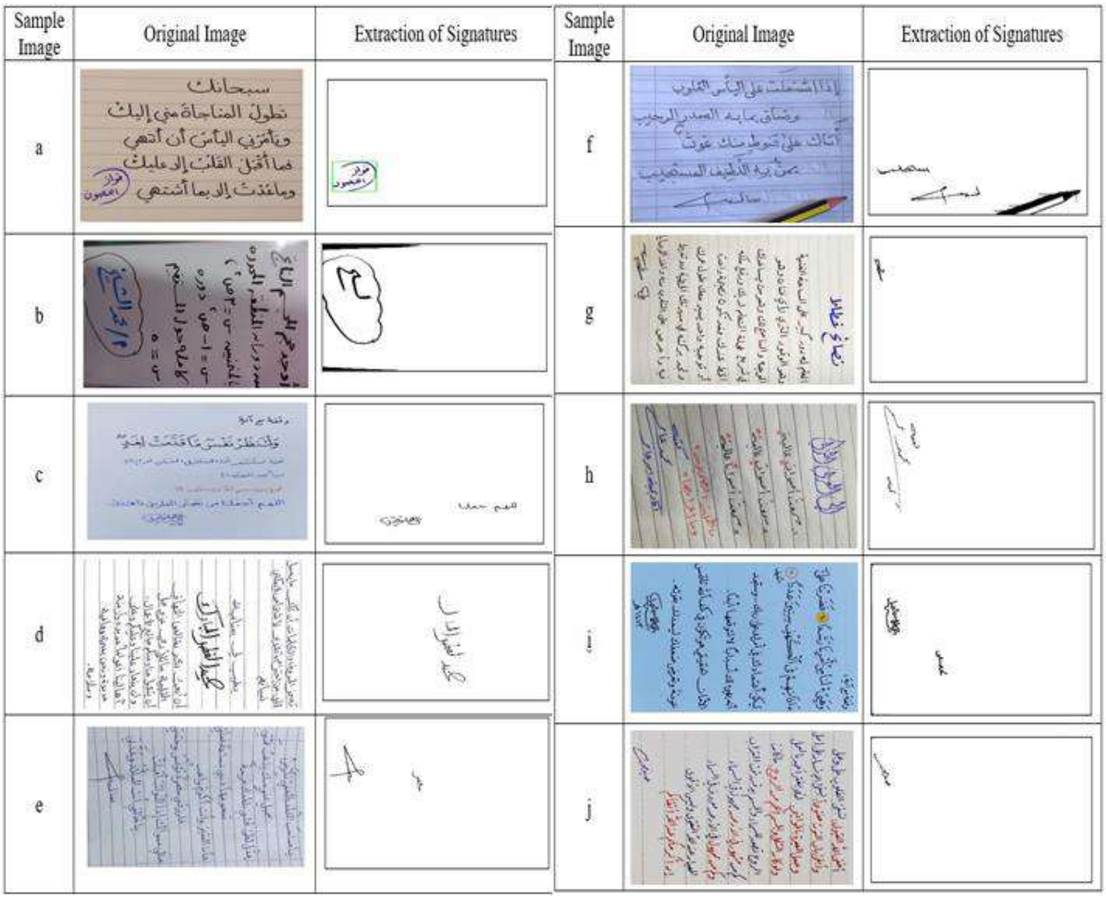


Figure 2.5: Extraction of Signatures from Samples Images.

140 images contain signatures as shown in Figure 2.5 for samples of these images. Signatures have been eliminated successfully in 122 images, the method failed in 7 images, segmentation excessively in 11 images, and the reason for excessive segmentation is that when the area of any word region where its components are connected is equal to the area of the signature region, the method considers it a signature and thus removes the particular region as shown in Figure 2.5. Table 2.2 shows the features of the signature regions.

2.2.3 Line Segmentation

The collected documents by the two previously mentioned ways (scanned, or collected from communication applications) were divided into independent lines, then, stored in as PNG images with an area which depends on the line's width and height because some lines contain a word, other some are two words and others are more, which increases the length and height of the saved line image as shown in figure 2.6.

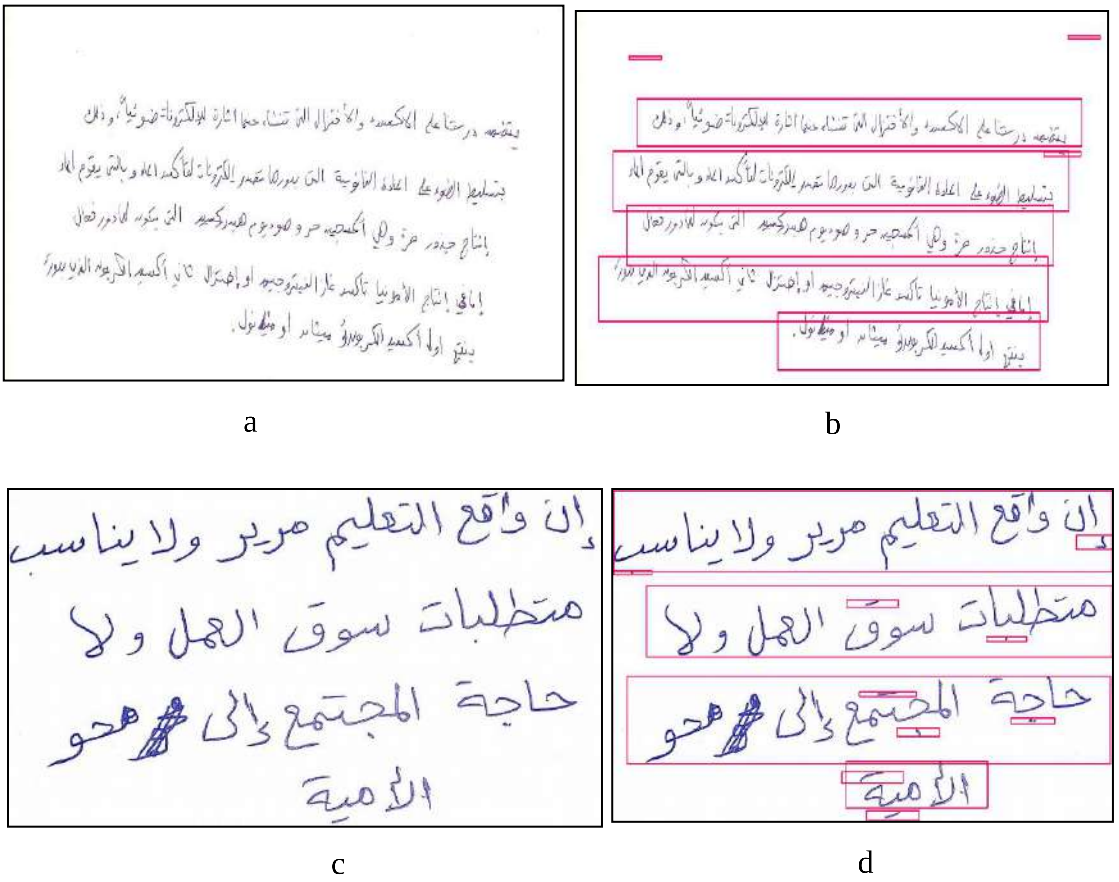
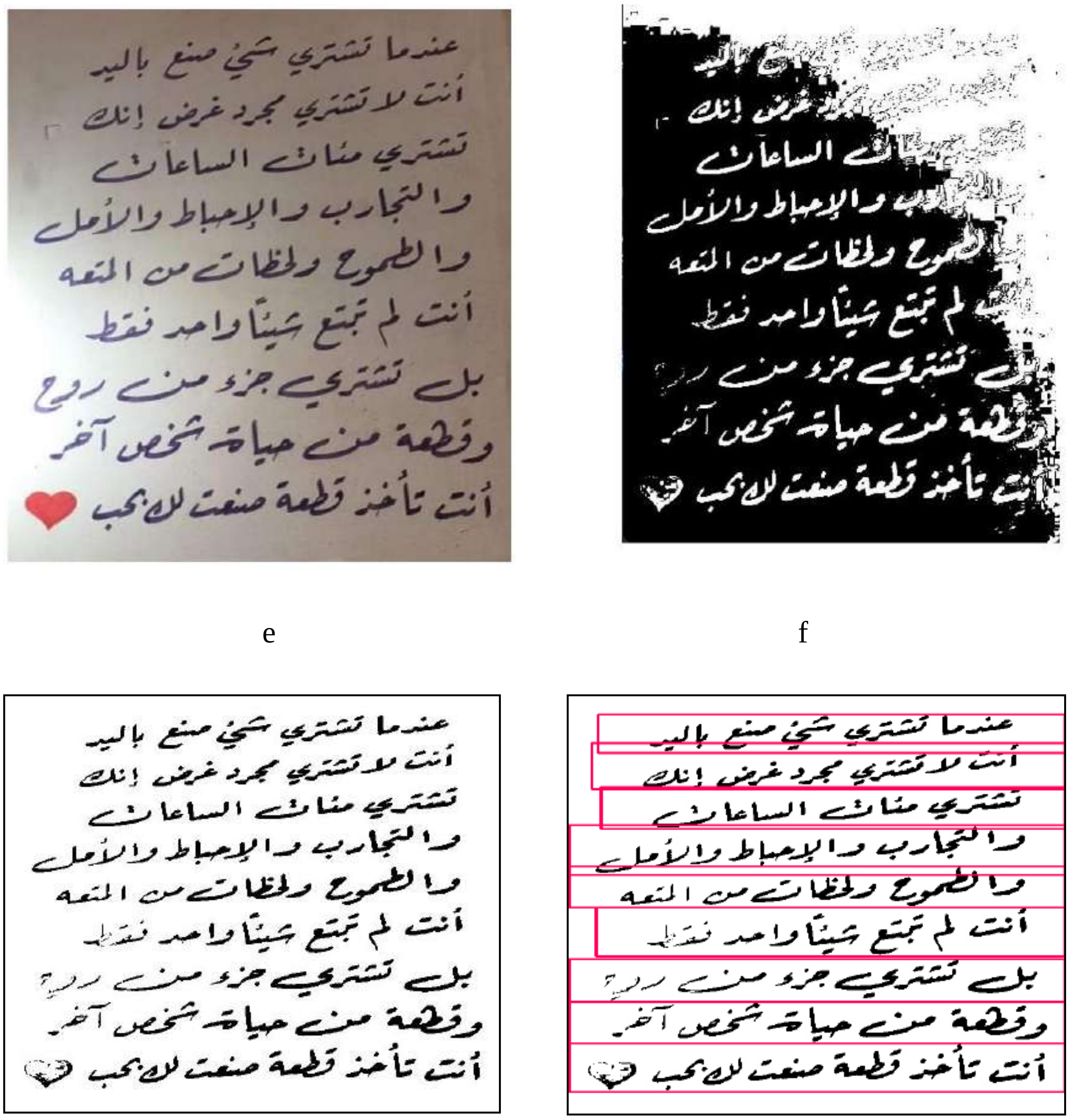


Figure 2.6: Two Samples of scanned images, (a) is the Original image which segmented into lines as shown in (b). (c) also is the original image and (d) shows the its line segmentation.



عندما تشتري شيء صنع باليد
أنت لا تشتري مجرد غرض إنك
تشتري مئات الساعات
والتجارب والإحباط والأمل
والطموح ولحظات من المتعه
أنت لم تبتع شيئاً واحد فقط
بل تشتري جزء من روح
وقطعة من حياة شخص آخر
أنت تأخذ قطعة صنعت لك بحب

e f

g h

Figure 2.7: (e) is one Sample of Original image collected from the communication app, (b) shows a failure of line segmentation. (g) is enhanced by Photoshop and (f) is line segmentation of the (g) image.

From Figure 2.7, we noticed that the image segmentation process into lines failed because of poor image quality and brightness such as in sample (e), although, (despite) the application of pre-processing techniques on it, the line segmentation process was bad, after improving the quality manually, segmentation process success better than automatic enhancement, such as in image (g), this procedure makes it impossible for applying the OCR system automatically on Arabic Handwritten text recognition. The number of collected documents from social media apps, that contain illumination and poor quality by 7% rate, most of the documents were good.

There are other techniques used in the pre-processing stage, they will be explained in detail in chapter 3, which will be discussed for database images enhancement. one of the main reasons for creating a database for this work is to test the success of the pre-processing process on different and varied images entered in many ways. Images from previously available Arabic databases are often binarized. Table 2.3 shows the statistics for the proposed database of Arabic documents.

Table 2.3: Statistics of Arabic Handwritten Database

Variables	**Amount**
Number of writers	450
Number of senders (members)	270
Number of documents	5500
Images contain signatures	140
Old handwritten books	1300
Transferred documents through apps	1500
Number of Lines of DB	
Number of words	302348
Number of Characters	1257191
Training sets	10 %
Testing sets	10 %
Total of document images	6300

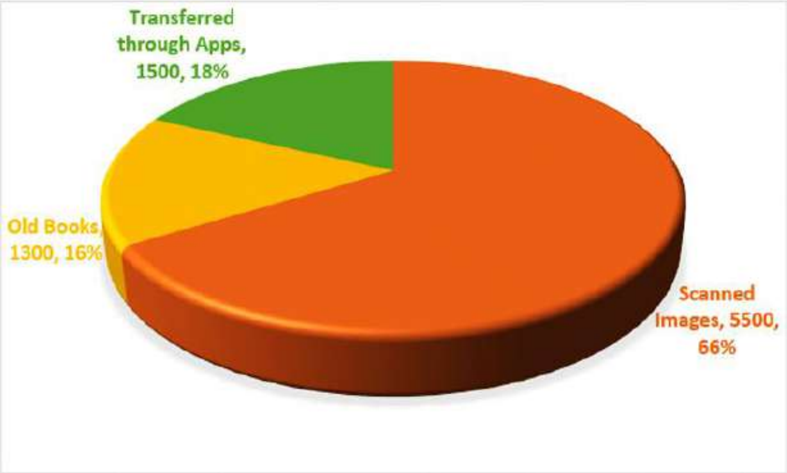


Figure 2.8: Percentage of the proposed Arabic Handwritten Database Sources.

Chapter 3

Enhancement of Arabic Document Images

3.1 Preamble

Many studies have been paying due attention to the recognition of Arabic words during the last few decades. Offline handwritten Arabic characters recognition has received more attention in these studies, due to Arabic document digitalization requirements (Ahmed, 2016). The domain of recognition (for Arabic handwritten characters in this study) has gained due attention from researchers and authors due to the recent development in operating systems which resulted in various fields of recognition such as image processing, pattern recognition, machine learning etc. which all have to go through a pre-process step that is effective in respect of noise removal, reducing blurring, increasing contrast, skew/slant correction [Biswas, 2011, Vishwanath V.N., 2018], and enhancing images quality for preparing such images to

Some parts of the material in this chapter have appeared in the following research papers:

1. Ravikumar, M., and Omar Ali Boraik. "Estimation and Correction of Multiple Skews Arabic Handwritten Document Images." International Conference on Innovative Computing and Communications. pp. 553-564. Springer, Singapore, 2021.
2. Ravikumar, M., and Omar Ali Boraik. "Low Pass Filter-Based Enhancement of Arabic Handwritten Document Images." International Conference on Information and Communication Technology for Intelligent Systems. pp. 271-277. Springer, Singapore, 2020.

the next stages in the process. Some of the past literature relied on ready database to apply their algorithms for character recognition.

Going through the pre-processing steps in recognizing the handwritten characters is mandatory to improve the readability and the automatic recognition of handwritten document images. These steps include converting-colored images into greyscale mandatory to improve the readability and the automatic recognition of handwritten document images. These steps include converting-colored images into greyscale images and then into binary images, along with noise removal, filtering, text normalization, slant normalization and skew correction. The feature extraction process is more effective and reliable while following such steps. When it comes to recognizing Arabic handwriting, many studies focused on improving its quality, but due to the unique nature of the script, the effectiveness of conventional methods was not proven fulfilled. Therefore, in order to enhance the handwriting recognition system, Pre-processing is crucial in reducing the viability of handwriting through correcting the factors for enhancing the accuracy of segmentation and recognition methods (Boukerma and Farah, 2012). The importance of pre-processing is based on assuming that extracting and distinguishing the objects from the background depends on the quality of binary images.

Besides, removing the noise and black spaces in the background is the next step in the pre-processing, after that comes the step of correcting the rotation of the text in the image to create stability for the image which provides a clear result for the recognition process especially for the feature extraction step (El Abed and Margner, 2007). Furthermore, the image normalization reduces the image size to prepare the last edition of the image and send it to the next step of the recognition process. Relying on these databases will save the effort of going through some techniques on the pre-process step such as converting colored images to gray scale, Binarization, Noise removal and skew correction. This study relayed on a sample of fresh images captured for the study. This study contributes to the literature regarding the pre-process stage of textual images of Arabic handwriting that were taken by capturing devices like sensors in order to enhance the blurry images of low quality and getting them ready for the next stages. Enhancing the captured images is affected by few factors such as quality and stability of the capturing devices, resolution, degraded and also historical documents, illumination effect, variability in the font and writing style, etc.

The performance of OCR depends on the quality of the textual image acquisition and the lack of blurring and noise. Document images of poor quality and high intra-class variation are of greater difficulty for the recognition process (El Abed and Margner, 2007). Hanen B. and others [Boukerma and Farah, 2012, El Abed and Margner, 2007] applied noise removal, smoothing, slop/skew correction, baseline detection on IFN/ENIT database, and reached quite satisfying enhancement which was 83% through eliminating diacritics. (Likforman-Sulem et al., 2009) compared two approaches Non-local means with total variation minimization to enhance the old printed documents and concluded that based on image features one or both approaches can be used for enhancement.

(El-Hajj et al., 2005) Focused on the shape of the word (upper and lower parts from baseline) in IFN/ENIT database and minimized the errors to 11%. (Humied, 2016) studied 540 input images characterized with overlapping words through suggesting a modified bounding box segmentation algorithm by using salt and pepper and median filter by 3×3 mask to remove noises. Skew correction and binarization proved the efficiency of the suggested approach. Ahmad Sahlol and Cheng (Sahlol and Suen, 2014), focused the statistical, morphological and topological of fonts from CENPRMI database through using Otsu algorithm to Binarization, median Filter by 3×3 mask as well as morphological for noise removal and isolating pixels and achieved accuracy range 61% - 83%. Manal A. and others (Abdullah et al., 2012), reached 81% accuracy through applying two stages over IFN/ENIT database started with segmentation and then enhancement by removing the white area around the word, Binarization and storage of the enhanced word images.

(Farooq et al., 2005) focused on removing diacritics marks from the words on IFN/ENIT database to draw paper effective baseline. Omar Balola and Adnan Shaout’s study (Ali and Shaout, 2014), presented series of techniques to perform and enhance the textual images such as Binarization using thresholding technique, thinning, Normalization, Noise removal, Skew detection and correction, Skeletonization and smoothing filterization. Atallah M.Al-Shatnawi (Al-Shatnawi, 2014), Makki Maliki and others (Maliki et al., 2012) applied noise analysis and removal, skew correction on the Naskh font type from specific database and presented a skew detection method and explained how to correct it using different ways through testing their studies on IFN/ENIT database and the accuracy rate increased from 83% to 92% .

From the literature review, we noticed that most researchers had used common techniques in the pre-processing stage such as Binarization, skew/slant correction, smoothing, noise removal, normalization and thinning and their methodologies were applied to recognize the textual images which most of them are from existing databases, i.e the existing algorithms are tailor made. Hence it is essential to develop an efficient algorithm for image enhancement.

3.2 Proposed Method for enhancement

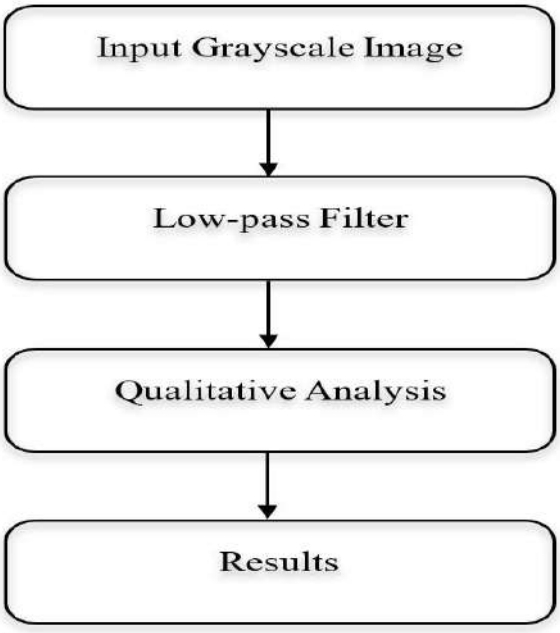


Figure 3.1: Shows the steps are followed for enhancing the poor image.

3.2.1 Noise Reduction and Smoothing

This step aims to reduce the noise, counteract the inappropriate bright and contrast in the captured image, and raise the quality of the targeted poor image. This step is known as the enhancement process. In the third chapter of this research, we have discussed the challenges, problems, and causes of the poor quality of captured images and the importance of this stage and its impact on increasing the efficiency of the Arabic Handwritten OCR system has already explained. In this section, the proposed filters used in the improvement process will be discussed and compared between them.

a) **Filters**

Enhancement processes are considered the primary and problem-oriented tasks (weakness problems, Blur, illumination problems. etc.). Most previous research prefers to use filters to perform these essential tasks. from the literature review, the filters can be categorized into Spatial domain filtering and Frequency Domain filtering.

b) Spatial Domain Filtering:

The methods in this group deal with the pixel values of the image in order to achieve the desired improvement. The idiom spatial refers to collecting pixels that make up the input image (Woods, 2004). This type of filter uses a specific mask to convolution the 2D matrix of the image.

c) Frequency Domain Filtering:

Even though spatial domain filtering techniques are easy to understand, implement but frequency domain method has few distinguishing benefits for pre-processing. The convolution and correlation are computational can be performed in a better and more efficient way using frequency domain pre-processing methods. This method is computationally beneficial and reduces the requirement of storage and bandwidth for pre-processing and other character recognition steps.

The 1-D (1 dimensional) Fourier transform for a single variable $F(x)$, x = 0,1,2…N-1, discrete function is given by

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{-j2\pi ux/N} \qquad u = 0, 1, \ldots\ldots N-1 \qquad (3.1)$$

The original formula can be gained using Discrete Fourier Transform DFT

$$F(x) = \frac{1}{N}\sum_{x=0}^{N-1} f(u)e^{j2\pi ux/N} \qquad u = 0, 1, \ldots..N-1 \qquad (3.2)$$

The corresponding discrete Fourier transform pair for the 2-D function is

$$F(u, v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x, y)e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \qquad u, v = 0, 1, \ldots..N-1 \qquad (3.3)$$

And

$$F(u, v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x, y)e^{j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \qquad x, y = 0, 1, \ldots..N-1 \qquad (3.4)$$

There are many steps mentioned in (Gonzalez, 2009) that should be followed to process the enhancement using frequency domain; these steps are:

- First multiply the digitized image by $(-1)^{x+y}$ to center the transform.
- The Discrete Fourier Transform DFT $F(u, v)$ converts the resulting image into the frequency domain.
- Perform the process on $F(u, v)$using the filter function $H(u, v)$, in frequency domain the process will be multiplication because the convolution in the frequency domain reduces to multiplication in spatial domain and vice versa. i.e: $f(x, y)h(x, y) \leftrightarrow F(u, v) * H(u, v)$
- The entire performance process is executed on document images in the frequency domain.
- Then inverse DFT of the outcomes after performing pre-processing processes i.e apply Inverse Fourier Transform IFTT on image which transforms the image into spatial domain.
- Finally, multiply the result by $(-1)^{x+y}$.

Median Filter:

A filter's response is dependent on the ordering of intensity values in the neighborhood of the pixel that's being analysed. In this type of filtering, the first set of intensity values of all the neighborhoods of (x,y) is taken. This set of intensity values is then arranged in a specified order. Select the median value of these orderings to determine the center pixel.

Gaussian Filter:

we used Gaussian Filter to enhance the document image in the experimentation. Gaussian filter is used to noise removeable and blur images, the equation (3.5) defined this filter. the quantitative results are shown in table 3.1, and the outcomes are shown in figure 3.2.

$$H(u,v) = e^{-\frac{D^2(u,v)}{D_0^2}} \qquad (3.5)$$

Where $D(u,v)$ is given $D(u,v) = \left[\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2\right]^{1/2}$ and D_0 is cut off high frequency.

Lowpass Filter:

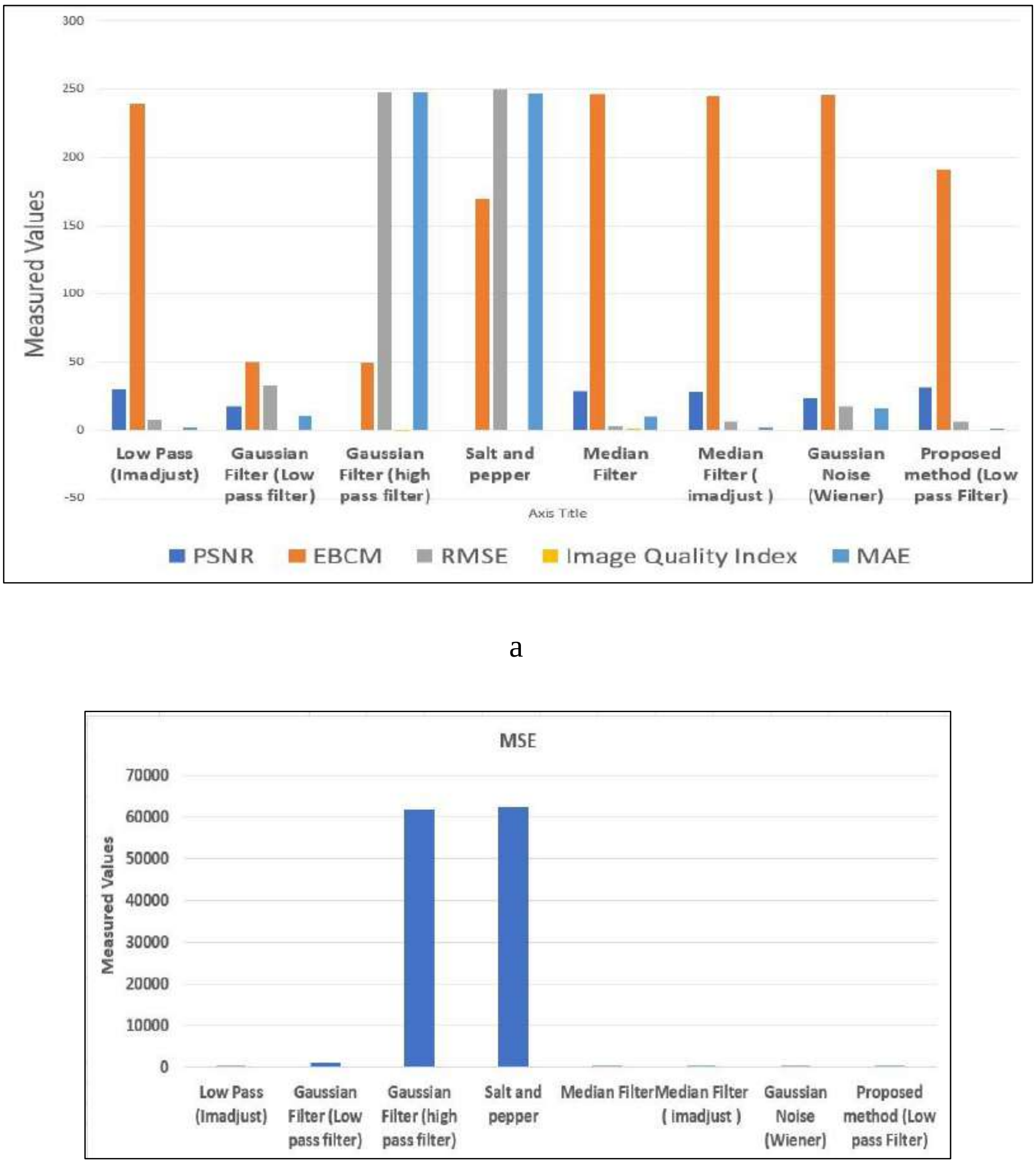
Image quality and the affected properties, such as contrast, brightness, texture, and brightness, are related to high and low frequencies. Low-frequency information helps enhance brightness and texture in poor-quality images. The lowpass filter is better than other filters in improving lighting, brightness, texture, and smoothing. the quality of the images is computed by using different quantitative measures which are given in table 3.1. After the experimentation, low pass filter performs better than other filters.

3.2.2 Experimentation on Filters

Table 3.1: Quantitative measures for different enhancement methods.

Filter(s)	**PSNR**	**EBCM**	**RMSE**	**Image Quality Index**	**MAE**	**MSE**
Low Pass (Imadjust)	30.00	239.27	8.09	0.08	1.68	65.00
Gaussian Filter (Low pass filter)	17.75	49.74	33.17	0.01	10.92	1100.02
Gaussian Filter (high pass filter)	0.26	49.22	248.39	- 0.023	248.39	61760.33
Salt and pepper	0.21	169.96	249.89	0.00	247.43	62444.66
Median Filter	28.68	246.36	2.98	0.94	10.39	88.87
Median Filter (imadjust)	27.92	245.45	6.28	0.78	1.86	105.77
Gaussian Noise (Wiener)	23.12	246.05	17.88	0.09	16.20	319.83
Proposed method (Low pass Filter)	**31.71**	**191.06**	**6.65**	**0.24**	**1.29**	**44.18**

Different enhancement techniques like Low Pass (Imadjust), Gaussian Filter (Low pass filter), Gaussian Filter (high pass filter), Salt and pepper, Median Filter, Median Filter (imadjust) and Gaussian Noise (Wiener) methods are compared with the proposed technique i.e., low pass filter (LPF) to prove the superiority of the proposed method. The results obtained from all the techniques are shown in the figure 3.4. And its corresponding values are tabulated in table 3.1. Finally, the graphs are plotted for quantitative measures given in figure 3.2 From the experimentation, it is observed that the proposed method gives better results.

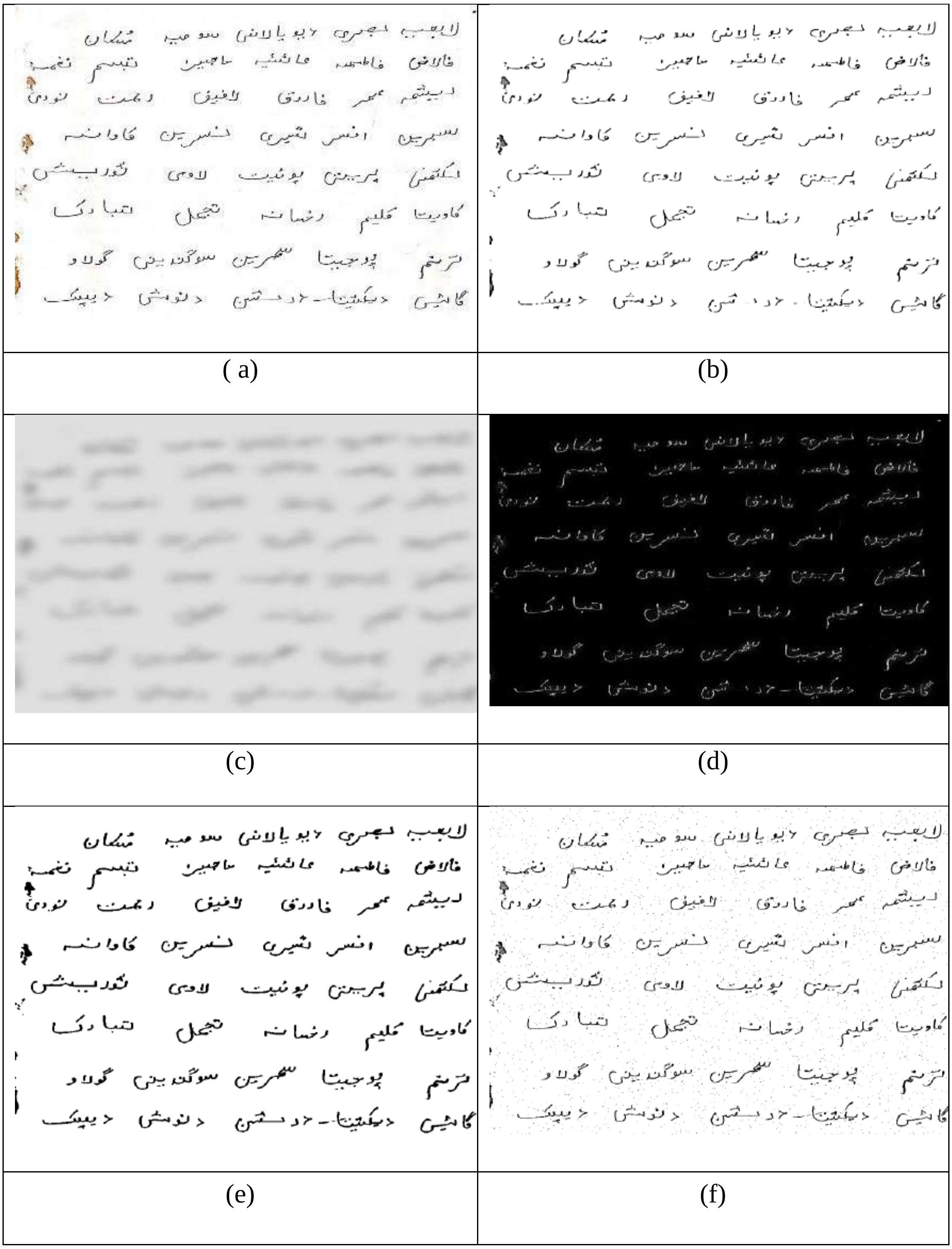


a

b

Figure 3.2: Qualitative analysis for different enhancement methods. (a) Represents the results for PSNR, EBCM, RMSE, IQI and MAE (b) represents the results for MSE.

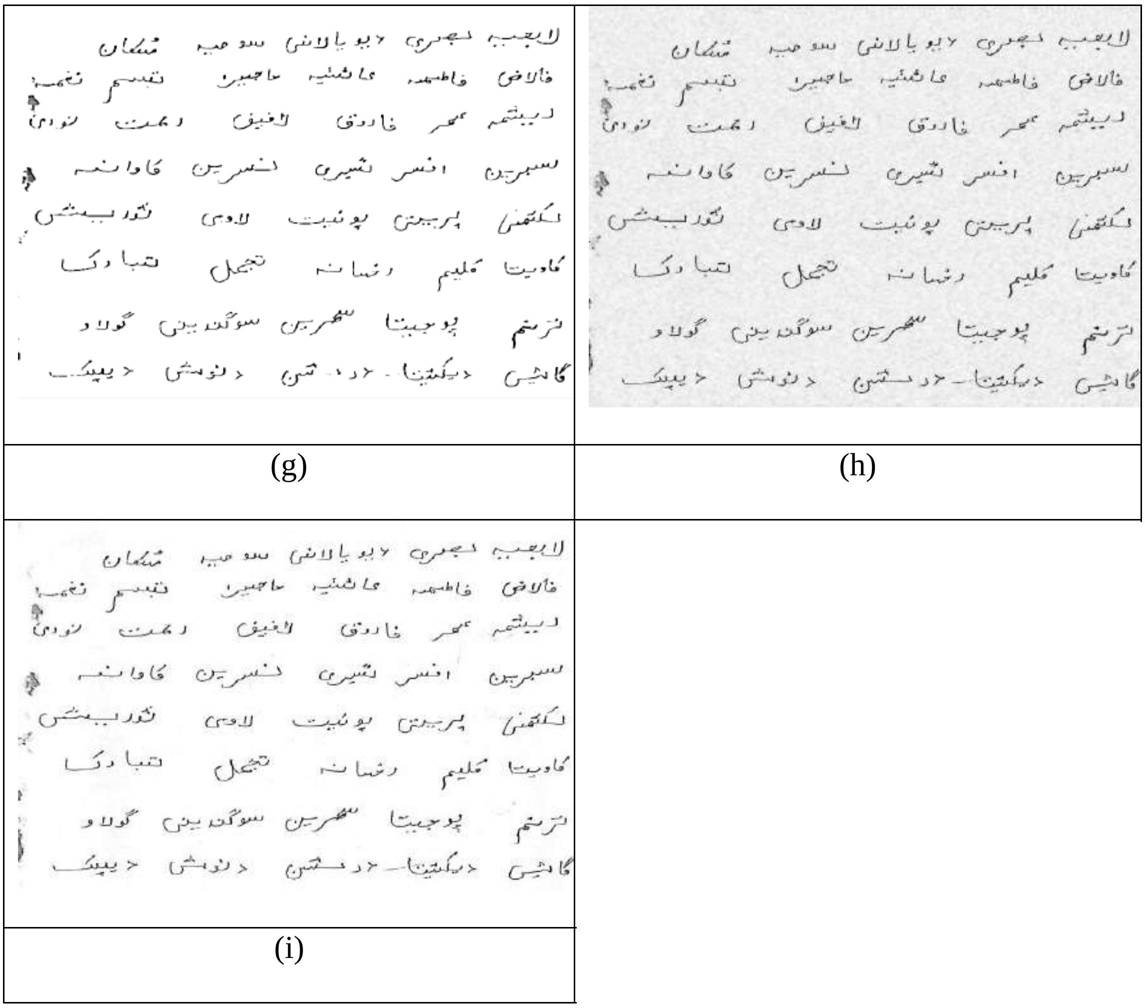
We have proposed a robust method for the enhancement of handwritten Arabic document images. To improve the picture quality, we have applied different enhancement techniques. After the experimentation, low pass filter in frequency domain was found to perform better. The experimentation was carried out on our own dataset. Because of the frequency domain approach, in general computational time is more. Figures 3.3 shows the resultant images.



(a) (b)

(c) (d)

(e) (f)



(g)

(h)

(i)

Figure 3.3: Showing the results of different methods visually. (a) Original Image (b) Median Filter (c) Gaussian Low Pass Filter (d) Gaussian High Pass Filter (e) Low Pass (Imadjust) (f) Salt and pepper (g) Median Filter (Imadjust) (h) Gaussian Noise (Wiener) (i) Proposed Method.

3.3 Skew Detection and Correction

Within the process of digitizing, documents may not come out as expected to be, i.e. documents may have noise, low contrast, the documents may have not been captured properly, or the documents could be of low quality as in old or badly written documents, for instance, the writing could have multiple skew angles due to handwriting mistakes or using unlined sheets (Ramanan, 2019), any mistakes or errors accompanying the digitizing process may cause inconvenience to subsequence image processing, and may even lead to wrong results.

In this section a method for detecting and correcting multiple skew angles in Arabic handwriting document images. This method is relied on process steps which start with pre-processing step and ends with the correction of lines skewness. The outcome of using the proposed method was significant in respect of running time, accuracy and memory consumption.

3.3.1 Various Skew Detection and Correction Techniques:

Over recent years, the techniques and methods for detecting and correcting skews on improperly acquired document images into OCR systems have been developed and improved (Ravikumar et al., 2019). Skew detection and correction are considered as mandatory steps in the pre-processing stage because they directly impact the reliability, accuracy and efficiency of the next stages in OCR systems [Boukharouba, 2017, Makkar and Singh, 2012]. Although these methods and techniques had been performed well on printed documents, they are still under improvement and development to overcome the challenges and difficulties of handwritten and historical documents (Guru et al., 2013). The most popular method is projection profile [Al-Khatatneh et al., 2015, Al-Shatnawi and Omar, 2009, Guru et al., 2013, Ramanan, 2019], and Hough transform [Ahmad, 2013; Jundale and Hegadi, 2015], Boukharouba and Abdelhak (Boukharouba, 2017) had concentrated on accuracy but it consumed a large memory [Al-Khatatneh et al., 2015; Ramanan, 2019]. Nearest neighbor (Al-Khatatneh et al., 2015) gets more errors when it is applied on older Arabic documents (Al-Shatnawi and Omar, 2009). And there are more than 13 methods discussed by [Ramanan, 2019, Ravikumar et al., 2019]. Techniques and methods have recently expanded to involve multilingual and multiple skew angles in a single handwritten document (Guru et al., 2013). Dharam Veer and Gurpreet Singh [Rezaei et al., 2013, Sharma and Lehal, 2009] had proposed a method for multiple skew angles in a printed Indic (Gurmukhi) Script. (Guru et al., 2013) suggested a method to detect and correct a multiple skew angle on multilingual handwritten documents, moreover, another method was proposed by (Ravikumar et al., 2019) which is estimated skew angle based on region properties of a word from Trilingual Handwritten Documents. There are several studies focusing on skew detection and correction in Arabic documents that may or may not have multiple skew angles [Abdullah et al., 2012, Al-Shatnawi and Omar, 2009, Rezaei et al., 2013].

There is a difficulty and complexity in the recognition of Arabic language text compare to other languages like Chinese, Japanese and Latin because Arabic language is written cursively, Arabic characters are written connectivity (Makkar and Singh, 2012), therefore a scant amount of literature focused on skew detection and correction in respect of Arabic text. This triggered the need for this study in order to focus on multiple skew detection and correction of Arabic handwritten documents.

3.3.2 Proposed method:

We discuss the proposed methodology for skew detection and correction of Arabic documents. The block diagram of this work in Figure 3.6 shows the steps are followed, the first stage describes the pre-processing steps of the input image, second stage describes line segmentation using morphological operation (dilation), bounding box, segmenting the bounding. The third stage explains detecting a skew angle using computational analysis and finally, the last step describes the line skew correction.

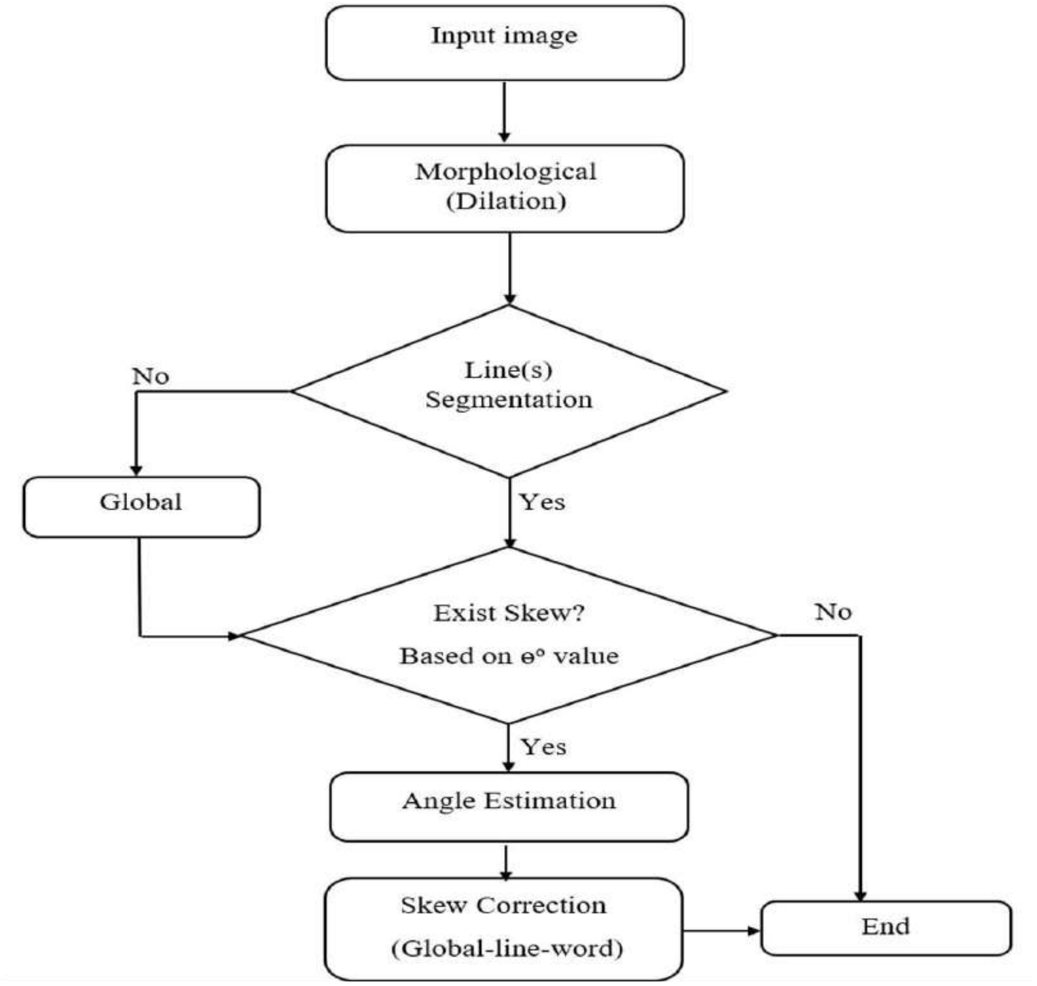
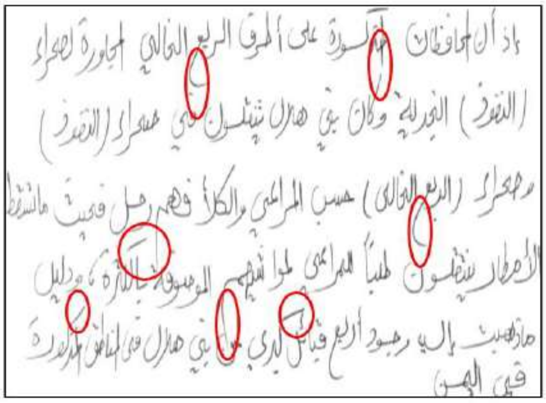


Figure 3.4: Block diagram of the proposed method.

Mostly, OCR system processes the input binary image so, if the input image is colored, it should be converted into grayscale image firstly, then Otsu's method is applied to binarization. After that, noises and dots are removed using several filters as mentioned previously. Some Arabic document images have diacritical marks or some letters have vertically long lines such as "أ، ح، خ، ج، ع، غ، کـ، ل، م ", are shown in figures 3.5 and 3.6

and these cause interference among lines and errors during the process. To solve these problems, if any area is < 40 pixels, it should be eliminated, dilation operation is used to expand the line structure to overcome the overlapping in the word and among the words in the same line as well as making its components connected, after that edges are detected in order to be in position to draw a Bounding Box (Al-Shatnawi, 2014).



إذ أن المحافظات المذكورة على الموقع الربع الخالي المجاورة لصحراء
(النفوذ) الكبرى وكان بني هلال يتنقلون بين صحراء (النفوذ)
وصحراء (الربع الخالي) حسب المراعي والكلأ فهم رحل فحيث ما اشتد
الأمطار يتنقلون طلبا للمراعي لمواشيهم الموسومة بالآثار ودليل
ما ذهبت إليه وجود أربع قبائل لدى بني هلال في المناطق المذكورة
في اليمن

Figure 3.5: Shows some letters have vertically long lines.

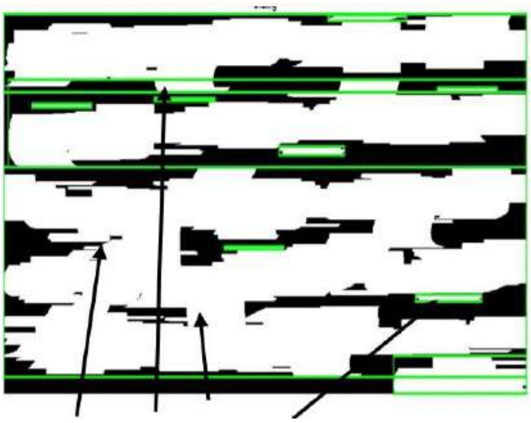


Figure 3.6: Shows interference Bounding Box among lines.

3.3.2.1 Line segmentation

Detecting edges or black pixels are checked on 0-1 or 1-0 conversion during the image tracing by using a matching window of 3×3 sizes, for determining the 8-neighbours of any given pixel. The 8 neighbours are used to find the edges in eight possible directions. Then a first minimum point (x_1, y_1) or (x_2, y_2), and the last highest point (x_3, y_3) or (x_4, y_4) of the line's connected components are determined for drawing Boundary Box. The detected edges coordinates are labelled and stored in a 2-D array to find out the skew angle and use it later in other stages (Al-Shatnawi, 2014).

3.3.3 Skew detection and correction

First, detecting the skew angle of line's coordinate as well as analysing multivariate data are carried through computing the mean vector and the line's matrix variance-covariance, the set of y_j observing and measuring x_i variable can be described by its mean vector and variance-covariance matrix, the Y_j variables from left to right are length, width and height of a certain object. The mean vector consists of the means of each variable and the variance-covariance matrix consists of the variances of the variables along the main diagonal and the covariances between each pair of variables in the other matrix positions, to compute the covariance of variables of x and y, we use this formula:

$$C = \frac{\sum_{i=1}^{n}(Xi-\bar{x})(Yi-\bar{y})}{n-1} \tag{3.6}$$

Where $\bar{x}$ and $\bar{y}$ denoting the mean of X and Y.

The results are:

$$\text{C= } \lambda \begin{bmatrix} x,x & y,y \\ x,y & y,x \end{bmatrix} \tag{3.7}$$

produces matrices of eigenvalues (λ) and eigenvectors (x) of matrix C, so that C =λ* x . Matrix D is the canonical form of C--a diagonal matrix with C's eigenvalues on the main diagonal. Matrix V is the modal matrix--its columns are the eigenvectors of C.

Getting the orientation of the ellipse from the eigenvectors and eigenvalues of C, the Eigenvalues and eigenvectors method of a symbolic matrix C returns matrices V and D. The columns of V present eigenvectors of C. The diagonal matrix D contains eigenvalues. If the resulting V has the same size as C, the matrix C has a full set of linearly independent eigenvectors that satisfy A*V = V*D.

Table 3.2: Shows eigenvalues and eigenvector of C matrix for each line and its angle (theta θ).

Samples	Number of lines	Eigenvalues and eigen vector	Angle for each line
Sample 1	3	C_1 = 1.0 e+03* 1.3723 8.1960 0.2819 1.3723 C_2 = 1.0 e+03* 0.1085 5.5811 0.0529 0.1085 C_3 = 1.0 e+03* 1.1991- 5.5960 -1.1991 0.3102	θ_1 = -170.4365 θ_2 = -178.8765 θ_3 = 167.7980
Sample 2	1	C_1 = 1.0e+03 * 2.4886 -2.2661 -2.2661 2.6383	θ_1 = 134.0541
Sample 3	4	C_1 = 1.0e+04 * 3.2216 -0.3924 -0.3924 0.0844 C_2 = 0.5432 -0.0988 -0.0988 0.9877 C_3 = 1.0e+04 * 3.4061 -0.2502 -0.2502 0.0997 C_4 = 1.0e+04 * 3.4126 -0.3843 -0.3843 0.0548 C_5 = 1.0e+04 * 1.1450 -0.2146 -0.2146 0.0430	θ_1 = 173.6061 θ_2 = 172.8575 θ_3 =170.8987 θ_4 = 171.3042
Sample 4	5	C_1 = 1.0e+04 * 3.1545 -0.1535 -0.1535 0.0190 C_2 = 1.0e+04 * 3.2805 -0.2147 -0.2147 0.0434 C_3 = 1.0e+04 * 3.1097 -0.3315 -0.3315 0.0565 C_4 = 1.0e+04 * 3.0927 -0.3162 -0.3162 0.0427 C_5 = 1.0e+03 * 1.4419 -0.1324 -0.1324 0.0483	θ_1 = 177.2032 θ_2 = 176.2214 θ_3 = 173.8740 θ_4 = 174.1434 θ_5 =174.6216
Sample 5	2	C_1 = 1.0e+04 * 3.1569 0.3996 0.3996 0.1366 C_2 = 1.0e+04 * 2.7954 0.2691 0.2691 0.1249	θ_1 = -172.5899 θ_2 = -174.3033

Sample 6	6	C_1 = 1.0e+05 * 3.4222 0.0859 0.0859 0.0075 C_2 = 1.0e+05 * 3.2960 0.0966 0.0966 0.0068 C_3 = 1.0e+05 * 3.0464 0.1055 0.1055 0.0071 C_4 = 1.0e+05 * 2.6201 0.1191 0.1191 0.0085 C_5 = 1.0e+04 * 4.8140 0.4509 0.4509 0.0775 C_6 = 1.0e+04 * 2.8751 0.1537 0.1537 0.0402	θ_1 = -178.5604 θ_2 = -178.3200 θ_3 = -178.0140 θ_4 = -177.3952 θ_5 =-174.6104 θ_6 =-176.9060

Now we can get the orientation of the ellipse from the eigen vectors and eigen values of C matrix which can be obtained by Velocity vector (quiver) where a quiver plot displays velocity vectors as arrows with components (u , v) at the points (x , y).

A velocity vector represents the rate of change of the position of an object. The magnitude of a velocity vector gives the speed of an object while the vector direction gives its direction.

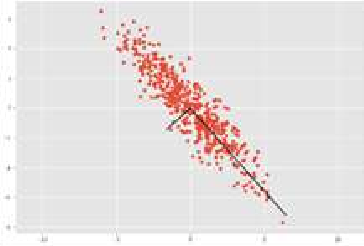
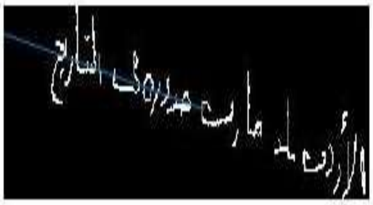
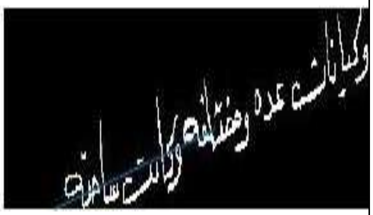
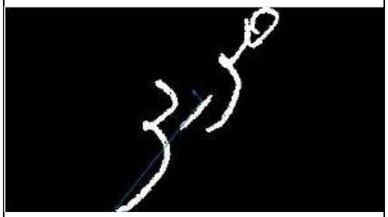
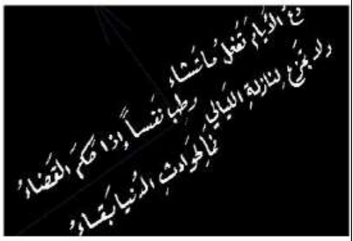
a	b	c
		
d	e	f
		

Figure 3.7: Images with different coordinate position (line, word, entire image).

From table 3.2, we can see that the eigen-vectors (columns of V) are approximately the pointing to the -X direction (first column) and the Y direction (second column). Examining the eigenvalues (diagonal of D) you can see that the second eigenvalue is much larger than the first - this is the major axis of your ellipse. Now we can recover the orientation of the ellipse by finding a major axis index (largest eigenvalue of diagonal D).

Second, according to major axis index of diagonal D we finally get the slope value. The skew angle can be calculated using the following formula

$$\theta = \operatorname{atan}2\big(V(2,m), V\ (1,m)\big)\ * \frac{180}{\pi} \qquad (3.8)$$

Where V presented rows of eigenvalues and eigen vectors from C matrix, m presents matrix of largest eigenvalues of diagonal D from C matrix, 180/π to convert the value into decree for readability. Based on theta (θ) we rotate the bounding box to correct its skewness as shown in table 3.2.

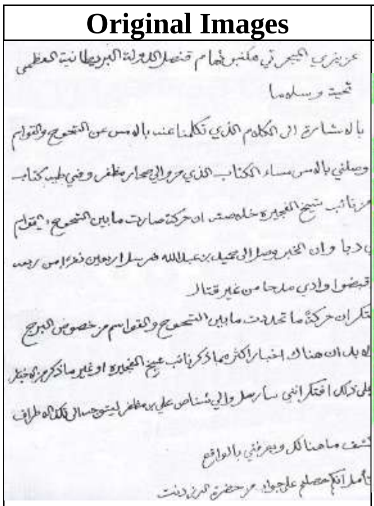
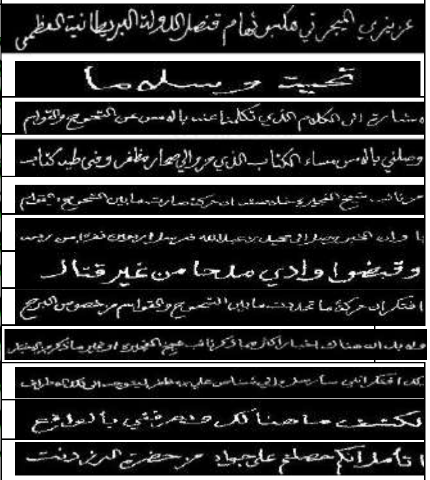
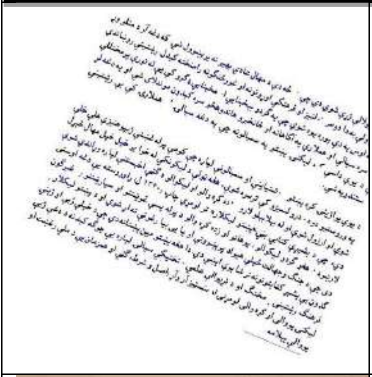
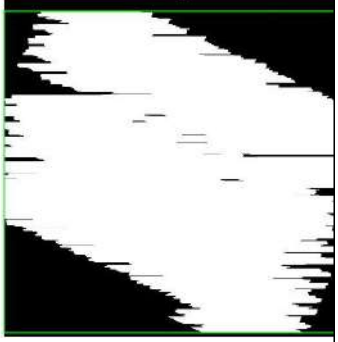
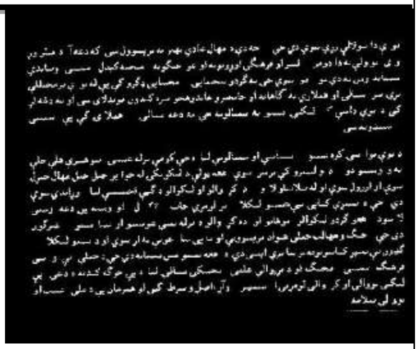
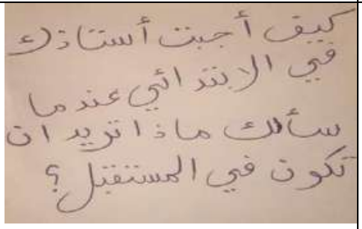
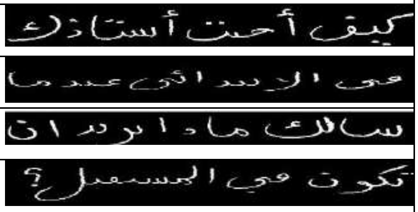
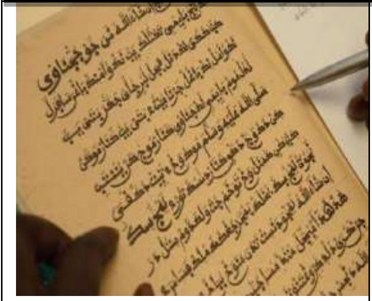
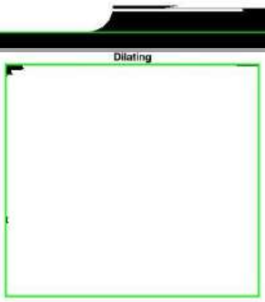
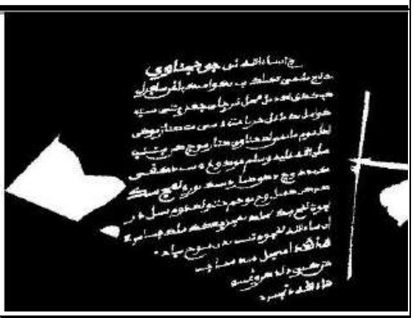
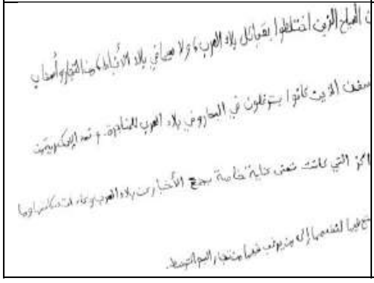
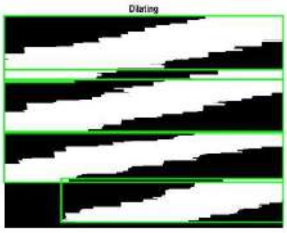
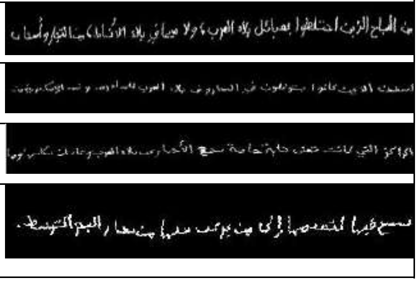
3.3.4 Experiments Results and Discussions.

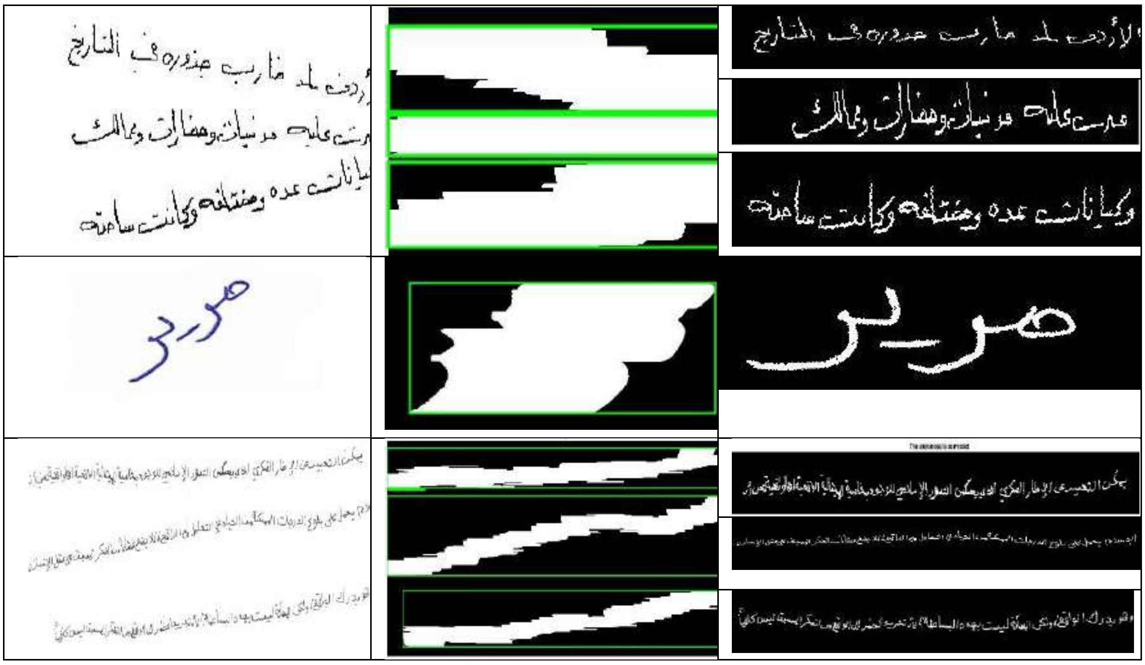
The proposed system experiments with more than 700 text documents for testing. These text documents are taken from KHAT database, old documents and our own handwritten dataset in Arabic language. Most of the collected handwritten document images are collecting that including multiple skew angles in a single document or global skew. The proposed method has provided satisfactory results such as. The algorithm has provided satisfactory with high accuracy, short time and required low memory in running comparing with previous studies.

The use of statistical method determination of skew angle is really simple and efficient when the line segmentation outputs are good the results of skew detection and correction be high accuracy. The tables 3.2 and 3.3 show and demonstrate the outputs and results of the proposed method where the input and output images are listed. Some input images have global skew therefore the line segmentation process faces problem to extract the skew line separately, instead of that the algorithm detects the slope for entire text as shown in table 3.3 and figures 3.8. Any suggested algorithm remains dependent on the process conditions with any method, otherwise, the results may be of low quality and accuracy due to the difficulty of the Arabic script, particularly the handwritten script which leads to unlimited issues and challenges (Makkar and Singh, 2012).

Table 3.3: Proposed system time for the results of some document images in figures 3.9.

Samples	**Pre-processing Per Second**	**Line segmentation**	**Skew detection and correction**	**Total Time**
Color image 1	0.110343	0.393267	2.03042	2.53403
Color image 2	0.098196	0.366027	0.660495	1.124718
Image 3	0.145516	0.420718	1.1112278	1.6774618
Image 4	0.130016	0.423822	1.669268	2.223106
Image 5	0.161941	0.432834	1.6100778	2.2048528
Image 6	0.1494	0.42276	1.636461	2.208621
Image 7	0.07607	0.333851	1.185558	1.595479
Old image 8	0.079928	0.313881	1.180443	1.574252

Original Images	preprocessing	Skew detection and correction
		
		
 كيف أحبت أستاذك في الابتدائي عندما سألك ماذا تريد ان تكون في المستقبل؟		
		
		



Figures 3.8: Results showing Arabic handwritten documents.

3.4 Size Normalization.

The size normalization of the input document images could increase the accuracy ratio of the feature extraction. According to the study in (Cheriet et al., 2007) "Normally, the character image is mapped onto a standard plane (with predefined size) so as to give a representation of fixed dimensionality for classification. The goal for character normalization is to reduce the within-class variation of the shapes of the characters/digits in order to facilitate the feature extraction process and also improve their classification accuracy. Basically, there are two different approaches for character normalization: linear methods and nonlinear methods". when all training and testing data size changed to size standardized then pixel alignment of character will have the same 2-dimensions. If the sizes of training and testing characters are not normalized before feature extraction, the size and pixel alignment of each character will diverse, which may result in incorrect feature extraction recognition. Because major intra-class disparities in word or letter length may be avoided by normalizing the training dataset, normalization maximizes the recognition rate in the training data. Figure 3.9 (a) depicts a sample character with different sizes, and Figure 3.9 (b) represents a normalized data collection of these characters, which could be 64x64, 48x48 or 32x32.

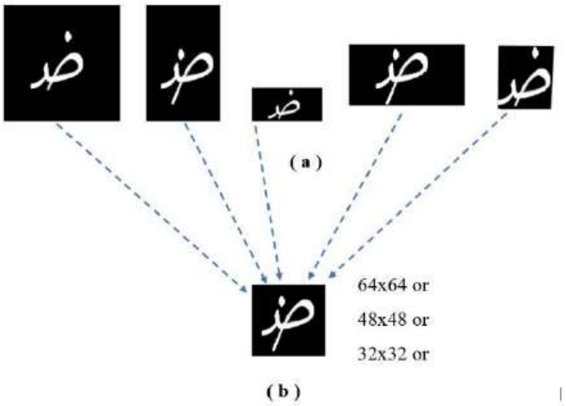


Figure 3.9: (a) different sizes of Arabic letter images. (b) Size Normalization which they could be 64×64, 48×48, or 32×32

3.5 Binarization.

Binarization is essential to focus on the characters' shapes while removing unwanted information of the background from the objects. Furthermore, there are many techniques of extractions’ feature which give better results when it works on binary images. Thresholding is the most basic method of binarization, including converting a (0-255) grayscale image to a binary image (0 or 1). Thus, saving the images with this format (0 or 1) is one of the more interesting advantages. Since there are only two potential values, it is simple to modify. Moreover, the processing time will be faster, less expensive and more compact in saving the binary images. On the other hand, the binarization may lead to loos information from the original image or it may include noise or abnormalities. Thus, the thresholding methods manipulate by two types; local thresholding methods which compute a threshold depending on the neighboring pixels. And global thresholding methods where they select one threshold for each entire image. This process is slower due to the use of local thresholding methods than global thresholding.

The Otsu method is considered the best and fastest method in converting the binary image [Gonzalez, 2009, Trier and Jain, 1995]. The Otsu method determines an optimized global threshold which depends on selecting a low point between two gray level histogram peaks from a target image. The foreground pixels have different values from the background pixels, i.e. the foreground levels differ from the background pixels in the grayscale of the same image. They are random values according to the levels of the gray level which are located in one of the two normal distributions. Vital information such as the standard deviation and variance can be calculated from these normal distributions.

Therefore, it is possible to calculate the load variance of gray levels values in the images using these two classes of pixels. Then it can also proceed to calculate the foreground pixels' variance separately from the background's variance. As the last step, the variance of the average values for each category from the overall average of all pixels is a between-classes variance. The goal of these calculations is to find an optimal threshold by reducing the ratio of between-classes variance to the total variance of the target image.

Variance within-class is calculated as follows:

$$\sigma_{\omega}^{2}(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \tag{3.9}$$

Where σ_1^2 and σ_2^2 are the variances of the two classes, ω_1 and ω_2 are the probabilities of these classes, t is a threshold used to separate these classes

According to Otsu, minimizing the within-class variance is the same as maximizing between-class variance and is formulated by the equation:

$$\sigma_b^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \qquad (3.10)$$

Which is expressed in terms of class probabilities ω_1, ω_2 and class means μ_1 μ_2.

Chapter 4

Segmentation of Arabic Handwritten Characters

4.1 Preamble

The segmentation stage is critical in OCR system. It often has three levels to segment the text document. Line, word segmentation and character segmentation which the most difficult. The problems in Arabic handwritten documents make segmentation a complicated task. The nature of Arabic handwriting, ligatures, wavy lines, touch and overlap and challenges are common to complex. We have previously explained other problems that have led to unsatisfactory segmentation results (Dave, 2015). We propose a segmentation model for Arabic handwritten text based on levels, lines, words and character segmentation. In addition, we suggest a hybrid method to solve interconnection, overlap and touch problems and improve the hashing process for Arabic handwriting. Recently, electronic devices and modern technology have become an important and essential part of human life. A lot of efforts and times were spent to protect and maintain valuable historical documents, letters, and books into digital images for scientific, service, and future uses. Optical recognition systems appeared

Some parts of the material in this chapter have appeared in the following research papers:
Boraik, Omar Ali, Ravikumar M. and Mufeed Ahmed Naji Saif. "Characters Segmentation from Arabic Handwritten Document Images: Hybrid Approach.", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 13, No. 4, pp.395-403, 2022.

as significant tools to avoid the loss of such valuable documents which convert text images into editable digital texts. There are various uses of machine learning techniques in OCR systems, the gap is still largely due to unlimited obstacles in Arabic handwriting. Comparing OCR to Latin and other languages with the recognition of Arabic, the Arabic recognition system is still incomplete and unsatisfactory. In today's world, interested parties in the field of documenting are required to save digital images possible to modify, i.e., repairing deterioration in historical books, correcting errors, using a text part of it in other applications.

Converting text images into editable digital forms is called Optical Character Segmentation (OCR) (Qaroush et al., 2019). The text images are either printed or handwritten. The deficiencies of Arabic handwriting OCR systems are more complex than printed and incomplete. Because handwriting does not abide by the font's criteria, specific size, different font and size style writing for a word repeated several times in the same document itself, other problems of interconnection, overlapping and touching, and difference gaps among word/sub-words increase the complexity of Arabic handwriting. There are common factors that make Arabic handwritten as well as printed text complex such as Arabic nature cursive, writing from right to left, connecting the letters, and so on.

Arabic handwritten character recognition has the same situation as other languages. In some cases, it seems to be more complex depending on the language, the challenges it faces, and difficulties for line, words [Jindal and Jindal, 2015, Keisham and Dixit, Louloudis et al., 2009, Ullah et al., 2019] and the challenges in character segmentation of the input document images. These challenges, such as interconnection letters word, cursive overlapping, touching existence of ligatures, diacritical marks and the position and number of dots above or under some letters. These challenges may lead to misclassification and unsatisfactory results. However, some of these challenges were recognized by many researchers in the OCR field using machine learning techniques (Jindal and Jindal, 2015).

The deficiencies are based on the failure of the Arabic character segmentation stage. Cursive, overlapping, and unrestricted writing challenges are the most long-term barriers to correct segmentation. A study (Qaroush et al., 2022) presented a projection profile technique for Arabic characters segmentation; which was tested successful in respected to Arabic database with various sizes, styles, and font types. But it is limited to the Arabic printed document. The proposed system fails to deal with handwriting documents.

Another study (Ullah et al., 2019), presented a solution for overlapping and touching Arabic characters segmentation by overlapping set theory and contour tracing. It is low accurate when segmenting the multiple touching letters. While (Shamsan et al., 2017) suggested a hybrid method focusing on the middle point of the word area. This study was focused on handwriting documents in the Hindi language. This method succeeded in fragmenting the multi-touching letters, and to apply it to the Arabic language; it needs to be developed because the Arabic calligraphy starts from right to left.

The segmentation solutions for single/ multiple touching, overlapping challenges, and interconnected characters problems still need to be expanded to include more Arabic handwritten documents. Achieving progress in respect to Arabic text recognition is hindered by such obstacles and challenges. Moreover, the complexity of finding a solution to the segmentation of overlapping and touching Arabic handwritten words or characters made few researchers interested in addressing these two problems and developing techniques to address these complex problems. These two challenges created gaps in attempts to process them, i.e., between every two research works two or more years [Ouwayed et al., 2009, Belaïd and Ouwayed, 2011, Aouadi, N. et al., 2013, Aouadi and Kacem, 2017, AbdAllah, N., and Viriri, S. 2021, Ahmed et al., 2022 as a review survey]. These are some of the motivations that encouraged us to propose a hybrid approach to reduce these complexities and improve segmentation techniques. Hence, this chapter aims to enhance and increase the accuracy ratio of Arabic handwritten character segmentation while dealing with overlapping, interconnected, and touching Arabic handwritten documents. This research work was divided into eliminating non-textual appendages in documents, segmentation, and extracting words from the image using connected components and thinning techniques. We used a hybrid method with computation analysis of the word's region to segment a word into characters. The middle point is detected to extract the structure features for dealing with the input, which contains the touching overlapping character and distinguishes from isolated character based on calculating the vacant space index value. The hybrid method has been proven to be a flexible and efficient method to deal with various renewed databases [Zeki, 2005, Kang and Doermann, 2011, Hamid and Haraty, 2001, Farulla et al., 2017, Belaïd and Ouwayed, 2011].

4.2. Proposed Methodology

This section describes the methodology of the proposed work. Initially, the pre-processing is performed for the input image to improve low quality and prepare for the next stage. Then, it segments the whole document into individual words, especially if the input image contains wavy lines. If the input image includes shapes or signatures, the approximate polygon methods remove these shapes. In contrast, the signatures are extracted and removed from the document based on the connected component analysis. Finally, divide the words into isolated letters.

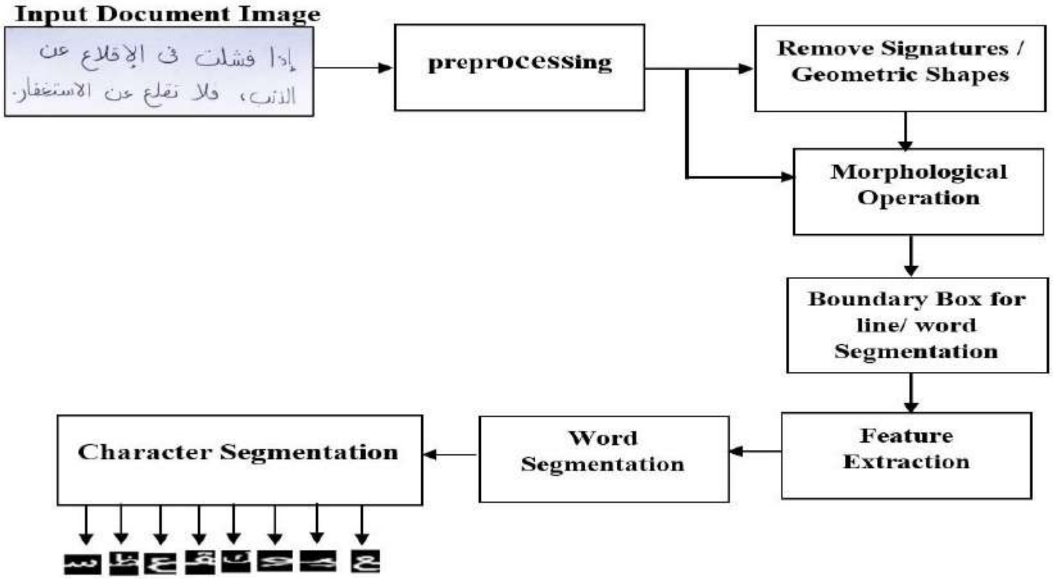


Figure 4.1: Block diagram of the proposed method.

4.2.1 Pre-processing.

Successful beginnings of any system lead to better accuracy of performance and achievement, so it is necessary to enhance the input document images as the beginnings of the first stage of Arabic OCR system processing stages. The first stage is pre-processing of the system. We have earlier explained this stage in detail and reviewed the problems and challenges in Arabic handwritten documents in the third chapter of this research work. These challenges hinder pre-processing techniques from improving poorly input images to reach the perfect resolution. We also reviewed the methods and solutions used to improving targeted Arabic scripts and removing non-textual and redundant components from the entered images. This section proposes an algorithm to divide the Arabic handwritten text document into lines, words, and then into characters. We also proposed solutions to address the interconnection problems, overlapping, touching and wavy in Arab documents. In addition to the database, two Arabic databases were used to test the proposed method IFN/ENIT and KHAAT. Due to the binary images in these two databases, the techniques used in the pre-processing are less, and the focus has been on removing impurities, morphological operations, skew detection and correction, normalization and thinning techniques.

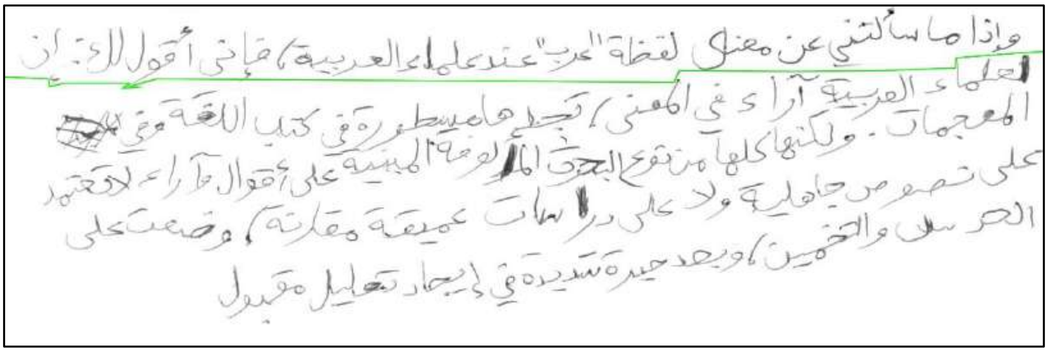
It should be noted that correcting the slant of the entire Arabic handwritten script has a great effect on line segmentation correctness. The thinning technique and morphological operation processes help in solving the touching and overlapping problems between lines. Currently, there is no effective solution if the lines are wavy and overlapping vertically. Therefore, in this research work, we have segmented Arabic documents into direct words if documents contain wavy and contiguous lines so that it is impossible to divide the document into lines as shown in figure 4.2. And the method of segmentation of words depends on the connected components and computational analysis of the word’s region.

4.2.2 Line Segmentation.

Separating the input images into isolated lines is successful with high accuracy in documents which may contain printed, historical or handwritten texts, these documents' lines have gaps between them that are orderly spaced (Radaideh and Rahim, 2016), and the handwriting is on lined sheets. These gaps are ordered and they may be almost empty of noises or unimportant objects. Because the feature of the gaps between the lines is the most important for the success of lines segmentation and extraction individually for using another application, in skew correction with multiple slanting or copy particular lines to other programs. Many handwritten documents lack clear orderly gaps between the lines, which are close, touching, and sometimes wavy, or they may be spaced at one side and touching or overlapped at the other. These challenges make the line segmentation process difficult and complex. Many studies have proposed techniques to solve these challenges and achieve satisfying success despite the convergence, contact and overlap between the lines. A study (Ahmad and Fink, 2016) proposed an A* path planning algorithm to line segmentation directly. The localization of each line is detected by two steps: Binarization method and projection profile analysis. The author's experiments tested handwritten historical documents from the MONK and the Saint Gall dataset [Brodic and Milivojevic, 2013, Surinta et al., 2014, Louloudis et al., 2009].

After it converted the input image into binarization in preprocessing phase, the input image locates the text (object) areas of the white background where the text is black. The process Uses Dilation operations to make the pixels' value of each line. This pixels' value has a single value so the line is considered one-component (Sanchez et al., 2008). The starting and ending four points of each component determine the bounding box location among these points around the line component (Hashrin et al., 2019). This boarded rectangle is segmented for each line, returned the segmented line to its original size using the erosion operation and the thinning method to give a better distance between words in preparation for their segmentation. Then the matrix of the rectangle border is saved into an image.

The overlapping and touching challenges between lines, make the dilation operation more extended, two or more lines are considered one component (Alaei et al., 2011). These challenges are solved by tracing the contour points of the touching area horizontally and overlapping vertically, using the method (direction contour tracing) in (Alaei et al., 2011). It traces the overlap path with calculation operations. This method is not good in separating lines in Arabic Handwritten documents because of the stretching of strokes and some letters are written vertically and extended such as these letters ‘أ‘,‘ک‘,‘ظ‘,‘ط‘,‘م‘. Also, most line segmentation techniques fail to segment wavy lines.



وإذا ما سألتني عن معنى لفظة "حرب" عند علماء العربية، فإني أقول إن
علماء العربية آراء في المعنى، تجدها مسطورة في كتب اللغة ومع
المعجمات. ولكنها كلها من نوع البحث الم اللوضعي المبنية على أقوال وآراء لا تعتمد
على نصوص جاهلية ولا على دراسات عميقة مقارنة، وضعت على
الحدس والتخمين، وبعد حيرة شديدة في إيجاد تعليل مقبول

Figure 4.2: Overlap and touch problems between the lines.

4.2.3 Word or Sub-word Segmentation

As it was mentioned earlier, many of the input images contain lines skewness, touching, overlapping, wavy and more closely spaced lines, where the line extraction fails, so the lines are separated into two or more lines as shown in Figure 4.3. In the study (Ahmad and Fink, 2016) Hidden Markov models were used for word segmentation from the entire document directly. Current study, the words/sub-words are segmented by applying the Connected Components (CC) method for overcoming and solving many challenges such as the distance between words is less than that between sub-words and overlapping in one word. The Boundary Box function is used to extract them. These two methods achieved a higher success rate and are better than dividing the lines before the words, as shown in Table 4.2

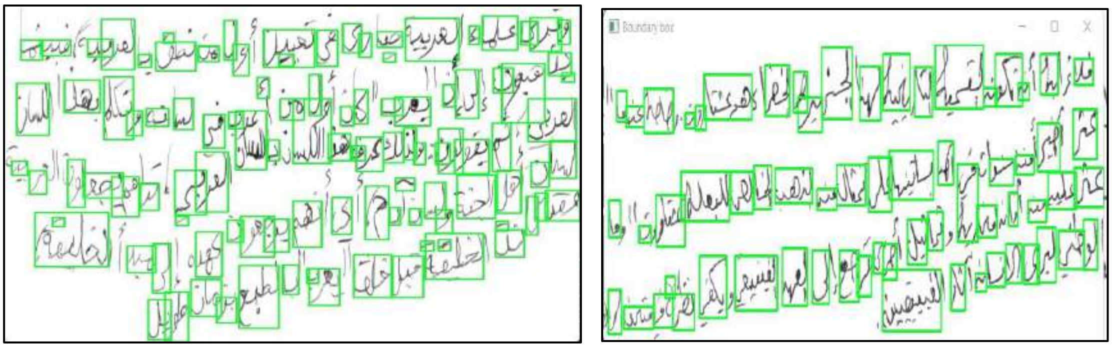


Figure 4.3: Segment the Document Image into words.

Figure 4.3 shows two samples of input images which contain the overlap touch challenges and the most obstacle is wavey between lines. These challenges make the lines segmentation process complex. Many methods were applied to segment the lines in such an image. The line segmentation methods such as projection profile method, a method based on tracing, another method based on contours, a third method based on the baseline, the fourth method based on the morphological operation or other methods. These algorithms did not successfully extract the lines, where the first line was only successfully extracted. The rest of the lines were considered one component or extracted a part of a particular line with the previous or next line in the same process. The words were extracted directly from the document with better accuracy and success. To extract words from input Arabic documents, we followed sequential steps to determine word segmentation points. These steps are:

4.2.3.1 Bounding box Detection.

After enhancing the poor input image and converting to binary, the Objects stand out differently from the background as the pixel values are different from the frontground's pixel values. These objects are connected components. Dilation operation is used to extract a line from a document; erosion operation can also extract a word or character. The extraction process is used by drawing a bounding box around the connected components (objects) region. The bounding box represents a rectangular zone where the edges are parallel to the x and y coordinate axis [Brodić and Milivojević, 2013, Tran et al., 2015]. Hence, each pixel P(i, j) belongs to the bounding box if it accomplishes the following inequalities:

$$P(I,j)|(x_{min} \leq i \leq x_{max}) \wedge (y_{min} \leq j \leq y_{max}) \qquad (4.1)$$

where x_{max}, y_{max}, x_{min} and y_{min} correspond to the bounding box's endpoints. Figure 4.4 shows the clarification of the bounding boxes of connected components. Therefore, segment every bounding box over the connected components and extract the text object. It is assigned as CC_n, where n = 1, 2, . . ., k, and k is the total number of detected objects.

Some redundant objects may be included during the extracting process for the bounding boxes of the connected components. Therefore, these redundant objects are stored temporarily because they may represent an isolated character. During the character segmentation step, the area of the bounding box region is calculated; If it is less than the average, it is eliminated.

4.2.3.2 Remove diacritical marks and dots.

The dots and diacritical marks which above or below Arabic letters are removed to avoid confusion in character segmentation and classification.

4.2.4 Character Segmentation.

This step is crucial to the segmentation stage. After overcoming the previously mentioned challenges as much as possible such as different light spots in the input image and existing shapes or signatures. After the success of words/sub-words segmentation, the character segmentation follows.

First, detecting the touching of characters: The Connected Components method is used to solve the overlapping problem between characters. It is also used to measure the weight of the character. A threshold value has been fixed to evaluate the weight. If the obtained value is less than the threshold, it can be considered a single character and split automatically. If the value is greater, it can be regarded as a touching character using the variable:

T$_c$ (2) is the aspect ratio of the touching character is greater than the character is automatically split. In order to determine the touching characters using the variable (Tc), this aspect needs to be improved due to the similarity between interconnected and touching of Arabic characters. The touching characters are defined by

$$T_c = \frac{e^g}{1+ e^g} \tag{4.2}$$

where $g = \frac{w}{h}$, w is width of the character, and h is height. After identifying the touching character, it is classified as either horizontal, vertical or multiple touching.

By comparing the two values of height and width, the type of touching is determined by $g = h1 > w1$ the touching is vertical. $g = w1 > h1$ the touching is horizontal. while the multiple touching is defined as $g = w2h2 > w1h1$.

Second, using hybrid approach by following these steps:

a- Find the area of a word/ sub-word throughout detect the start point and endpoint of this area, allocate the area on width (w) and height (h). (w, h = contour area (word)).

b- Using thinning, closing and opening operation

c- Using midpoint steps to separate word/ sub-word isolated.

- The vacant space index value is calculated on the word/ sub-word's constrained such as width and height.

- Check the previous pixel and the next one (i+1, i+2, i-1, i-2) to save the column values and check if broken character appears.

d- Saved the vacant space between the characters in array_index.

- Detect the centre value of every vacant space between the characters until the end of the word.

  $M\_point = (start_{index} + last_{index})/2$.

- This mid-point as a centre value is considered as the detection points to split the isolated character.

e- To determine the existence of joining (touching) characters, the total number of characters in a word is calculated.

f- Total value of characters is detected by the ratio of width and height.

$Total = \frac{width}{height}$.

g- Total value of characters is compared with the segmentation point.

Value of M_point = total + 1.

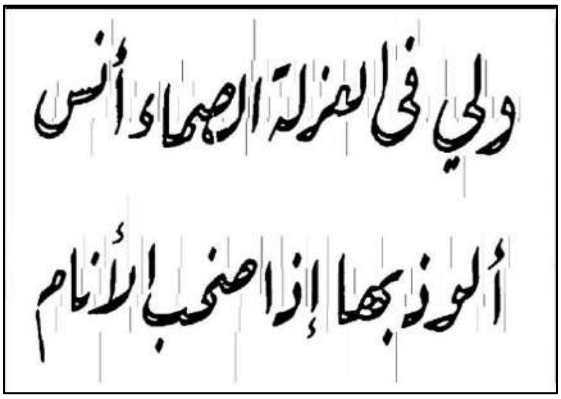
h- If joining character is exist then number of segmentation points does not exceed the total no of characters in a single word. Otherwise go to f.

f- Calculate the distance sequentially between the middle values, if the distance above 110% of height, there must be a joining character present, which could be single or multiple joining characters.

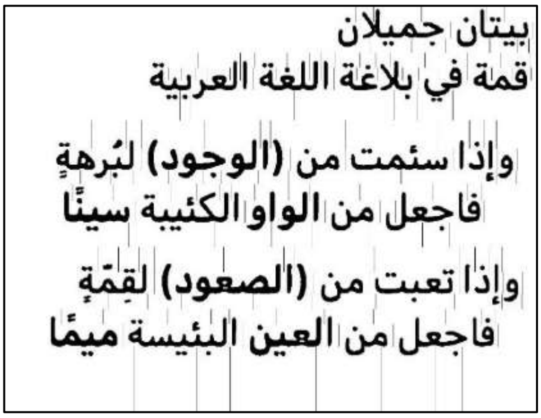
i- Using clustering method to find the cluster in identified area of importance of the character in the middle part.

j- Discover the region of importance cluster between M_point1 + 10-(M_point2 +10) to obtain the heap of pixel.

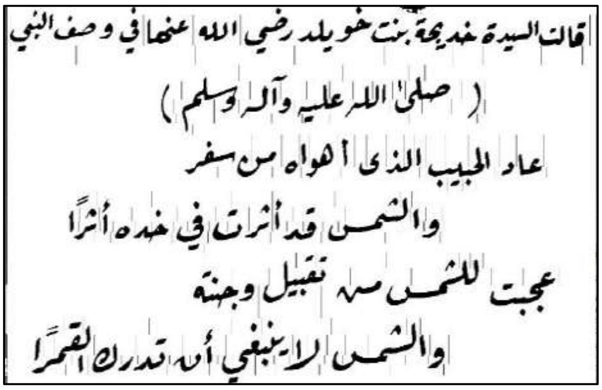
- Scan every column to determine the cluster, if pixel calculate is found to be 10 then it is considered as joining point of the character.
- By leaving three columns in a row, you can segment the joining character.
- The new segmentation points should be extracted.
- Split the word from all the segmentation points.



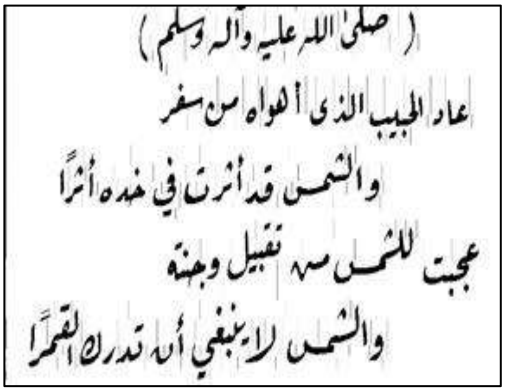
ولي فى العزلة الصماء أنس
ألوذ بها إذا صخب الأنام



بيتان جميلان
قمة في بلاغة اللغة العربية
وإذا سئمت من (الوجود) لبُرهةٍ
فاجعل من الواو الكئيبة سينًا
وإذا تعبت من (الصعود) لقِمّةٍ
فاجعل من العين البئيسة ميمًا



قالت السيدة خديجة بنت خويلد رضي الله عنها في وصف النبي
(صلى الله عليه وآله وسلم)
عاد الحبيب الذى أهواه من سفر
والشمس قد أثرت في خده أثرًا
عجبت للشمس من تقبيل وجنته
والشمس لا ينبغي أن تدرك القمرا



(صلى الله عليه وآله وسلم)
عاد الحبيب الذى أهواه من سفر
والشمس قد أثرت في خده أثرًا
عجبت للشمس من تقبيل وجنته
والشمس لا ينبغي أن تدرك القمرا

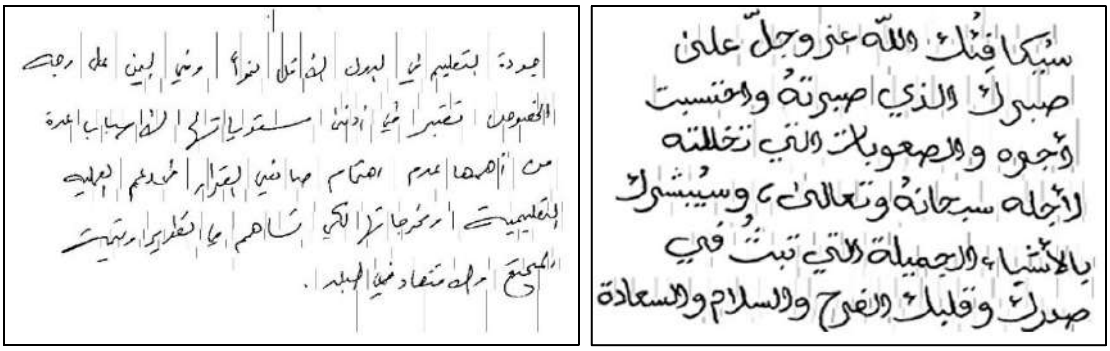


Figure 4.4: Sample images Show the Segmentation of Isolated, touching and joined Characters.

4.2.5. Evaluation Metrics.

To evaluate the performance of the hybrid approach for character segmentation. The performance measures which the evaluation equations were explained in the study (S Deshmukh, M., and R Kolhe, S. 2019); the similar evaluation strategy was also applied in this work, which uses five factors for evaluation: successful segmentation rate (SR), precision (P), recall (Re), correct segmentation rate (CS), and F-measure (Fm). They are illustrated in Equations 4.3- 4.7. These factors are figured out by counting the number of matches between the resultant segmented words and then characters by the algorithm and ground truth characters in text word segments.

$$SR = \frac{NCc}{NCr} * 100 \tag{4.3}$$

$$P = \frac{(NCc+NCo)}{NCr} \tag{4.4}$$

$$Re = \frac{(NCc+NCo)}{NCg} \tag{4.5}$$

$$CS = \frac{NCr-(NCi+NCo)}{NCg} \tag{4.6}$$

$$F_m = \frac{(Re*P)}{(Re+P)} * 2 \qquad (4.7)$$

where NCg: Number of ground truth words, characters respectively.

NCc: Number of segment correct words, character isolated.

NCr: Number of segmentation results (words, characters).

NCi: Number of incorrect segmentations for words, characters.

NCo: Number of over segmentations of touching in words, characters

4.2.6. Experimental Results and Discussion

Implementation of the proposed method: The proposed method was implemented using Python 3.8.8, Open CV environment (Spyder4 [MSC V.1916 64 bits]), Win 11 pro 64_bit OS, with Intel(R) Core (TM) i5-9300HF CPU 2.40 GHz, RAM 8 GB.

We proposed hybrid approaches to segment the Arabic handwritten document into direct words. In case line segmentation failed because some images contain touching, overlapping, wavy and closed (convergent) lines which are challenges that make the document segmentation process into lines difficult. The proposed method is tested on three Arabic databases. The first database is our database created for this research work. It was explained in the third chapter, also, the techniques and filters which used in the pre-processing stage. The removal of redundant components from Arabic documents was also already mentioned and the proposed algorithm's success rate in removing signatures and redundant components was reviewed.

Second, KHATT database contains 1000 documents, 9327 lines, 165890 words (Mahmoud et al., 2014). Third, is IFN/ENIT database involves 26459 words which are names of Tunisian cities (Pechwitz et al., 2002). Testing the proposed model on three databases was in stages: First, constant documents were segmented into lines, while inconstant documents contain overlap, wavy, or touching problems which are mostly segmented into direct words. The hybrid approach achieved high success rates, as shown in table 4.1 which figures out the distribution of success rates over the three databases. The proposed method achieved a lower accuracy rate in our database for the following reasons:

1. Poor lighting and quality problems.
2. Degradations in the document images.
3. Colored images, while the images in the other two databases are binary.
4. Our database involves more documents than the other two databases.

Table 4.1: Segmentation accuracy of words and characters.

Databases	**Number of:**		**Correct Segmentation**		**Incorrect Segmentation**		**Accuracy Rate**	
	Words	**Characters**	**Words**	**Characters**	**Words**	**Characters**	**Words**	**Characters**
KHATT	165890	589924	161411	548629	4479	41294	97.3%	93%
IFN/ENIT	26459	212211	24342	188867	2910	12732	92%	89%
Our DB	302348	1257191	275136	1111356	27211	150863	91%	88%

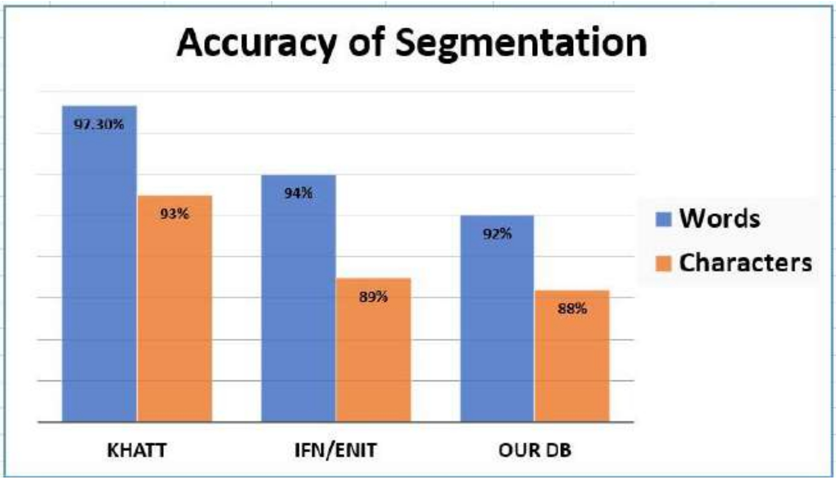


Figure 4.5: Segmentation Accuracy of words and characters for each database.

Table 4.1 and figure 4.5 show the words and character segmentation for three databases. In addition, 1450 Arabic handwritten document images were taken from the three mentioned databases. We focused on images containing overlap and touch problems of words/ letters and wavy lines. Table 4.2 explains the outputs of the proposed method in solving these problems.

Table 4.2: Segmentation accuracy of touching words and characters.

Inputs	NCg	NCr	NCo	NCi	NCc	SR	P	Re	Fm	CR
words	26468	25409	985	833	25357	99	1.00	0.99	1	89.130
characters	79404	74639	2896	2328	74071	99.23	1.00	0.96	0.99	87.4

Second: 26468 Arabic handwritten words were taken from the three databases, which contain complex touching and overlapping problems. The ground-truth value of lines and words was calculated manually in our database. The statistics of the performance measure for words, as shown in Table 4.2, indicate that the segmentation rate was 99% successful (SR); this segmentation rate is divided into over-segmentation, correct and incorrect segmentation. The accurate segmentation rate (CR) for words is 89%; Recall (Re), precision (P) and F-measure (Fm) obtained approximately 1.00, respectively. The performance measure indicates that the incorrect segmentation rate is approximately

11%; it means that the number of over-segmentations is 985 words (NCo), and 833 words are missed or incorrect segmentation (NCi). As for segmentation, the number of touching Arabic letters is 79404. These letters are either isolated from the word's origin or have single or multiple touching. Table 4.3 also indicates that 99% of the total segmented Arabic letters, 87% of the correct segmentation (CSR), and 13% of them was wrong segmentation. 79404 of touching characters were segmented. (P) was 0.96. (Fm) and (Re) were 1.00. Table 4.3 shows the types of challenges in Arabic handwriting obtained from Table 4.2.

Table 4.3: Shows the accuracy of the hybrid method in each type of inputs.

Type of Characters	**No of characters**	**Correct Segmentation**	**Accuracy**
Isolated	28401	26696	94%
Single Touching	32573	28012	86%
Multiple Touching	18430	15186	82%
Total	79404	69452	87.4%

Table 4.4: Comparison analysis.

Reference	Technique	No. of images	Types of Input	Segmentation ratio
(Aouadi and Kacem, 2017)	Nearest model selection and detect the model's parts centres	820	Segment touching lines and words.	94%
(Ullah et al., 2019)	Overlapping Set Theory and Contour Tracing	220	Segment a single touching character	97.2 %
Proposed method	Hybrid approach	huge	Segment a overlapping and single/ multiple touching characters	90%

Table 4.4 shows the comparison results between the outputs of this work with previous works that discussed such problems. On the other hand, the comparison results are difficult because of the different documents collected by researchers from various Arabic handwritten sources that involve these relevant challenges. The primary purpose is to reach satisfactory and accurate solutions even though the collected documents

differ, but the essence of the problem is common. It should be noted that the comparison results are based on the findings of the researchers' works.

4.3. Challenges

Arabic characters, like other cursive languages, can be combined or changed in shape depending on their context. A character's appearance is affected by its relation to other characters, the font used to render the character, and the application or system environment (Alaei et al., 2011). An Arabic ligature is a compound of two or sometimes three characters such as (Lam Alef ﻻ). Ligatures unfortunately complicate the segmentation task of any Arabic Optical Character Recognition (OCR) system. In our system, ligatures are treated as isolated characters.

4.4 Conclusion

Finding solutions to the endless and volatile problems in Arabic handwritten characters segmentation is gradual, to improve and increase the efficiency of the recognition system. Every recent segmentation work is a continuation of previous works' failures. Overlap and contact problems in long-term Arabic handwriting. Therefore, a flexible hybrid approach has been proposed for Arabic handwritten character segmentation which is overlapping or multiple touching. It also aims to improve the efficiency of the segmentation stage. The proposed method has been tested on Arabic handwritten databases which contain more complex challenges than the previously existing ones. The results showed that the proposed method was highly efficient in word segmentation. And it is an effective, feasible and flexible approach in the segmentation of interconnected, overlapping or multi-touching Arabic characters. But errors or over-segmentation may occur in multi-touched characters.

Chapter 5

Recognition of Arabic Handwritten Characters Using CNN

5.1 Preamble

Arabic language is spoken by more than 447 million in the middle east region, also known as official language in various sectors like education, media and government offices (Muaad et al., 2021). Arabic handwritten characters recognition is one of the most challenging and complex tasks in computer vision. The task of Arabic handwritten characters recognition can be applied in the banking system for checque text recognition, postal code recognition and plate Number detection, etc. [Almisreb et al., 2022, Sushma and Lakshmi, 2020, El-Sawy et al., 2017]. High accuracy is achieved when applying successful Deep CNN models to handwritten character classification and recognition. Researchers who are developing Optical Character Recognition (OCR) systems are expanding text recognition to reach the main goals of obtaining high accuracy, faster process (Aneja and Aneja, 2019), better achievement, and getting an error percentage close to zero, like the studies [Awni et al., 2021,

Some parts of the material in this chapter have appeared in the following research papers:

Boraik, Omar Ali, M. Ravikumar, 2022, “Arabic Handwritten Character Classification and Recognition Using CNN and Transfer Learning Approach”, The Seybold Report Journal (TSRJ), Vol.17, No.11. (Scopus Indexed).

El-Sawy et al., 2017, AlJarrah et al., 2021]. Thus, researchers resort to using existing databases to compare the developed system with previous systems, along with the possibility of expanding the quantity of the databases such as the public new dataset in (Khan, 2022) for various purposes.

Many challenges hinder the process of text recognition systems, such as the great similarity among Arabic letters. The difference in Arabic letter shapes, based on their position in the word, makes complexity in classification. Each letter has two or four forms (isolated - beginning of the word - middle – and end). In addition, there is a difference in the way the letter is written. Place and number of dots, which is an integral part of the letter's body, make it completely different in pronunciation. Another challenge that makes Arabic handwritten text recognition techniques complicated is that it leads to misclassification and unsatisfactory results (Younis, 2017).

Since 2012, machine learning and deep learning techniques have dominated the handwritten text recognition of various languages (Aneja et al., 2018) (online, machine-printed or handwritten) and achieved better and faster performance in dealing with huge databases despite the challenges and difficulties in each language [Lawgali, 2015, Noubigh et al., 2020, Yousif and Shaout, 2014]. As a result of this, better, faster, and less error-free performance, researchers' interest has increased overtime in developing various models and architectures for handwritten text recognition task, based on the simple Convolution Neural Networks (CNN) idea. The structure of the particular method still differs in layers. The pre-trained neural network represents some of the most effective convolutional neural networks for

ImageNet challenging through transfer learning, such as extracting and tuning associated features (Jiang et al., 2021).

Transfer learning that is introduced in this work via deep learning approach which employs a pre-trained models that were trained for different tasks as a starting point for another. As a result, it is a learning optimization strategy that enhances the implementation of the second task. Nevertheless, it only performs if the model features learned in the first task are general. Transfer learning may be applied to another task via a pretrained approach in which one of the approaches accessible from different research organizations is selected as a starting point to regulate it for the other similar challenge. [Goularas and Kamis, 2019, Pouyanfar et al., 2018, Wang, 2019]. This work compares the outputs of pre-trained learning transfer approaches in classifying and identifying Arabic handwritten characters. The points listed below give a summary for the main contributions of the current study:

1- A comprehensive study is conducted through CNN models and transfer learning approaches to solve misclassification problems of some similar Arabic characters.

2- Two various databases (public and private) for Arabic handwritten characters are used to evaluate the proposed deep learning approaches.

3- The proposed work investigates the best Transfer Learning Approaches for classifying and recognizing the Arabic handwritten characters.

5.2 Classification

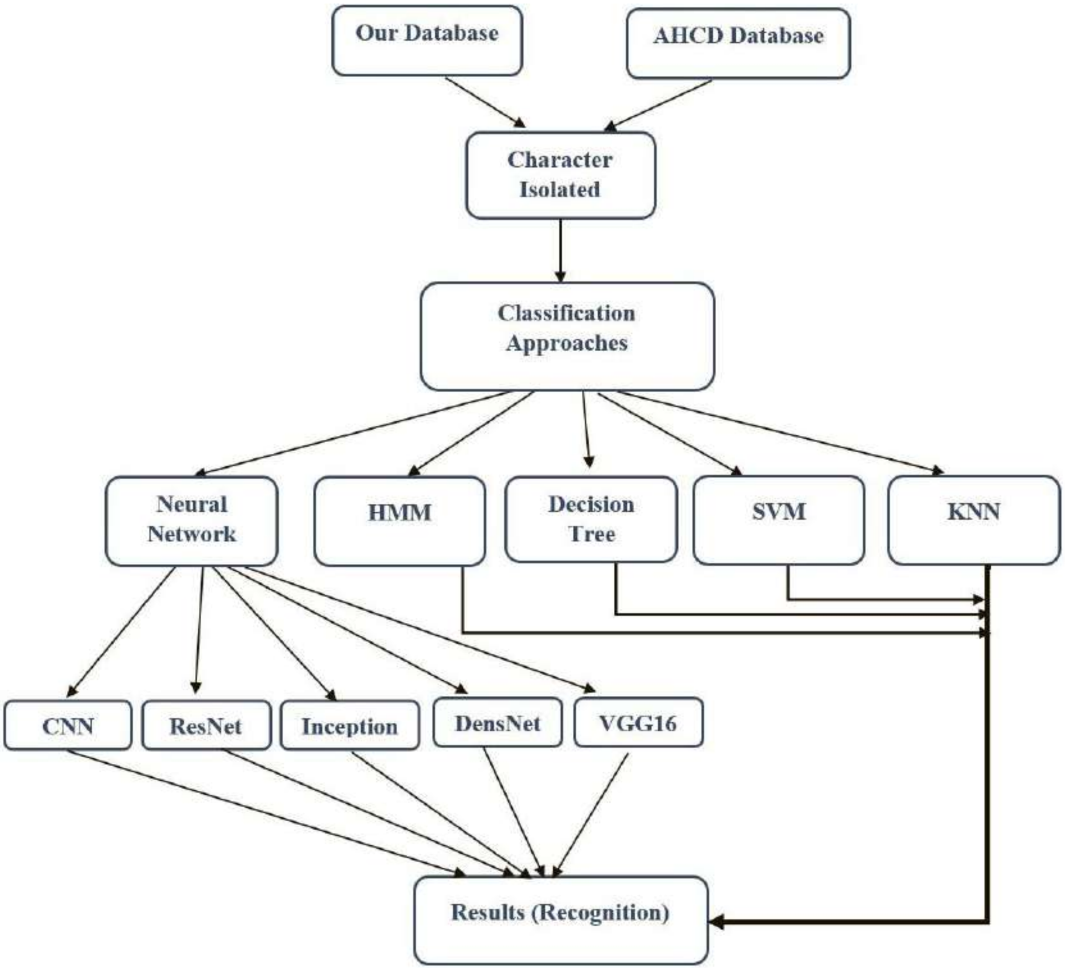


Figure 5.1: Different Machine Learning Approaches are used in this Work.

There are many machine learning algorithms used in classifying the input images as shown in figure 5.1. This has resulted in complexities in the Arabic handwritten characters regardless of the better performance and accuracy (Almisreb et al., 2022) in the classification that is provided by deep learning techniques. This chapter introduces many techniques used in deep CNN that is deep and comparisons between the results of each model to reach the primary objective of choosing the optimized model to classify the image into its appropriate output category accurately. The most popular technologies used in the classification process in OCR systems are SVM (Shams et al., 2020), KNN (Athoillah, 2019) and Random forest (Rashad and Semary, 2014).

Researchers turned into deep CNN techniques and transfer learning in extracting features and recognizing the characters in the last decade.

5.3 Materials and Methods:

5.3.1 Preparing Arabic Handwritten Characters database

Available datasets for classification and recognition purposes are images of isolated Arabic handwriting, which were extracted from the segmentation process as explained in Chapter 4. In addition, images of Arabic handwritten characters were added from various sources to the used dataset in this chapter as we referred to in the third chapter of this study. The proposed model is fed with 131,000 images of Arabic handwritten characters. These characters' images are classified into 28 classes depending on number of Arabic Alphabet, which is 28 letters. Each class has almost 4190 character images, which involve the four shapes depending on their position in the word. Figure 5.2 shows samples of these characters. The classification model was fed in two steps: First, the number of images was reduced to 13446, split into 80% for training, 10% for testing, and 10% for validation. Second step, the number of images is 131,000 images of characters with the same split, 80% training, 10% testing and 10% validation.

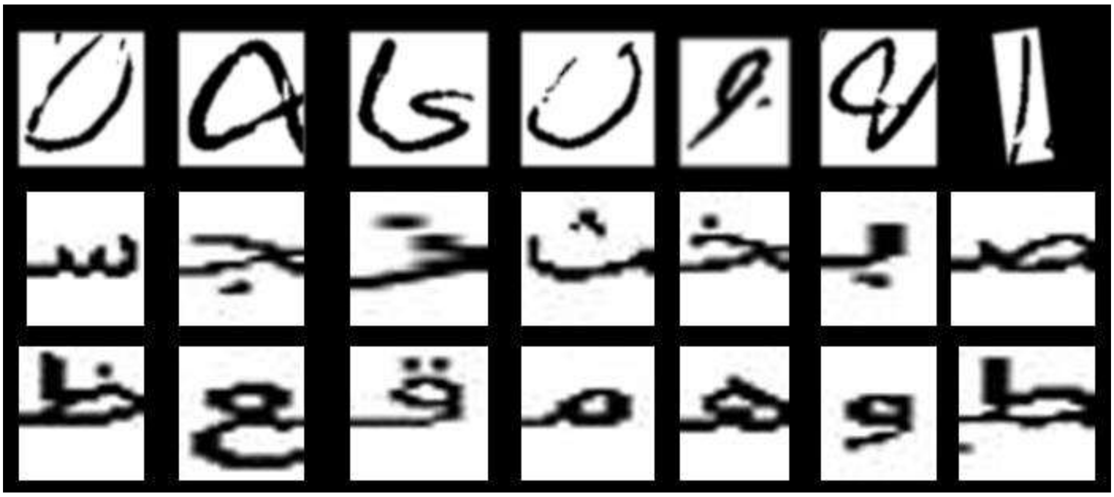


Figure. 5.2: Sample Images of Arabic handwritten characters isolated.

ImageNet Dataset

ImageNet dataset is used as source data to evaluate the four deep CNN approaches. ImageNet is a dataset that contains nearly 15 million fully annotated images. This dataset was created by collecting WordNet's synonym sets for each term. This synonym set equates to 14,197,122 labeled photos. In ILSVRC contests, a subset of ImageNet is used.

5.3.2 Proposed the method of CNN

Data and functions in a CNN have extra structure. Images are used as input data for $\chi 1$; $\chi 2$, ..., χn. Essentially, the input image to a convolutional layer is M×M×C image information, where M represents the image's height and width, M is the number of pixels in the image, and C represents the number of channels per pixel. Three channels of RGB image C = 3, while C=1 for a grayscale image's one channel. A CNN comprises numerous layers, including convolutional layers, pooling layers, and fully linked layers. The convolutional layer is made up of K filters (kernels) of N×N×R, where N is the height and width of the filter (kernels) and R is equal to or less than the number of image channels C and may vary for each filter (kernel). Figure 5.3 depicts the filter (kernel) convolved with the image to yield M×N+1 k feature maps. If the input is 28×28, the max-pooling output is 14×14. As illustrated in Figure 5.4, each feature map is then pooled, often utilizing choose maximum pooling across q×q, where q is the maximum value of inputs. Following the convolutional and pooling layers, any number of fully connected layers may be added as in a normal multi-layer neural network.

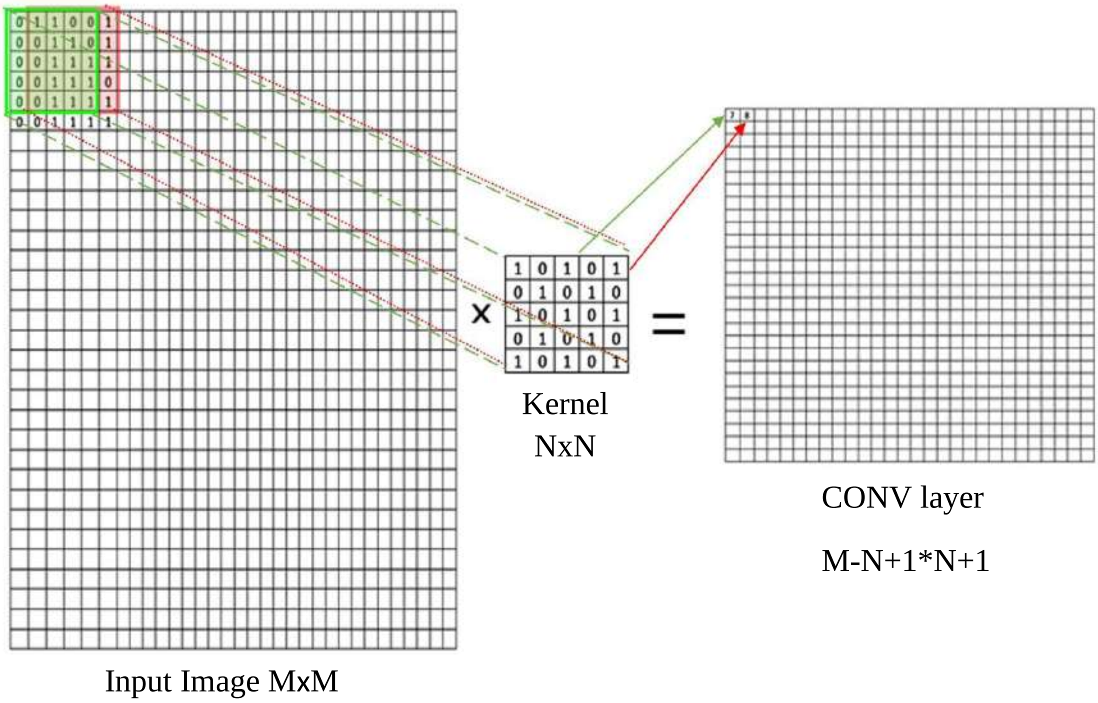


Figure 5.3: Shows convolution steps on an input image with M×M size and multiplied with N×N size of the kernel.

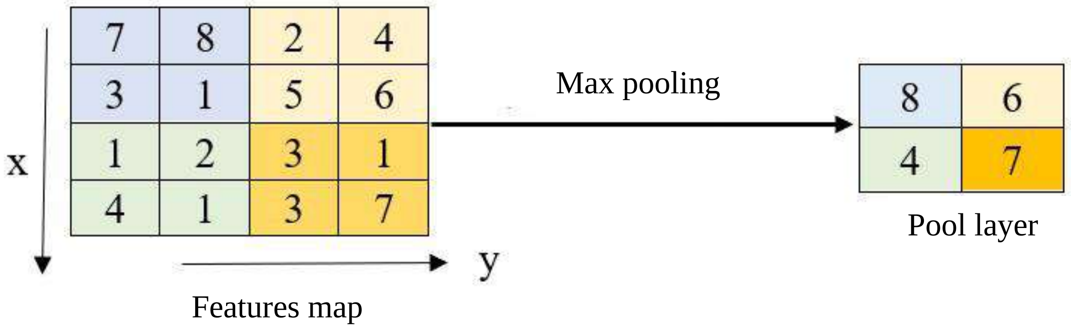


Figure 5.4: Shows the pooling process by selecting the largest value of the feature map to reduce the size.

Layers

a) Convolution Layers.

If W is a filter with an N*N kernel. The filter W is used on a convolutional layer L followed by M*M square neuron nodes the of the convolutional layer output will be defined by (MN+1) (MN+1), resulting in the k-feature maps shown in Figure 5.3. The convolutional layer functions as a feature extractor that extracts prominent input characteristics like endpoints and edges, layer l1 is used to calculate the pre-nonlinearity input to a unit. is then calculated as follows:

$$Y_i^l = B_i^{(l)} + \sum_{a=1}^{M} \sum_{b=1}^{M} W_i \, X_{(i+a)(j+b}^{l-1} \qquad (5.1)$$

where Bi(l) : bias matrix, Wi(l): filter of size N × N. Then, the convolutional layer uses its activation function as:

$$Z_i^l = \sigma \left(Y_i^l\right) \qquad (5.2)$$

In the proposed model, the activation function Rectified Linear Unit (ReLU) is applied with non-saturating σ(Y_i^l) = max(0; Y_i^l). The activation function ReLU is used on the output of every convolutional layer and fully connected layer. The ReLU (Jaderberg et al., 2015) improves the nonlinear features of the decision function and the entire structure without changing the receptive fields of the convolution layer.

b) Pooling Layer

The pooling layer may follow convolutional layers in CNN. We get fewer parameters and hence fewer calculations by lowering spatial dimensions. Another benefit of pooling layers is that the output feature maps from the convolution layer are generalized, making the classification insensitive to direction changes and distortion effects. Pooling is accomplished by dividing the convolution layer's feature map into zones and subsampling it as a single output for each zone. There are two ways for subsampling: taking the maximum or the average of the pooled zone. Using max rather than average is a bulge and noticeable in the previous studies. The max-pooling is done entirely within a 2×2 pixel window.

c) Fully-Connected Layers

Following numerous convolution and subsampling layers, the output features are flattened into dense (fully connected layers) producing features piecing into recognizable objects. Every neuron in one fully connected layer is coupled to every neuron in the next layer. Classifiers are trained in the fully connected layer, where each neuron conducts a sum of dot products between previous layer inputs and weights; then, the activation function SoftMax is usually used in this part to output the prediction. Figure 5.5 displays the suggested architecture in the proposed model before applying the Deep CNN architectures.

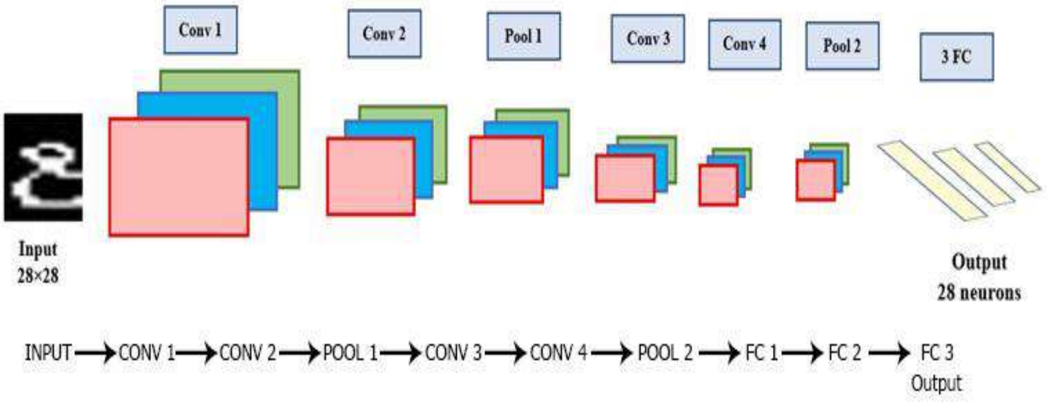

Conv 1
Conv 2
Pool 1
Conv 3
Conv 4
Pool 2
3 FC
Input
28×28
Output
28 neurons
INPUT → CONV 1 → CONV 2 → POOL 1 → CONV 3 → CONV 4 → POOL 2 → FC 1 → FC 2 → FC 3
Output

Figure 5.5: Architectures of CNN for Arabic Handwritten Characters.

It is to be pointed out here that two subsequent convolutional layers were implemented, each one with 16 kernels, followed by a pooling layer of 2 × 2 and stride 2. All of this is followed by two successive convolution layers, each one with 32 kernels and the last pooling layers are 2 × 2 of stride 2. The output of these layers is then flattened and squished in FC1 and FC2, with the final FC layer (output layer) containing 28 neurons, which correspond to the number of distinct classes of Arabic alphabet. The last fully connected layer is the input classifier.

5.4 Transfer Learning

A dataset may be not sufficient to train CNN neural network models, obtained misclassification of the characters or inaccuracy desired of character recognition. transfer learning approaches can be applied to character classification or recognition with high performance. through training process on an extensive dataset is to reuse the model weights from previously trained models. The trained model was developed for standard computer vision data sets such as ImageNet recognition tasks. ImageNet is a project which aims to provide an extensive database of images for research purposes. It contains more than 14 million belonging to more than 2000 categories. The ImageNet challenge is the de-facto standard for computer vision classification algorithms when it comes to image classification (Russakovsky et al., 2015). Networks exclusive to this challenge have been dominated by CNN and deep learning technologies since 2012. The modern, pre-trained Of CNN models included in Keras's library represents some of the best performing CNN on ImageNet challenge through transfer learning. Such as extracting associated features and adjusting the weights.

In general, transfer learning refers to a process in which a model is trained on a specific problem. This trained model is used in another task to solve a related and similar situation. In deep learning, transfer learning is the best way to introduce a CNN architecture, then retrain another task to solve a similar problem. The benefit of transfer learning is that it reduces the training time of the deep learning model (Aytar and Zisserman, 2011), reduce misclassification rate. The weights of the pretrained model in reused neural networks can be used as a starting point for the training process and adapted to the new problem. The Weights in reused neural networks can be used as a starting point for the training process and adapted to the new problem. This way may be helpful when the related problem contains much more labelled data than the problem of interest. The simple CNN model proposed and compared with VGG, Resnet and Inception V3 models are used to handle the task of classification of handwritten Arabic characters.

5.4.1 Transfer Learning Methods

There are three ways through which transfer learning can be achieved: Given a source model that has already been trained with ImageNet dataset A, a target model can be trained with dataset B by the following:

- Instead of randomly initializing the weights, use the source models architecture with weights of A dataset.
- The source model can be used as a feature extractor by replacing the layer with the number of classes in dataset B.
- Using the source model with some of the layers frozen, generally the initial layers with more available features and retraining the last layers with more specific features for dataset B. The generalization needs to be identified when the specificity becomes evident. To accomplish this, start freezing layers from top to down.

Four different CNN architectures, DenseNet121, ResNet50, Inception V3, and VGG16, were used to implement the transfer learning process. They (refers to what?) were previously trained on the ImageNet database. Then the saved weights were used to train the models on Arabic handwritten characters. The Arabic character images were classified into 28 classes according to the number of the Arabic alphabet is 28.

The various CNN architectures used to classify Arabic handwritten characters are discussed in this part. Using Keras library which includes the CNN architectures.

Table 5.1: Shows the parameters inputted into each of these architectures.

Arguments	**inputs**
Image size	28
Batch	64
epoch	5
Include_top	Fully connected
weights	Pretrain on ImageNet ‘imageNet’
Input shape	Image_size, image_size, 3 channel
pooling	Global max pooling ‘max’
Classifier activation	softmax
Number of classes as output	28 classes (28 Arabic alphabet)

5.4.1.1 DenseNet 121 (Huang et al., 2017): Mainly, in the DenseNet, it is noticed that it commences with a convolution representation of 7×7 and 3×3 that represents pooling block maximization which precedes four dense blocks, three transition layers and a classifying layer that includes a 7×7 global average pooling and a 1000-D layer that is fully-connected. Regarding the DenseNet 121 architecture, four dense blocks with a convolutional layer count of 6, 12, 48, 32 can be noticed. Each of the dense blocks has its layer count. Transition blocks keep the transferred feature maps manageable among the four dense blocks. Utilizing a 2×2 average pooling with a stride of 2, the number of feature maps and their sizes are reduced by half using a 1×1 convolution.

5.4.1.2 ResNet-50 (He et al., 2016): ResNet-50, ResNet-18 and the ResNet-101. The authors in (He et al., 2016) proposed ResNet architecture. The ResNet is a neural network constructed from pyramidal cells in the cerebral cortex. The ResNet was applied to skip connections or shortcuts to move through many layers. The typical ResNet model was developed by skipping two or three layers, including nonlinearity

(ReLU) and batch normalization. ResNet also introduced the concept of residual learning to Neural Networks (NN) to transform the competition in the NN architecture and add effective ways to train deep networks. For example, ResNet18 is a pre-trained deep learning model for image classification. The network was 18 layers deep and trained with 1 million images based on 1000 categories. ResNet50 is a pre-trained model for image classification. The network was 50 layers deep and trained with 1 million images in 1000 classes. ResNet101, on the other hand, is a pre-trained deep learning image classification model with 101 layers trained on 1 million images in 1000 categories.

The ResNet-50 approach begins with a Convolution. A block consisting of a 7×7 convolution layer with 64 output channels followed by a series of layers. Batch normalization with step size 2, softmax, and 3×3 max pooling. Four modules are stacked in the center, each one has two remaining blocks. The remainder identity block applied to the output receive field is precisely the same dimension as the input receive field. The convolutional remainder block applies to the input receive field in the same size. The convolutional remainder block uses a 1x1 convolution operation to match the dimensions between the inputs and outputs of the remaining blocks. The last layer is adaptive mean pooling, followed by a fully connected layer where the number of neurons is the same number of output classes.

5.4.1.3 Inception V3: Inception V3 is an improved version of the fundamental model Inception V1. It was introduced in 2014 as the GoogleNet model. Inception V3 is a pre-trained CNN with 22 ImageNet-trained layers. The network design comprises a 1 to 1 Convolution layer at the network's center. Furthermore, global average pooling was utilized at the network's conclusion (Szegedy et al., 2015).

5.4.1.4 VGG16 (Sushma and Lakshmi, 2020): VGG16 (visual geometry group) is a CNN model presented by Simonyan and Zisserman (Jaderberg et al., 2015). This model achieves the top-five test accuracy of 92.7% on ImageNet dataset. This dataset includes 1000 classes with over 14 million images. The name VGG16 comes from the fact that there are 16 layers, including a convolution layer, a max-pooling layer, an activation layer, and a fully connected layer. It has 13 convolutional layers, five max-pooling layers, and three dense layers, which sum up to 21 layers, but there are only 16 weight layers, finally, VGG19 is a pre-trained deep learning model for image classification. The network consists of 19 layers and trains with 1 million images from 1000 categories in the ImageNet database. Simonyan and Zisserman (2015) point out that VGG19 has 19 layers, specifically 16 convolutional layers, three fully connected CNNs with 3x3 filters with 1 step size and padding, and vice versa. There is a 2x2 maximum pooling layer, data augmentation was also developed and used as additional dataset generated from the existing images. As for data augmentation, two methods were used: augmentation by mirroring or creating a mirror image, and augmentation by random crops. The experimental results and analysis will be discussed in the next section.

5.5 Results and Discussion

This section illustrates details concerning the proposed approaches' implementation and evaluation. The proposed approaches were developed using Python 3.8.8, Google Colab, and the deep learning frameworks Keras and Tensorflow. Win 11 Pro 64 bit OS, Intel(R) Core (TM) i5-9300HF CPU 2.40 GHz, 8 GB RAM.

5.5.1 Evaluation Metrics

To evaluate the performance of proposed Arabic handwritten character recognition system based on confusion matrix, calculation of matrices like accuracy, precision (also called Positive Predictive Value (PPV)), Specificity (SPE), recall (called Sensitivity (SEN)), F1-score, Mathews Correlation Coefficient (MCC), and Negative Predictive Value (NPN) were used. These indices have the following mathematical definition:

$$SEN = \frac{TP}{TP+FN} \tag{5.4}$$

$$SPE = \frac{TN}{TN+FP} \tag{5.5}$$

$$PPV = \frac{TP}{TP+FP} \tag{5.6}$$

$$F1\_score \ = \frac{TP+TN}{TP+FN+TN+FP} \tag{5.7}$$

$$NPN = \frac{TN}{TN+FN} \tag{5.8}$$

$$MCC = \frac{(TP \times FN)-(FP \times FP)}{\sqrt{(TP+FP)\times(TN+FN)\times(TP+FN)\times(TN+FP)}} \tag{5.9}$$

$$ACC = \frac{TP+TN}{TP+TN+FN+FP} \tag{5.10}$$

where TP indicates to true positive

TN indicates to true negatives.

FP indicates to false positives.

FN indicates to false negatives. These parameters can be calculated using confusion matrices for both binary and multiclass problems [Al-Antari et al., 2018, Al-Antari et al., 2020, Al-Antari et al., 2021, Al-Masni et al., 2018].

5.5.2 Results

The proposed CNN model with Pertained models ResNet 50, Inception V3, DenseNet 121 and VGG16 are implemented on the collected dataset. The experiments have been explained and discussed in two stages: first, the results of CNN were proposed on our database, and secondly, the description of experiments has been applied on learning transfer approaches re-trained on our dataset.

First, regarding the CNN model results on the AHCD dataset and our dataset, in which the total images are 131000 character images. The two datasets are split into 80% training, 10% testing and 10% validation. Arabic characters were also represented with sequential numbers of (0 - 27), where 0 represents ‘أ’, 1 represents ‘ب’, 2 represents ‘ت’, and so on. To facilitate the carry-out process, using the equation (5.4), (5.6) and (5.10), the proposed CNN model obtained a test set accuracy of around 88 %, precision (Positive Predictive Value (PPV)) of 87.88%, Sensitivity (SEN) of 87.81%, and an F1-score of 87.8%, shown in table 5.2.

Table 5.2: Experimental results of CNN for each character.

NO	Character	PPV	SEN	F1_score	No	Character	PPV	SEN	F1_score
0	أ	0.99	98.	0.98	14	ض	0.84	0.86	0.85
1	ب	0.92	0.97	0.94	15	ط	0.92	0.92	0.92
2	ت	0.89	0.89	0.89	16	ظ	0.93	0.92	0.93
3	ث	0.90	0.87	0.88	17	ع	0.79	0.79	0.79

4	ج	0.89	0.92	0.91	18	غ	0.79	0.79	0.84
5	ح	0.85	0.78	0.81	79	ف	74	0.83	0.78
6	خ	0.88	0.84	0.86	20	ق	0.88	0.86	0.87
7	د	0.82	0.74	0.78	21	ك	0.87	0.90	0.89
8	ذ	0.74	0.74	0.74	22	ل	0.91	0.94	0.93
9	ر	0.83	0.93	0.88	23	م	0.89	0.90	0.89
10	ز	0.87	0.88	0.88	24	ن	0.84	0.82	0.83
11	س	0.93	0.93	0.93	25	هـ	0.86	0.87	0.86
12	ش	0.90	0.93	0.92	26	و	0.90	0.89	0.89
13	ص	0.84	0.88	0.86	27	ي	0.95	0.92	0.93
Train_acc		**93 %**							
Test_acc		**88 %**							

The transfer learning approaches are applied to the same database, The database is used as a target for pretrain transfer learning models, while the ImageNet dataset is used as the source dataset to train these models randomly.

Two experiments have been done. First, evaluating CNN and its different architectures separately to recognize Arabic handwritten characters, and secondly, evaluating all models in the Arabic characters classification. Table 5.3 shows the Multi-class evaluation matrices' results of accuracy, Positive Predictive Value, Sensitivity, F1-score, Mathews Correlation Coefficient and Negative Predictive Value which the trained dataset obtained is used in transfer learning techniques. Moreover, the table 5.3 displays the runtime is taken from each of the Deep CNN architectures.

Table 5.3: Balanced Multi-class evaluation metrices for CNN and each model on the combined dataset.

Confusion Matrices / Models	SEN	SPE	MCC	F1_score	PPV	NPV	ACC	Runtime (Hours)	
								Train	Test
DenseNet151	100	99.52	96.44	96.43	93.75	100	98.42	13:34	1:49
InceptionV3	100	99.52	96.44	96.43	93.75	100	98.94	7:14	2:48
ResNet50	100	99.71	97.69	9673	95.83	100	97.90	0:54	0:15
VGG16	100	97.44	93.52	91.49	92.84	99	97.71	22.53	4.55
CNN		87.81			87.88		93	0:25	0:4

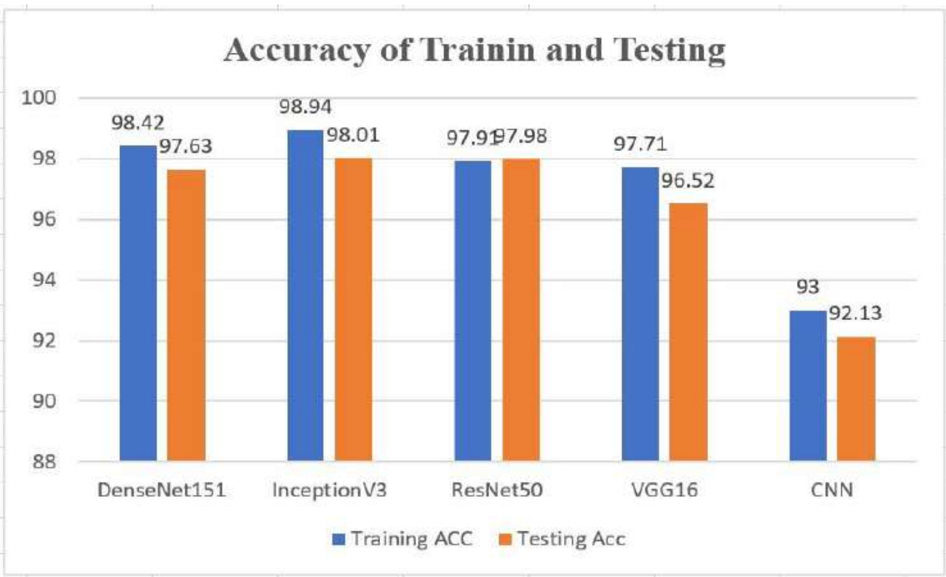


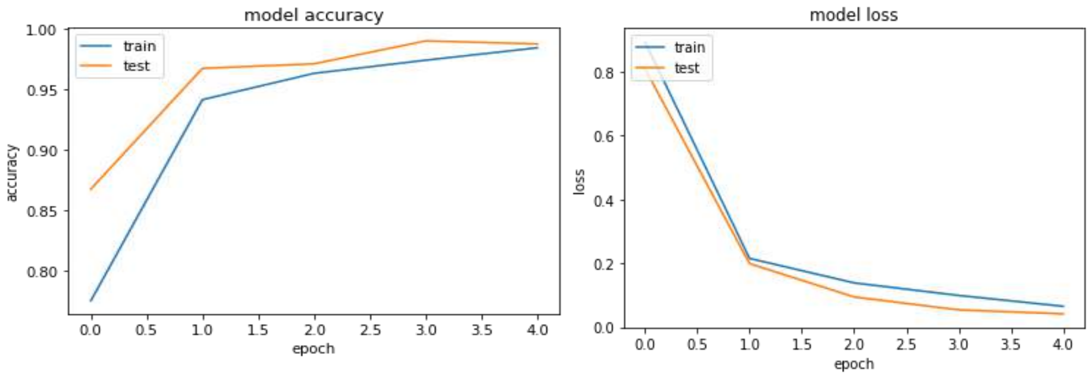
Figure 5.6: Accuracy of Training and Testing for Arabic characters.

Table 5.3 exhibits the performance matrix computed using Equations (5.4 – 5.10). Comparisons are described in terms of accuracy and the time spent in carrying out the process. The Inception V3 architecture outperformed the VGG, ResNet and CNN models on the combined dataset (AHCD and our dataset). Inception V3 based model achieved accuracy of 99%, precision as 93.75%, recall (SEN) 100 %, Specificity as 99.8%; the total runtime for training is almost 7 hours. While the VGG16 architecture achieved accuracy as 97.71%, 93.75%, sensitivity 100%, Specificity (SPE) 99.8%.

Comparatively higher training time of 23 hours. Therefore, in this architecture, the images of the combined dataset have been reduced from 131,000 images to 13346 images.

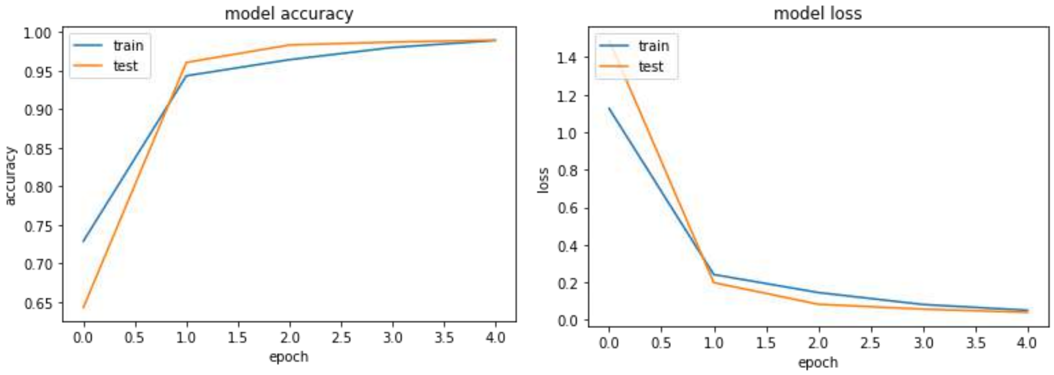
The reason for such reduction is that Google Collab (and Spyder.4[MSC V.1916 64 bits]) can't execute the classification or recognition process on 131,000 images using VGG16, as this VGG requests a high feature of computers. The proposed CNN is lightweight in nature when compared different models which are being used in the proposed work. The performance of CNN based model as accuracy of 93%, Positive Predictive Value 87.88%, Specificity 87.81%. CNN model's training time is nearly 25 minutes on the dataset A and dataset B. Figure 5.7 displays Arabic handwritten characters recognition accuracy for the proposed CNN and transfer learning approaches.

We used two methods in regularization to prevent overfitting. First, dropout, which in some of the output nodes of a layer are ignored randomly (This technique in which random discarding occurs in the output nodes of the layer). During the training, the connections to that node are such that the network behaves as if this node does not exist. When the weights are updated, the network appears as a new network because the discarded nodes are not the same on each iteration. We made the dropout layer as 0.50 for all the transfer learning approaches. Second, Batch normalization is the process of normalizing the inputs in each layer. It is used to reduce the internal covariance shift in hidden layers because, in a deep network, there may be an internal covariance in the hidden layers due to the various inputs of the network. A value of Batch was set to 64 for all transfer learning algorithms. The data augmentation method is used when the dataset is small. In this work, this technique increases the input data. It causes crowding of input data during the training process, making the machine learning process take a very, very long time up to days. The implementation stops training for devices with low key features, and the suspension problem occurs, especially while using VGG16. For that we do not use any data augmentation.



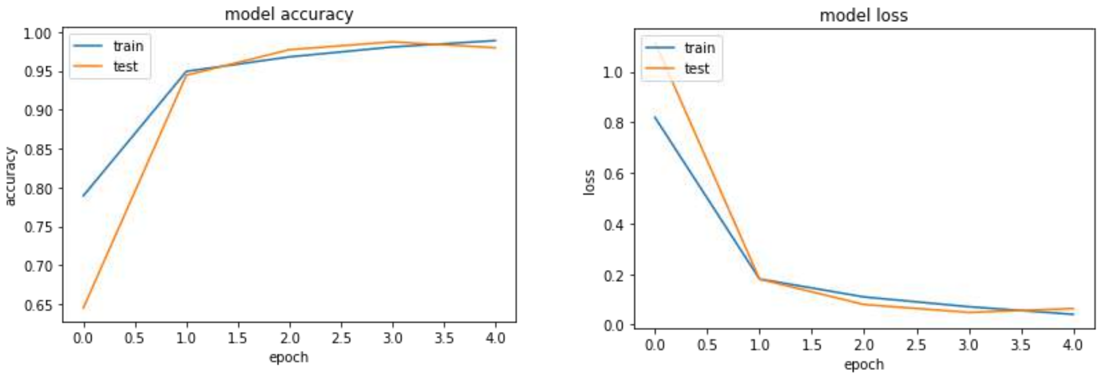
a b

Figure 5.7: (a) Accuracy and (b) loss of DenseNet 121.



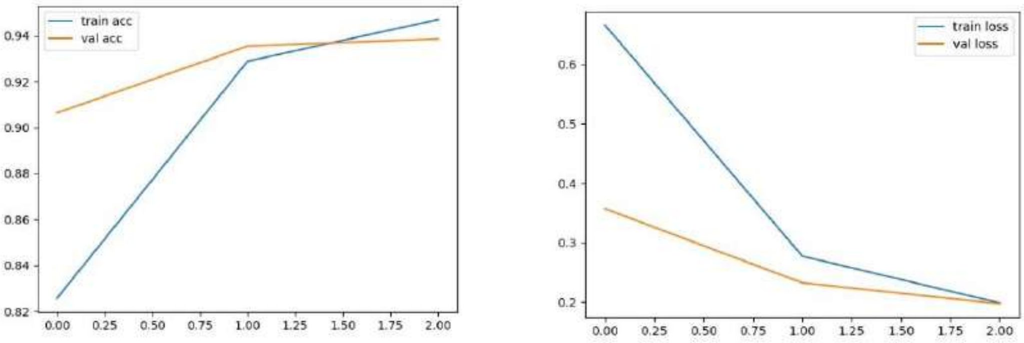
a b

Figure 5.8: (a) Accuracy and (b) loss of Inception V3.



a b

Figure 5.9: (a) Accuracy and (b) loss of ResNet 50.



a **b**

Figure 5.10: (a) Accuracy and (b) loss of VGG16.

Because of the time spent on training and testing processes, especially in VGG16, the epochs of training have been set be 5. The accuracy of training and testing was documented and loss in shapes from 5.6 to 5.10. These figures display good results. Figure 5.6 (a) showed that the test process was better than the training. Thus, the training result was acceptable. Likewise, (b) of figure 5.6 in the loss, as the model gave a loss in the test less than training in all training repetitions. Figure 5.7 The training provided a better accuracy during the training and test process, the two curved lines of train and test are slightly closed together, which means that this model is perfect to recognize Arabic handwritten characters. However, the dropout was 0.5 in all used models. If the epochs increase, the prediction of this model will be better. As shown in figure 5.7, ResNet-50, higher accuracy of train and test was shown as well as loss that is lower than VGG16 and CNN model. Thus, the two models, Inception V3 and ResNet-50, achieved better performance in updating the bias and weights to the recognition performance. The following tables report the evaluation matrices reports for each model used in classifying Arabic handwritten characters, providing the comparison as well.

Table 5.4: Evaluation Matrices for DenseNet 121 character classification.

character	SEN	SPE	ACC	MCC	F1-score	PPV	NPV
0. alif	100	98.09	93.19	0.97	95.71	96.75	100
1. baa	100	100	100	1	100	100	100
2. taa	100	100	100	1	100	100	100
3. thaa	100	100	100	1	100	100	100
4. jeem	100	100	98.55	1	100	100	100
5. hhaa	100	100	98.55	1	100	100	100
6. khaa	100	100	98.55	1	100	100	100
7. daal	100	100	98.55	1	100	100	100
8. thaal	100	100	99.88	1	98.67	100	99.64
9. raa	98.64	99.64	99.76	0.94	94.43	95.43	99.88
10. zay	98.64	99.64	99.4	0.94	98.43	98.43	99.28
11. seen	98.64	99.64	97.88	0.94	94.43	95.43	99.24
12. sheen	98.64	94.79	96.12	0.91	94.43	95.43	94.81
13. saad	99.64	99.62	98.36	0.94	94.43	95.43	99.51
14. zhaad	97.64	98.67	99.64	0.51	98.92	98.92	99.28
15. tta	98.64	99.64	99.76	0.94	98.83	98.92	99.27
16. dhaa	99.64	99.52	99.88	0.98	99.52	99.52	99.39
17. ain	98.84	99.39	99.88	0.93	99.62	99.52	99.4
18. gain	99.74	99.76	95.52	0.96	99.76	99.72	99.27
19. faa	99.03	99.03	99.88	0.51	99.03	99.52	99.28
20. qaaf	99.03	99.64	96.76	0.94	99.64	99.64	99.24
21. kaaf	93.45	94.67	98.92	0.91	94.63	93.63	94.4

22. laam	93.45	94.67	94.98	0.91	94.63	95.22	94.27
23. meem	93.45	94.28	98.29	0.91	93.43	91.66	94.27
24. noon	93.45	94.27	98.17	0.91	94.43	95.43	94.26
25. haa	93.45	94.27	96.17	0.94	94.43	95.43	94.01
26 waw	93.45	94.02	97.93	0.91	93.43	91.66	94.14
27. yaa	94.13	100	95.81	0.94	100	99.00	94.46

Table 5.5: Evaluation Matrices for Inception V3 character classification.

character	SEN	SPE	ACC	MCC	F1-score	PPV	NPV
0. alif	100	98.09	98.19	0.857	85.71	75	100
1. baa	100	100	100	1	100	100	100
2. taa	100	100	100	1	100	100	100
3. thaa	100	100	100	1	100	100	100
4. jeem	100	100	100	1	100	100	100
5. hhaa	100	100	100	1	100	100	100
6. khaa	100	100	100	1	100	100	100
7. daal	100	100	100	1	100	100	100
8. thaal	100	99.88	99.88	0.925	92.31	85.71	100
9. raa	66.67	100	99.76	0.815	80	100	99.76
10. zay	33.33	99.88	99.4	0.688	44.44	66.67	99.52
11. seen	100	99.88	99.88	0.925	92.31	85.71	100
12. sheen	100	100	100	1	100	100	100
13. saad	66.67	100	99.76	0.815	80	100	99.76
14. zhaad	50	100	99.64	0.758	66.67	100	99.64

15. tta	83.33	99.88	99.76	0.832	83.33	83.33	99.88
16. dhaa	80	100	99.88	0.899	88.89	100	99.88
17. ain	80	100	99.88	0.899	88.89	100	99.88
18. gain	50	99.88	99.52	0.611	60	75	99.64
19. faa	83.33	100	99.88	0.913	90.91	100	99.88
20. qaaf	66.67	100	99.76	0.855	80	100	99.76
21. kaaf	100	100	100	1	100	100	100
22. laam	100	100	100	1	100	100	100
23. meem	100	100	100	1	100	100	100
24. noon	100	100	100	1	100	100	100
25. haa	100	100	100	1	100	100	100
26 waw	100	100	100	1	100	100	100
27. yaa	100	100	100	1	100	100	100

Table 5.6: Evaluation Matrices for ResNet-50 character classification.

character	SEN	SPE	ACC	MCC	F1-score	PPV	NPV
0. alif	100	98.85	98.92	0.976	90.91	83.33	100
1. baa	100	100	100	1	100	100	100
2. taa	100	100	100	1	100.	100	100
3. thaa	100	100	100	1	100	100	100
4. jeem	98.88	100	99.88	0.997	99.44	100	99.87
5. hhaa	100	100	100	1	100	100	100
6. khaa	100	100	100	1	100	100	100

7. daal	100	100	100	1	100	100	100
8. thaal	50	100	99.64	0.758	66.67	100	99.64
9. raa	83.33	99.76	99.64	0.769	76.92	71.43	99.88
10. zay	66.67	99.76	99.52	0.664	66.67	66.67	99.76
11. seen	33.33	99.88	99.4	0.488	44.44	66.67	99.52
12. sheen	100	100	100	1	100	100.	100
13. saad	83.33	100	99.88	0.912	90.91	100	99.88
14. zhaad	66.67	99.88	99.64	0.785	72.73	80	99.76
15. tta	50	100	99.64	0.758	66.67	100	99.64
16. dhaa	100	100	100	1	100	100	100
17. ain	100	99.52	99.52	0.745	71.43	55.56	100
18. gain	50	100	99.64	0.758	66.67	100	99.64
19. faa	83.33	99.88	99.76	0.831	83.33	83.33	99.88
20. qaaf	83.33	99.76	99.64	0.797	76.92	71.43	99.88
21. kaaf	100	100	100	1	100	100	100
22. laam	100	100	100	1	100	100	100
23. meem	100	100	100	1	100	100	100
24. noon	100	100	100	1	100	100	100
25. haa	100	100	100	1	100	100	100
26 waw	100	100	100	1	100	100	100
27. yaa	100	100	100	1	100	100.	100

Table 5.7: Evaluation Matrices for VGG16 character classification.

Character	**SEN**	**SPE**	**ACC**	**MCC**	**F1-score**	**PPV**	**NPV**
0. alif	100	98.85	94.19	0.976	90.91	83.33	100
1. baa	100	100	100	1	100	100	00.
2. taa	100	100	100	1	100.	100	100
3. thaa	100	100	100	1	100	100	100
4. jeem	98.88	100	95.65	0.997	99.44	100	99.87
5. hhaa	100	100	95.65	1	100	100	100
6. khaa	100	100	95.65	1	100	100	100
7. daal	100	100	98.55	1	100	100	100
8. thaal	50	100	97.88	0.758	66.67	100	99.64
9. raa	83.33	99.76	99.76	0.767	76.92	71.43	99.88
10. zay	66.67	99.76	99.4	0.664	66.67	66.67	99.76
11. seen	33.33	99.88	97.88	0.568	44.44	66.67	99.52
12. sheen	100	100	96.12	1	100	100.	100
13. saad	83.33	100	98.36	0.913	90.91	100	99.88
14. zhaad	66.67	99.88	99.64	0.785	72.73	80	99.76
15. tta	50	100	99.76	0.758	66.67	100	99.64
16. dhaa	100	100	99.88	1	100	100	100
17. ain	100	99.52	99.88	0.745	71.43	55.56	100
18. gain	50	100	95.52	0.754	66.67	100	99.64
19. faa	83.33	99.88	99.88	0.832	83.33	83.33	99.88

20. qaaf	83.33	99.76	95.76	0.767	76.92	71.43	99.88
21. kaaf	100	100	98.92	1	100	100	100
22. laam	100	100	94.98	1	100	100	100
23. meem	100	100	97.39	1	100	100	100
24. noon	100	100	98.17	1	100	100	100
25. haa	100	100	96.17	1	100	100	100
26 waw	100	100	97.93	1	100	100	100
27. yaa	100	100	95.81	1	100	100.	100

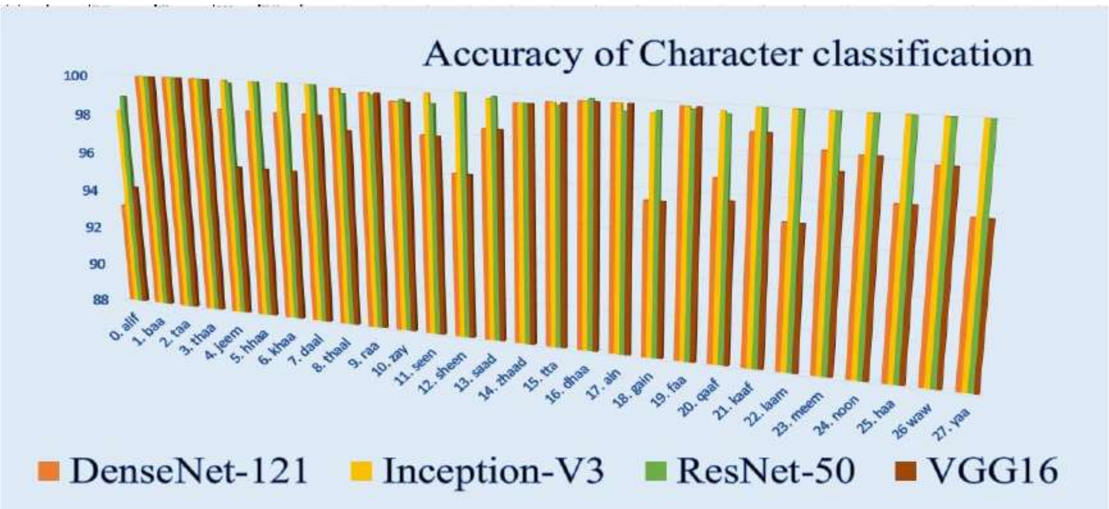


Figure 5.11: Arabic Character classification.

By the way, Matthews Correlation Coefficient (MCC) is a statistical method used to evaluate a model, and it measures the difference between the predicted value and the actual value. Using equation (6.9) to obtain the MCC value. The MCC value is usually between -1 to 1. The closer the outcome value to 1 means that the classification performance is excellent (Chicco et al., 2021). From the experiment results, we noticed that the proposed CNN model for recognizing isolated Arabic handwritten characters is an approach that can be promising and model methods to apply in the application of Arabic text recognition. This approach achieved high performance compared with traditional methods for Arabic text recognition. The most significant advantage of this approach is that it can classify and identify a large dataset. Accuracy (91-95) in identifying more than 131,000 images of Arabic letters. The misclassification occurred was due to the original image's quality, as explained earlier in the third chapter. The tables 5.4 – 5.7 display the evaluation matrices measures of each mode's performance for Arabic handwritten characters' classification.

As noticed in figure 5.8, Inception V3 provides better classification. There are similarities in shapes between the characters such as 'غ، ع' ,'كـ، ل' ,'ص، ض', 'بـ، يـ، ي', 'ق، ف، و', 'ش، س' in handwriting. So DenseNet 121 and VGG16 faced challenge to classify these characters. Based on the results explained above, it can be seen that the CNN method and deep CNN approaches are perfect models that can be used in other OCR systems for Arabic text recognition.

5.5.3 Comparison

Table 5.8 exhibits the comparisons between previous related studies that used the AHCD database with the proposed models.

Table 5.8. Comparison of performance between our approach with the related works.

Reference	Year	Method	Size of images	Accuracy
(El-Sawy et al., 2017)	2017	CNN	16800	94.9%
(Younis, 2017)	2018	Deep CNN	16800	97.6%
(AlJarrah et al., 2021)	2021	CNN with data augmentation	16800	97.6%

Our approach (Inception V3 and ResNet 50)	**2022**	**Transfer Learning approaches**	**16800 + 114200**	**98.94%**

5.6 Conclusion

Classifying Arabic handwritten characters is a challenging task. In this paper, the classification of Arabic handwritten character images was analyzed using four deep learning approaches, which are DenseNet 121, ResNet-50, Inception V3, and VGG16 approaches of deep transfer learning. Furthermore, the proposed CNN model evaluates and verifies the ability of each model to distinguish and recognize Arabic characters. The proposed approaches were applied to 131,000 images of Arabic handwritten characters. The dataset was split into 80% training, 10% tests and 10% verification. The results of tables 5.2, 5.4- 5.7 and figures (5.7- 5.10) are clarified that the Inception V3 and ResNet-50 models are the most suitable models for achieving high accuracy of almost 99% in training and 98% testing. Moreover, the perfect model for classifying Arabic characters is the DenseNet 121 model, which achieved a satisfactory accuracy rate in recognizing Arabic characters. The model VGG16 is a robust and more reliable model for classifying Arabic characters. Still, it needs high computer specifications and a very long task process, up to days depending on the dataset's size.

Chapter 6

Epilogue

6.1 Preamble

Arabic language occupies a global position. It is considered the fifth most common language in the world. Because Arabic language is strongly related to Islamic religion, it is one of the most widely spoken languages in the world.

This thesis presents an analysis of Arabic handwritten text from its inception until recent decades. In recent decades, the interest of researchers is in the computer vision field to transform Arabic text into editable digit text, printed or handwritten, comes in the priority.

A summary of the stages was presented to develop further attempts to transform Arabic text into digital. Furthermore, the problems and challenges that faced researchers in their research works are also reviewed in this study. When it comes to converting Arabic text into digital text, we have developed various appropriate techniques to improve and increase of the accuracy rate of the Arabic handwritten text recognition system. This study focuses on Arabic handwritten text recognition because of the challenges and problems which reduce the efficiency of recognition systems and make Arabic handwritten text recognition processes more complicated than printed recognition. We have enhanced many methods used in the pre-processing stage. And then proposed a robust approach to segmenting Arabic words and the segmentation of the Arabic characters. Then optimized techniques are suggested in machine learning that is used to classify isolated Arabic characters and then identify them. The evaluation measurements of classification recognition accuracy rates of each proposed model were shown to choose the optimization models that are used in Arabic handwritten text recognition processes. This chapter will present a summary of the study in addition to listing the significant contributions made during this research work as well as discussing the direction of additional research based on the work.

6.2 Summary

In this thesis, we divided this research work concerning recognizing Arabic handwritten characters into four stages. In the first stage, we initially propose a model for enhancing poor images of Arabic textual documents by using sequential methods and the correction of multiple skews in many images to increase the quality.

The second stage deals with the segmentation of the document into lines. Many images have wavy or overlapping lines. Therefore, the lines cannot be extracted individually. Then the word is segmented directly from the input image and then the word is segmented into characters. Morphological operators and Bounding boxes are used to extract the region of a line or word to estimate every character's region in the word. The computation analysis is used to extract the features for segmenting this isolated character. The last stage classifies the segmented Arabic characters and then their identification. The classification process is done by proposing a CNN model. Furthermore, transfer learning proposed approaches already trained with the ImageNet dataset pre-train on our dataset. Using transfer learning approaches to choose the appropriate model for recognizing Arabic handwritten characters to convert Arabic scripts into digital format.

Moreover, the proposed techniques were verified using various available databases such as IFN/ENIT, KHAAT and AHCD databases. In addition, we created a new database for this research.

Below is a description outlines a brief explanation of each chapter of this research:

Chapter 1: It gives a general introduction to Arabic writing and the stages of its development over the ages, then the transition to digital writing in recent decades, and also, a literature review of the Arabic OCR system. In addition, it presents a review of the various methods proposed in machine learning for Arabic text recognition.

Chapter 2: Reviews the various available Arab databases and a new database for evaluating this research work was created.

Chapter 3: Suggests methods and filters were used in the pre-processing stage to improve the quality of poor images.

Quantitative measures for different enhancement methods such as peak signal-to-noise ratio (PSNR), Edge-based contrast measure (EBCM), Root Mean Square Error (RMSE), Image Quality Index, and Mean Absolute Error (MAE), and Mean Square Error (MSE). These measurements were used for the efficiency of the proposed methods and to measure the percentage of errors. Many Arabic scripts contain unwanted shapes or signatures. To eliminate these unwanted things, we have suggested a technique to remove this unwanted information.

Chapter 4: Clarifies segmentation of Arabic handwritten words into independent characters. This stage is one of the most challenging and complex tasks because Arabic writing is written in curved and connected letters. Therefore, we proposed a flexible and hybrid method that mainly depends on connected objects, boundary boxes, computational analysis of the word area and estimation of the area of each word to extract the target characters.

Chapter 5: In this chapter several machine learning and deep learning models were proposed for classifying isolated Arabic characters to recognize them. The proposed models were trained and tested on a large dataset that includes images of isolated Arabic handwritten characters. Evaluation matrices and the confusion matrices were utilized to know the effectiveness of the presented models. The experiment results were presented and shown in tables. Through the results of the experiments, it is noticed that our method works better and optimally and that CNN models and learning transfer approaches are the best methods for recognizing machine-printed or handwriting Arabic texts, as they can process with extensive data.

6.3 Contributions

1. Creating a new database of Arabic handwritten scripts
2. Developing efficient methods for enhancing Arabic document images using image processing techniques.
3. Constructing a robust algorithm for the segmentation of Arabic handwritten characters.
4. Proposing efficient models for classifying and recognizing Arabic handwritten characters using machine learning and deep learning approaches.

6.4 Scope for Future Work

Future research works include the following:

1. An investigation of alternative pre-processing approaches for producing clean and simple Arabic text pictures for processing.

2. Due to the cursive nature of the Arabic language, considerable study into segmenting handwritten Arabic text should be conducted in order to achieve the highest segmentation rate.
3. Segmenting Arabic ligatures is complex in both machine-printed and handwritten of Arabic language. As a result, more research into segmenting Arabic ligatures is advised.
4. Because Arabic words are so similar, post-processing procedures should be thoroughly researched. Post-processing will undoubtedly boost the recognition rate.
5. Attempt to enhance other pre-processing techniques to supply clean and easy Arabic text images for processing.

Author's Publications

➢ Journals:

1. Boraik, Omar Ali, Ravikumar M. and Mufeed Ahmed Naji Saif. , 2022, "Characters Segmentation from Arabic Handwritten Document Images: Hybrid Approach.", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 13, No. 4, Pp.395-403, [SCOPUS].

2. Boraik, Omar Ali and Ravikumar M., 2022, "Arabic Handwritten Character Classification and Recognition Using CNN and Transfer Learning Approach", The Seybold Report Journal (TSRJ), Vol.17, No.11, [SCOPUS].

3. Boraik, Omar Ali and M. Ravikumar, 2022. "Recognition Of Arabic Handwritten Characters: A Review", Journal of Object Technology [SCOPUS] [Communicated].

➢ Conferences:

4. Ravikumar, M., and Boraik, O. A. 2021, "Estimation and Correction of Multiple Skews Arabic Handwritten Document Images," In International Conference on Innovative Computing and Communications, pp. 553-564, (Springer).

5. Ravikumar, M. and Boraik, O. A., 2020, "Low Pass Filter-Based Enhancement of Arabic Handwritten Document Images," In International Conference on Information and Communication Technology for Intelligent Systems. pp. 271-277. (Springer).

Bibliography

Abdalkafor, A. S., Awad, W. K., and Alheeti, K. M. A. (2020). A Novel Comprehensive Database for Arabic And English Off-Line Handwritten Digits Recognition. Indonesian Journal of Electrical Engineering and Computer Science, 20(1), pages 145-149.

Abdalkafor, A.S. (2018). Survey for Databases on Arabic Off-line Handwritten Characters Recognition System. 1st International Conference on Computer Applications and Information Security, pages 1-6.

Abdullah, M. A., Al-Harigy, L. M., and Al-Fraidi, H. H. (2012). Off-Line Arabic Handwriting Character Recognition Using Word Segmentation, pages 40-44.

Abuhaiba, I. S., Mahmoud, S. A., and Green, R. J. (1994). Recognition Of Handwritten Cursive Arabic Characters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(6), pages 664-672.

Aburas, A.A., and Gumah, M.E. (2008). Arabic handwriting recognition: Challenges and solutions. International Symposium on Information Technology, 2, pages 1-6.

Adel Al-Aloosi., (2008). Arabic Script: Evolution and Improvements. Arabic house for books, Cairo, Egypt, 411/6.

Ahmad, I. (2013). A Technique for Skew Detection of Printed Arabic Documents. 10th International Conference Computer Graphics, Imaging and Visualization, pages 62-67.

Ahmad, I., and Fink, G.A. (2016). Class-Based Contextual Modeling for Handwritten Arabic Text Recognition, 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 554-559.

Ahmed Abdelmajid Khalifa., (2015). Al-Khalil ibn Ahmad al-Farahidi- the legend. His poetry and efforts to serve Arabic, International Islamic University in Malaysia and the Omani Students Association at the university.

Ahmed, R. I., and Mohamed EM Musa. (2016). Preprocessing Phase for Offline Arabic Handwritten Character Recognition. International Journal of Computer Applications Technology and Research, 5(12), Pages 760-763.

Ahmed, Rawia IO, and Mohamed EM Musa., (2016). Pre-processing Phase for Offline Arabic Handwritten Character Recognition. Int. J. Compute. Appl. Technol. Res 5.12, pages 760-763.

Al Hamad, H. A. (2013). Use an efficient neural network to improve Arabic handwriting recognition. International Conference on Signal and Image Processing Applications, pages 269-274.

Al Maadeed, S., Ayouby, W., Hassaine, A., and Aljaam, J. M. (2012). QUWI: an Arabic and English handwriting Dataset for Offline Writer Identification. International Conference on Frontiers in Handwriting Recognition, pages 746-751.

Alaei, A., Pal, U., and Nagabhushan, P. (2011). A New Scheme for Unconstrained Handwritten Text-Line Segmentation. Pattern Recognition, 44(4), pages 917-928.

AlAgha, I., and Abu-Samra, Y. (2020). Tag Recommendation for Short Arabic Text by Using Latent Semantic Analysis of Wikipedia. Jordanian Journal of Computers and Information Technology (JJCIT), 6(02), pages 165-181.

Alamri, H., Sadri, J., Suen, C. Y., and Nobile, N. (2008). A Novel Comprehensive Database for Arabic Off-Line Handwriting Recognition. Proceedings of 11th international conference on frontiers in handwriting recognition, ICFHR. pages 01-07.

Al-Antari, M. A., Al-Masni, M. A., Choi, M.-T., Han, S.-M., and Kim, T.-S. (2018). A Fully Integrated Computer-Aided Diagnosis System for Digital X-Ray Mammograms Via Deep Learning Detection, Segmentation, And Classification. International journal of medical informatics, 117, pages 44-54.

Al-Antari, M. A., Hua, C.-H., Bang, J., and Lee, S. (2021). Fast Deep Learning Computer-Aided Diagnosis Of COVID-19 Based on Digital Chest X-Ray Images. Applied Intelligence, 51(5), pages 2890-2907.

Alghamdi, M., and Teahan, W. (2018). Printed Arabic script recognition: A Survey. International Journal of Advanced Computer Science and Applications, 9(9), pages 415-428.

Alginahi, Y. (2010). Preprocessing techniques in character recognition. Character recognition, 1, pages 01-19.

Al-Hajj, R., Mokbel, C., and Likforman-Sulem, L. (2007). Combination of HMM-Based Classifiers for the Recognition of Arabic Handwritten Words. Ninth International Conference on Document Analysis and Recognition, pages 959-963.

Al-Huri., Ibrahim. (2015). Arabic Language: Historic and Sociolinguistic Characteristics. English Literature and Language Review 1.4, pages 28-36.

Ali, M., and Alasadi, H. (2015). High Accuracy Arabic Handwritten Characters Recognition Using Error Back Propagation Artificial Neural Networks. (IJACSA) International Journal of Advanced Computer Science and Applications, 6(2), pages 145-152.

Ali, O. B., and Shaout, A. (2014). Isolated Arabic Handwritten Character Recognition: A Survey. International Journal, 4(10).

AlJarrah, M.N., Zyout, M.M., and Duwairi, R. (2021). Arabic Handwritten Characters Recognition Using Convolutional Neural Network. 12th International Conference on Information and Communication Systems (ICICS), pages 182-188.

Al-Khatatneh, A., Pitchay, S.A., and Al-qudah, M. (2015). A Review of Skew Detection Techniques for Document. 2015 17th UKSim-AMSS International Conference on Modelling and Simulation (UKSim), pages 316-321.

Al-Masni, M. A., Al-Antari, M. A., Choi, M.-T., Han, S.-M., and Kim, T.-S. (2018). Skin Lesion Segmentation in Dermoscopy Images Via Deep Full Resolution Convolutional Networks. Computer methods and programs in biomedicine, 162, pages 221-231.

Al-Masni, M. A., Al-Antari, M. A., Park, J.-M., Gi, G., Kim, T.-Y., Rivera, P., Valarezo, E., Choi, M.-T., Han, S.-M., and Kim, T.-S. (2018). Simultaneous Detection and Classification Of Breast Masses In Digital Mammograms Via A Deep Learning YOLO-Based CAD System. Computer methods and programs in biomedicine, 157, pages 85-94.

Almisreb, A. A., Turaev, S., Saleh, M. A., and Al Junid, S. A. M. (2022). Arabic Handwriting Classification using Deep Transfer Learning Techniques. Pertanika Journal of Science and Technology, 30(1). pages 641 - 654.

Al-Muhtaseb, H. A., Mahmoud, S. A., and Qahwaji, R. S. (2008). Recognition Of Off-Line Printed Arabic Text Using Hidden Markov Models. Signal processing, 88(12), pages 2902-2912.

Al-Ohali, Y., Cheriet, M., and Suen, C. (2003). Databases For Recognition Of Handwritten Arabic Cheques. Pattern Recognition, No. 36(1), pages111-121.

Al-Shatnawi, A. M., and Omar, K. (2009). Skew Detection and Correction Technique For Arabic Document Images Based On Centre Of Gravity. Journal of Computer Science, 5(5), page 363.

Altwaijry, Najwa, and Isra Al-Turaiki.(2021) .Arabic Handwriting Recognition System Using Convolutional Neural Network, Neural Computing and Applications 33.7, pages 2249-2261.

Amara, N. E. B., Mazhoud, O., Bouzrara, N., and Ellouze, N. (2005). ARABASE: A Relational Database for Arabic OCR Systems. Int. Arab J. Inf. Technol., 2(4), pages 259-266.

Amrouch, M. and Rabi, M., (2017). The Relevant CNNs Features Based HMM for Arabic Handwriting Recognition. Int. J. Imaging Robot, 17, pages 98-109.

Aneja, N., and Aneja, S. (2019). Transfer learning using CNN for handwritten Devanagari character recognition. 1st International Conference on Advances in Information Technology (ICAIT), pages 293-296.

Aneja, S., Aneja, N., and Islam, M. S. (2018). IoT device fingerprint using deep learning. In IEEE international conference on internet of things and intelligence system (IOTAIS), pages 174-179.

Anwar, K., and Nugroho, H. (2015). A segmentation scheme of Arabic words with harakat. In 2015 IEEE International Conference on Communication, Networks and Satellite (COMNESTAT), pages 111-114.

Aouadi, N. and A. Kacem. (2017). A Proposal For Touching Component Segmentation In Arabic Manuscripts. Pattern Analysis and Applications, 2017. 20(4): Pages 1005-1027.

Aouadi, N., Amiri, S., and Echi, A. K. (2013). Segmentation Of Connected Components In Arabic Handwritten Documents. Procedia Technology, 10, pages 738-746.

Arivazhagan, M., Srinivasan, H., and Srihari, S. (2007). A statistical approach to line segmentation in handwritten documents. In Document recognition and retrieval XIV Vol. 6500, pages 245-255.

Arunachalam, T., (2017). An Efficient Color Image Segmentation using Edge Detection and Thresholding Methods. International Journal of Advanced Networking and Applications (IJANA) V: 08, Issue: 05, pages 19-23.

Assayony, Mohammed O., and Sabri A. Mahmoud., (2016). An Enhanced Bag-of-Fatures Framework for Arabic handwritten Sub-words and Digits Recognition. J. Pattern Recognition. Intell. Syst, Pages 27-38.

Athoillah, M. (2019). K-Nearest Neighbor for Recognize Handwritten Arabic Character. Jurnal Matematika MANTIK, 5(2), pages 83-89.

Awel, Muna Ahmed, and Ali Imam Abidi., (2019). Review On Optical Character Recognition. International Research Journal of Engineering and Technology (IRJET) 6.6. pages 3666-3669.

Awni, M., Khalil, M. I., and Abbas, H. M. (2021). Offline Arabic Handwritten Word Recognition: A Transfer Learning Approach. Journal of King Saud University-Computer and Information Sciences, No. 10, pages 9654-9661.

Aytar, Y., and Zisserman, A. (2011). Tabula Rasa: Model Transfer for Object Category Detection. International Conference on Computer Vision, pages 2252-2259.

Batawi, Y., and Abulnaja, O. (2012). Accuracy Evaluation of Arabic Optical Character Recognition Voting Technique: An Experimental Study. IJECS: International Journal of Electrical and Computer Sciences, No.12(1), pages 29-33.

Belaïd, A. and N. Ouwayed, (2011). Segmentation of Ancient Arabic Documents, in" Guide to OCR for Arabic Scripts,". Eds. Volker Märgner and Haikal El Abed., Springer-Verlag, London. Pages 103-122.

Benjelil, M., Kanoun, S., Mullot, R., and Alimi, A. M. (2009, July). Arabic and Latin script identification in printed and handwritten types based on steerable pyramid features. 10th International Conference on Document Analysis and Recognition. pages 591-595.

Besbas, W. S., Sunni, M. R., and Elbokhare, A. F. Improved Method for Sliding Window Printed Arabic OCR. 4th Int'l Conference on Advances in Engineering Sciences and Applied Mathematics. Pages 35-40.

Biswas, D. (2011). Novel Gray Scale Conversion Techniques Based on Pixel Depth. Journal of Global Research in Computer Science, No. 2(6), pages 118-121.

Boukerma, H., and Farah, N. (2012, November). Preprocessing algorithms for Arabic handwriting recognition systems. In 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), pages 318-323.

Boukharouba, A. (2017). A New Algorithm for Skew Correction and Baseline Detection Based on The Randomized Hough Transform. Journal of King Saud University-Computer and Information Sciences, No. 29(1), pages 29-38.

Brodić, D., and Milivojević, Z. N. (2013). Text line segmentation with the algorithm based on the oriented anisotropic Gaussian kernel. Journal of Electrical Engineering, 64(4), pages 238-243.

Cavalin, P. R., de Souza Britto Jr, A., Bortolozzi, F., Sabourin, R., and Oliveira, L. E. S. (2006). An implicit segmentation-based method for recognition of handwritten strings of characters. In Proceedings of the 2006 ACM symposium on Applied computing. pages 836-840.

Charfi, M., Kherallah, M., Baati, A. E., and Alimi, A. M. (2012). A New Approach for Arabic Handwritten Postal Addresses Recognition. arXiv:1204.1678. pages 01-07.

Chaudhuri, Arindam, et al., (2017). Optical Character Recognition Systems. Optical Character Recognition Systems for Different Languages with Soft Computing. Springer, Cham, Pages 9-41.

Chen, J., Cao, H., Prasad, R., Bhardwaj, A. and Natarajan, P., (2010). Gabor Features for Offline Arabic Handwriting Recognition. In Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, pages 53-58.

Chergui L.; Maamar Kef., (2015). SIFT Descriptors for Arabic Handwriting Recognition. International Journal of Computational Vision and Robotics, pages 441-461.

Cheriet, M., Kharma, N., Suen, C., and Liu, C.-L. (2007). Character Recognition Systems: A Guide for Students and Practitioners. John Wiley and Sons.

Chicco, D., Tötsch, N., and Jurman, G. (2021). The Matthews Correlation Coefficient (MCC) Is More Reliable Than Balanced Accuracy, Bookmaker Informedness, And Markedness in Two-Class Confusion Matrix Evaluation. BioData mining, 14(1), pages 1-22.

Choudhary, A. (2014). A Review of Various Character Segmentation Techniques for Cursive Handwritten Words Recognition. Int J Inf Comput. Technol, 4(6), pages 559-564.

Cote, R. A. (2009). Choosing one dialect for the Arabic speaking world: A status planning dilemma. Journal of Second Language Acquisition and Teaching, 16, pages 75-97.

Dave, N. (2015). Segmentation Methods For Hand Written Character Recognition. International Journal of Signal Processing. Image processing and pattern recognition, 8(4), pages 155-164.

El Abed, H., and Margner, V. (2007). Comparison of different pre-processing and feature extraction methods for offline recognition of handwritten Arabic words. In Ninth International Conference on Document Analysis and Recognition (ICDAR), Vol. 2, pages 974-978.

El Moubtahij, H., Halli, A., and Satori, K. (2014). Review Of Feature Extraction Techniques for Offline Handwriting Arabic Text Recognition. International journal of advances in engineering and technology, 7(1), page 50.

El-Hajj, R., Likforman-Sulem, L., and Mokbel, C. (2005). Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling. Eighth International Conference on Document Analysis and Recognition (ICDAR), pages 893-897.

Elleuch, M., Tagougui, N., and Kherallah, M., (2015). Towards Unsupervised Learning for Arabic Handwritten Recognition Using Deep Architectures. International Conference on Neural Information Processing. Springer, Cham, pages 363-372.

El-Sawy, A., Loey, M., and El-Bakry, H. (2017). Arabic Handwritten Characters Recognition Using Convolutional Neural Network. WSEAS Transactions on Computer Research, No. 5, pages 11-19.

El-Sherif, E. A., and Abdelazeem, S. (2007, July). A Two-Stage System for Arabic Handwritten Digit Recognition Tested on a New Large Database. In Artificial intelligence and pattern recognition, pages 237-242.

Elzobi, Moftah, et al. Arabic Handwriting Recognition Using Gabor Wavelet Transform And SVM, IEEE 11th International Conference on Signal Processing, Vol. 3, pages 2154-1258.

Fadeel, M. A. (2016, August). An efficient segmentation algorithm for Arabic handwritten characters recognition system. In 2016 Third International Conference on Mathematics and Computers in Sciences and in Industry (MCSI), pages 172-177.

Fakhr, M. (2011). Arabic Optical Character Recognition (OCR) Systems Overview, pages 01-36.

Farooq, F., Govindaraju, V., and Perrone, M. (2005). Pre-Processing Methods for Handwritten Arabic Documents. Eighth International Conference on Document Analysis and Recognition (ICDAR'05),

Farulla, G.A., N. Murru, and R. Rossini., (2017). A fuzzy approach to segment-touching characters. Expert Systems with Applications. 88: pages 1-13.

Gonzalez, R. C. (2009). Digital Image Processing. Pearson Education India.

Goularas, D., and Kamis, S. (2019, August). Evaluation of deep learning techniques in sentiment analysis from Twitter data. International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), pages 12-17.

Guru, D.S., Ravikumar, M., and Manjunath, S. (2013). Multiple Skew Estimation in Multilingual Handwritten Documents. International Journal of Computer Science Issues (IJCSI), 10(5), page 65.

Haghighi, P. J., Nobile, N., He, C. L., and Suen, C. Y. (2009). A new large-scale multi-purpose handwritten Farsi database. In International Conference Image Analysis and Recognition, pages 278-286.

Hamid, A., and Haraty, R. (2001). A neuro-heuristic approach for segmenting handwritten Arabic text. In Proceedings ACS/IEEE international conference on computer systems and applications, pages 110-113.

Haraty, R. (2004). Arabic Text Recognition. The International Arab Journal of Information Technology, Vol.1, No. 2, pages 156-163.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770-778.

Helli, B., and Moghaddam, M.E. (2008). A Text-Independent Persian Writer Identification System Using LCS Based Classifier. 2008 IEEE International Symposium on Signal Processing and Information Technology, pages 203-206.

Huang, G., Liu, Z., and Weinberger, K.Q. (2016). Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2261-2269.

Humied, I. A. (2016). Segmentation Accuracy for Offline Arabic Handwritten Recognition Based On Bounding Box Algorithm. Communications on Applied Electronics, 5(9), pages 21-30.

Jaderberg, M., Simonyan, K., and Zisserman, A. (2015). Spatial Transformer Networks. Advances in neural information processing systems, 28.

Jiang, Z.-P., Liu, Y.-Y., Shao, Z.-E., and Huang, K.-W. (2021). An Improved VGG16 Model for Pneumonia Image Classification. Applied Sciences, 11(23), pages 01-19.

Jindal, P., and Jindal, B. (2015, December). Line and word segmentation of handwritten text documents written in Gurmukhi script using midpoint detection technique. In 2015 2nd International Conference on Recent Advances in Engineering and Computational Sciences (RAECS) pages 1-6. IEEE.

Jundale, T. A., and Hegadi, R. S. (2015). Skew Detection and Correction of Devanagari Script Using Hough Transform. Procedia Computer Science, 45, pages 305-311.

Kamel Suliman Al-Jabori., (2000). Encyclopedia of Arabic Fonts", Al-Helal Library, Beirut, Lebanon.

Kang, L., and Doermann, D. (2011). Template Based Segmentation of Touching Components in Handwritten Text Lines, 2011 International Conference on Document Analysis and Recognition, pages 569-573.

Kanoun, S., Moalla, I., Ennaji, A., and Alimi, A. M. (2000). Script identification for Arabic and latin printed and handwritten documents. In Proc. of DAS, pages 159-165.

KeshavarzBahaghighat, M., and Mohammadi, J. (2012). Novel Approach for Baseline Detection and Text Line Segmentation. International Journal of Computer Applications, 51, pages 9-16.

Kessentini, Y., Paquet, T., and Hamadou, A. B. (2010). Off-Line Handwritten Word Recognition Using Multi-Stream Hidden Markov Models. Pattern Recognition Letters, 31(1), pages 60-70.

Khan, M. A. (2022). Arabic Handwritten Alphabets, Words and Paragraphs Per User (AHAWP) Dataset. Data, in Brief, 41, 107947.

Khandelwal, A., Choudhury, P., Sarkar, R., Basu, S., Nasipuri, M., and Das, N. (2009). Text Line Segmentation for Unconstrained Handwritten Document Images Using Neighborhood Connected Component Analysis. In International Conference on Pattern Recognition and Machine Intelligence, Springer, pages 369-374.

Kharma, N., Ahmed, M., and Ward, R. (1999). A new comprehensive database of handwritten Arabic words, numbers, and signatures used for OCR testing, Engineering Solutions for the Next Millennium, IEEE Canadian Conference on Electrical and Computer Engineering, pages 766-768.

Khosravi, H., and Kabir, E. (2007). Introducing a very large dataset of handwritten Farsi digits and a study on their varieties. Pattern Recognition Letters, 28(10), pages 1133-1141.

Lamghari, N., M. E. H. Charaf, and S. Raghay. (2018). Hybrid feature vector for the recognition of Arabic handwritten characters using feed-forward neural network." Arabian Journal for Science and Engineering, pages 7031-7039.

Lawgali, A. (2015). A Survey on Arabic Character Recognition. International Journal of Signal Processing, Image Processing and Pattern Recognition, 8, pages 401-426.

Lawgali, A., Bouridane, A., Angelova, M., and Ghassemlooy, Z. (2011). Automatic segmentation for Arabic characters in handwriting documents. 18th IEEE International Conference on Image Processing, pages 3529-3532.

Likforman-Sulem, L., Darbon, J., and Smith, E.H. (2009). Pre-Processing of Degraded Printed Documents by Non-local Means and Total Variation. 10th International Conference on Document Analysis and Recognition pages 758-762.

Liu, W., Sun, C.-J., Lin, L., and Liu, B. (2017). ITNLP-AiKF at SemEval-Task 1: Rich features based svr for semantic textual similarity computing. Proceedings of the 11th International Workshop on Semantic Evaluation pages 159-163.

Louloudis, G., Gatos, B., Pratikakis, I., and Halatsis, C. (2009). Text line and word segmentation of handwritten documents. Pattern recognition, 42(12), pages 3169-3183.

Mahmood Abbas, (2000). Arabic Handwriting Developments. Nahadhat Al-Ahariq House, Cairo University, Egypt.

Mahmoud, A., and Zrigui, M. (2017). Semantic Similarity Analysis for Paraphrase Identification In Arabic Texts. Proceedings of the 31st Pacific Asia conference on language, information and computation, pages 274-281.

Mahmoud, S. A., Ahmad, I., Al-Khatib, W. G., Alshayeb, M., Parvez, M. T., Märgner, V., and Fink, G. A. (2014). KHATT: An open Arabic offline handwritten text database. Pattern Recognition, 47(3), pages 1096-1112.

Mahmoud, S.A., Ahmad, I., Alshayeb, M.R., Al-Khatib, W.G., Parvez, M.T., Fink, G.A., Märgner, V., and Abed, H.E. (2012). KHATT: Arabic Offline Handwritten Text Database. 2012 International Conference on Frontiers in Handwriting Recognition, pages 449-454.

Makkar, N., and Singh, S. (2012). A brief tour to various skew detection and correction techniques. International journal for science and emerging technologies with latest trends, 4(1), pages 54-58.

Maliki, M., Jassim, S., Al-Jawad, N., and Sellahewa, H. (2012). Arabic handwritten: pre-processing and segmentation. In Mobile Multimedia/Image Processing, Security, and Applications, Vol. 8406, pages 100-107.

Manaf, M. (1998). Development of a system to recognize handwritten Jawi scripts and conversion of Jawi scripts to Rumi scripts. 6th International Conference and Exhibition on Multi- Lingual Computing, Kajang, Malaysia.

Märgner, V., El Abed, H., and Pechwitz, M. (2006). Offline handwritten Arabic word recognition using hmm-a character-based approach without explicit segmentation, pages 259-264.

Memon, J., Sami, M., Khan, R. A., and Uddin, M. (2020). Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR). IEEE Access, 8, pages 142642-142668.

Mezghani, A., Slimane, F., Kanoun, S., and Märgner, V. (2014). Identification of Arabic/French Handwritten/Printed Words using GMM-Based System. In CORIA-CIFED, pages 371-374.

Mohammed, S. W., and Soora, N. R. (2018). Global skew detection and correction using morphological and statistical methods. In Computational Vision and Bioinspired Computing, pages 556-568.

Mohebbi, M., and Talebpour, A. (2016). Texts semantic similarity detection-based graph approach. Int. Arab J. Inf. Technol., 13(2), pages 246-251.

Mousa, M. A., Sayed, M. S., and Abdalla, M. I. (2017). Arabic Character Segmentation Using Projection Based Approach with Profile's Amplitude Filter, pages 122-126.

Muaad, A.Y., Davanagere, P.H., Al-antari, M.A., and Lee, S. (2021). ArCAR: A Novel Deep Learning Computer-Aided Recognition for Character-Level Arabic Text Representation and Recognition. Algorithms, pages 01-24.

Naeem-Ullah, U., Ansari, M. A., Iqbal, N., and Saeed, S. (2019). First authentic report of Spodoptera frugiperda (JE Smith) (Noctuidae: Lepidoptera) an alien invasive species from Pakistan. Applied Sciences and Business Economics, 6(1), pages 1-3.

Nashwan, F., Rashwan, M. A., Al-Barhamtoshy, H. M., Abdou, S. M., and Moussa, A. M. (2018). A holistic technique for an Arabic OCR system. Journal of Imaging, pages 01-11.

Naz, S., Razzak, M. I., Hayat, K., Anwar, M. W., and Khan, S. Z. (2014). Challenges in baseline detection of Arabic script-based languages. Intelligent Systems for Science and Information, pages 181-196.

Naz, S., Umar, A. I., Ahmed, R., Razzak, M. I., Rashid, S. F., and Shafait, F. (2016). Urdu Nasta'liq text recognition using implicit segmentation based on multi-dimensional long short term memory neural networks. Springer Plus, 5(1), pages 1-16.

Neche, C., Belaid, A., and Kacem-Echi, A. 2019. Arabic handwritten documents segmentation into text-lines and words using deep learning. International Conference on Document Analysis and Recognition Workshops (ICDARW). Vol. 6, pages 19-24. IEEE.

Noubigh, Z., Mezghani, A., and Kherallah, M. (2020). Transfer Learning to improve Arabic handwriting text Recognition. 2020 21st International Arab Conference on Information Technology (ACIT),

Padmavathy, V., and Priya, R. (2018). Image contrast enhancement techniques-a survey. International Journal of Engineering and Technology, 7(2.33), pages 466-469.

Pak, I., and Teh, P. L. (2018). Text segmentation techniques: a critical review. Innovative Computing, Optimization and Its Applications pages 167-181.

Parhami, B., and Taraghi, M. (1981). Automatic recognition of printed Farsi texts. Pattern Recognition, 14(1-6), pages 395-403.

Patel, S. R., and Jha, J. (2015). Notice of Removal: Handwritten character recognition using machine learning approach-A survey. International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO).

Pathan, I. K., Ali, A. A., and Ramteke, R. (2012). Recognition of offline handwritten isolated Urdu characters. Advances in Computational Research, 4(1), pages 117-121.

Pechwitz, M., Maddouri, S. S., Märgner, V., Ellouze, N., and Amiri, H. (2002, October). IFN/ENIT-database of handwritten Arabic words. In Proc. of CIFED, Vol. 2, pages 127-136

Popa, R. (2014). Melanocytic lesions screening through particle swarm optimization. In Handbook of Research on Novel Soft Computing Intelligent Algorithms: Theory and Practical Applications, IGI Global, pages 355-384.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., and Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys (CSUR), 51(5), pages 1-36.

Qaroush, A., Awad, A., Modallal, M., and Ziq, M. (2020). Segmentation-based, omnifont printed Arabic character recognition without font identification. J. King Saud Univ. Comput. Inf. Sci., 34, pages 3025-3039.

Qaroush, A., Jaber, B., Mohammad, K., Washaha, M., Maali, E., and Nayef, N. (2022). An efficient, font independent word and character segmentation algorithm for printed Arabic text. Journal of King Saud University-Computer and Information Sciences, 34(1), pages 1330-1344.

Radaideh, A. A., and Rahim, M. S. M. (2016). Existing techniques in Arabic characters recognition (ACR). Journal of Informatics and Mathematical Sciences, 8(5), pages 347-360.

Ramanan, M. (2019). A hybrid approach for skew detection and correction in the multi-script scanned document. Asian J. Res. Comput. Sci, 4(2), pages 1-8.

Ramdan, J., Omar, K., Faidzul, M., and Mady, A. (2013). Arabic handwriting data base for text recognition. Procedia Technology, 11, pages 580-584.

Rashad, M., and Semary, N. A. (2014). Isolated printed Arabic character recognition using KNN and random forest tree classifiers. International Conference on Advanced Machine Learning Technologies and Applications, pages 11–17.

Ravi, S., and Khan, A. (2013). Morphological operations for image processing: understanding and its applications. Proc. 2nd National Conference on VLSI, Signal Processing and Communications NCVSComs, pages 17-19.

Ravikumar, M., Shivaprasad, B., Shivakumar, G., and Rachana, P. (2019). Estimation of Skew Angle from Trilingual Handwritten Documents at Word Level: An Approach Based on Region Props. In Soft Computing and Signal Processing pages 419-426. Springer.

Rehman, A., Mohamad, D., and Sulong, G. (2009). Implicit vs explicit based script segmentation and recognition: a performance comparison on benchmark database. Int. J. Open Problems Compt. Math, 2(3), 352-364.

Rezaei, S. B., Sarrafzadeh, H., and Shanbehzadeh, J. (2013). Skew detection of scanned document image, International MultiConference of Engineers and Computer Scientists, pages 01-07.

Roy, K., Das, S. K., and Obaidullah, S. M. (2011). Script identification from handwritten document. 2011 Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., and Bernstein, M. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), pages 211-252.

S Deshmukh, M., and R Kolhe, S. (2019). A hybrid character segmentation approach for cursive unconstrained handwritten historical Modi script documents. In Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India.

Saabni, R. (2014). Efficient recognition of machine-printed Arabic text using partial segmentation and Hausdorff distance. 2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR).

Sahlol, A. T., Elhoseny, M., Elhariri, E., and Hassanien, A. E. (2017). Arabic handwritten characters recognition system, towards improving its accuracy. In IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), pages 1-7.

Sahlol, A., and Suen, C. (2014). A novel method for the recognition of isolated handwritten Arabic characters. arXiv preprint arXiv:1402.6650. pages 01-13.

Saleem, S., Cao, H., Subramanian, K., Kamali, M., Prasad, R., and Natarajan, P. (2009). Improvements in BBN's HMM-Based Offline Arabic Handwriting Recognition System. 2009 10th International Conference on Document Analysis and Recognition, pages 773-777.

Saleh Ibrahim AL-Hassan., (2003). Arabic Script from Inscriptions to Text Documents. Al-Faisal Cultural House, KSA, 19/411, no.2925/22.

Sanchez, A., Suarez, P. D., Mello, C. A. B., Oliveira, A. L. I., and Alves, V. M. O. (2008, November). Text line segmentation in images of handwritten historical documents. In 2008 First Workshops on Image Processing Theory, Tools and Applications, pages 1-6.

Shaikh, N. A., Mallah, G. A., and Shaikh, Z. A. (2009). Character segmentation of Sindhi, an Arabic style scripting language, using height profile vector. Australian Journal of Basic and Applied Sciences, 3(4), pages 4160-4169.

Shams, M., Elsonbaty, A., and ElSawy, W. (2020). Arabic Handwritten Character Recognition based on Convolution Neural Networks and Support Vector Machine. arXiv preprint arXiv:2009.13450, pages 144-149.

Shamsan, E. A., Khalifa, O. O., Hassan, A., and Hamdan, H. M. (2017). Off-line Arabic handwritten character using neural network. IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), pages 1-5.

Sharma, D. V., and Lehal, G. S. (2009). A fast skew detection and correction algorithm for machine printed words in Gurmukhi script. In Proceedings of the International Workshop on Multilingual OCR, pages 1-5.

Soleymani, M., and Razzazi, F. (2003). An efficient front-end system for isolated Persian/Arabic character recognition of handwritten data-entry forms. International Journal of Computational Intelligence, 1, pages 193-196.

Solimanpour, F., Sadri, J., and Suen, C. Y. (2006). Standard databases for recognition of handwritten digits, numerical strings, legal amounts, letters and dates in Farsi language. Tenth International workshop on Frontiers in handwriting recognition. Pages 01-06.

Suen, C. Y., Nadal, C., Legault, R., Mai, T. A., and Lam, L. (1992). Computer recognition of unconstrained handwritten numerals. Proceedings of the IEEE, 80(7), pages 1162-1180.

Surinta, O., Holtkamp, M., Karaaba, M.F., Oosten, J.V., Schomaker, L., and Wiering, M.A. (2014). A Path Planning for Line Segmentation of Handwritten Documents. 2014 14th International Conference on Frontiers in Handwriting Recognition, pages 175-180.

Sushma, L., and Lakshmi, K. (2020). An analysis of convolution neural network for image classification using different models. Int J Eng Res Technol (IJERT), 9(10). Pages 629-637.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1-9.

Tharwat, A., Gaber, T., Hassanien, A. E., Shahin, M. K., and Refaat, B. Sift-based Arabic sign Language Recognition System., (2015) In Afro-european conference for industrial advancement; Springe; Berlin, Germany, pages 359-370.

Trier, O. D., and Jain, A. K. (1995). Goal-directed evaluation of binarization methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(12), pages 1191-1201.

Trier, Ø. D., Jain, A. K., and Taxt, T. (1996). Feature extraction methods for character recognition-a survey. Pattern Recognition, 29(4), pages 641-662.

Ullah, I., Azmi, M. S., Desa, M. I., and Alomari, Y. M. (2019). Segmentation of touching Arabic characters in Handwritten documents by overlapping set theory and contour tracing. Int. J. Adv. Comput. Sci. Appl, 10(5), pages 155-160.

Vishwanath, V.N., Manaswithareddy, G, Navya, K.D.P., and Ramya. M (2018). The Different Preprocessing Techniques Used in Handwritten Telugu Character Recognition. India. pages 17-21.

Wang, G. (2019). The status of Chinese handwriting identification and the improvement of methodologies. Forensic Sci, pages 4, 1-7.

Yamina, O.J., Mamoun, M.E., and Kaddour, S. (2017). Printed Arabic Optical Character Recognition Using Support Vector Machine. International Conference on Mathematics and Information Technology (ICMIT), pages 134-140.

Younis, K. S. (2017). Arabic Handwritten Character Recognition Based on Deep Convolutional Neural Networks. Jordanian Journal of Computers and Information Technology (JJCIT), 3(3), pages 186-200.

Yousif, I., and Shaout, A. (2014). Off-Line Handwriting Arabic Text Recognition: A Survey. International Journal of Advanced Research in Computer Science and Software Engineering, 4(9).

Zaiz, F., Babahenini, M. C., and Djeffal, A. (2016). Puzzle-Based System for Improving Arabic Handwriting Recognition. Engineering Applications of Artificial Intelligence, 56, pages 222-229.

Zeki, A. M. (2005, August). The Segmentation Problem In Arabic Character Recognition The State Of The Art. International Conference on Information and Communication Technologies, pages 11-26.

Zheng, L., Hassin, A. H., and Tang, X. (2004). A New Algorithm For Machine Printed Arabic Character Segmentation. Pattern Recognition Letters, 25(15), pages 1723-1729.

Zheng, L., Hassin, A. H., and Tang, X. (2004). A New Algorithm for Machine Printed Arabic Character Segmentation. Pattern Recognition Letters, 25(15), pages 1723-1729.

Ziaratban, M., Faez, K., and Bagheri, F. (2009). FHT: An Unconstraint Farsi Handwritten Text Database. 2009 10th International Conference on Document Analysis and Recognition, pages 281-285.