Republic Of Yemen
Ministry of Higher Education& Scientific Research
Taiz University
Al-Saeed Faculty Of Engineering & IT
Information Technology Department

الجمهورية اليمنية
وزارة التعليم العالي والبحث العلمي
جامعة تعز
كلية الهندسة وتقنية المعلومات
قسم تقنية معلومات

Arabic HandWriting Recognition System (AHWRS)

This Project Is For Getting A Bachelor's Degree

In Engineering Of Information Technology.

Done By

Ahmed Hasan Mohsen Alzubair

Eyad Ameen Taher Mansour

Ezadeen Mansour Almoshra

Amr Abdulkreem Abdulrahman

Nader Abdualqwi Qaid Ahmed

Supervised By

Dr : Mujeeb AL-Hakimy

TAIZ-YEMEN

2022-2023

ABSTRACT

Arabic handwriting recognition system (AHWRS) is a system that help converting image that contain handwriting/text to text suitable for computer processing ,so we can recognize the image of character/text and handwriting, but fewer works are available for the Arabic language due to the diversity in character shapes based on their positions in the words. Because Arabic handwriting is cursive, right to left in writing, and the letters convert their shapes and structures when placed at the beginning, middle, isolation, or end of words, it presents unique technical challenges. Arabic handwriting recognition is developed and designed in python programming language and Convolutional neural network (CNN), as ResNet, CTC, LSTM, Greedy decoding ,and using some image processing concepts such as filters and Segmentation.

the major objective of the project is to study the characteristics of the Arabic handwriting and building a system as accurate as passable with less errors ,because most of the font recognition systems are built for English, Latin ,Chinese characters ,so we have build a new model and we have achieved an accuracy of 95.3% .

Keywords:

Text Recognition, Text Detection , Deep learning, Patterns Recognition, AHRs, computer vision, CRAFT .

الخلاصة

نظام التعرف على الكتابة اليدوية العربية (AHWRS) هو نظام يساعد في تحويل الصورة التي تحتوي على الكتابة اليدوية او النص إلى نص مناسب لمعالجة الكمبيوتر ، حتى نتمكن من التعرف على الكتابة اليدوية او النصوص الموجودة داخل الصورة ، ولكن يتوفر عدد أقل من الأعمال للغة العربية بسبب التنوع في أشكال الاحرف بناءً على مواقعها في الكلمات في اللغة العربية. نظرًا لأن الكتابة اليدوية العربية متصلة من اليمين إلى اليسار في الكتابة ، وتحول أشكال الحروف وهياكلها عند وضعها في بداية الكلمات أو وسطها نهايتها أو عزلها ، فإنها تمثل تحديات فريدة. تم تطوير التعرف على الكتابة اليدوية العربية وتصميمه في لغة برمجة Python والشبكة العصبية التلافيفية (CNN) مثل ResNet و LSTM,Gready decoding, CTC بالإضافة إلى ذلك استخدمنا بعض مفاهيم معالجة الصور مثل الفلاتر والتجزئة.

الهدف الرئيسي من المشروع هو دراسة خصائص الكتابة اليدوية العربية وبناء نظام دقيق بقدر ما هو ممكن مع أخطاء أقل ، لأن معظم أنظمة التعرف على الخطوط مبنية للأحرف الإنجليزية واللاتينية والصينية , حيث حقق المودل الخاص بنا دقة مقدارها 95.3 .

Acknowledgement :

This work would not have been possible without the support of our teachers and colleagues. First, we would like to express our special thanks of gratitude to our supervisor Dr. Mujeeb Al-Hakimy. for his consistent support throughout the completion of project. Without his support, the project could not be completed on time. we also would like to mention our colleagues readers for guiding us on writing the report. Similarly, we would like to thank our colleagues and parents for their guidance time to time in making this project.

Table of Contents

LIST OF ABBREVIATIONS

AHRS	Arabic HandWritten Recognition System
AHWS	Arabic HandWritten System
CNN	Convolutional Neural Network
MTO	Morphological Thinning Operation
AHW	Arabic HandWriting System
LSTM	Long Short-Term Memory
PR	Pattern Recognition
GANs	Generative Adversarial Networks
ANNs	Artificial Neural Networks
HMMs	Hidden Markov Models
HWCR	Handwriting character Recognition
RNN	Recurrent Neural Network
HWR	HandWriting Recognition
RBF	Radial Basis Function
DL	Deep learning
MDLSTM	Multi-dimensional Long-Short Term Memory
CRAFT	Character Region Awareness For Text detection
SGD	Stochastic Gradient Descent

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 :INTRODUCTION

1.1 Introduction :

Automatic recognition of handwritten word is one of the applications of artificial intelligence, which is an interesting and important research field in various fields, (AHWS) is the detection of word from images, documents and other sources and changes them in machine-readable shape for further processing. The accurate recognition of intricate-shaped compound handwritten text is still a great challenge. AHWS is the use of technology to distinguish handwritten text inside digital images. The purpose of distinguish handwritten word inside digital images, is to translating the characters into code that can be used for data processing. In other words, handwritten text recognition (HWTR), is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices. The most important step for recognize handwritten word images is the translating, since digital texts can easily by read, edited or analyzed by machine at a very high speed. Because of these intrinsic properties, HWRs are an essential component in many applications. The development of an efficient, accurate Arabic HWR systems has become one of the most important and challenging task because it is a repeated work which is done by humans and because of that error can occur, since it have been used in many applications related to the information retrieval process which has a big revolution recently, to the search engines and data entry operations which can be done in faster and more accurate way. HWR systems has been done in many languages included English, Chinese, Latin and Arabic, The most language considered as a field to recognize its words is English language , in the contrast Arabic language has the lowest studies in Handwritten words and text recognition[45].

1.2 Aims :

1) To train the model on synthetic dataset.
2) To extract handwritten word not just from paper or electronic documents but also from natural scene images.
3) To help the people to store data digitally.

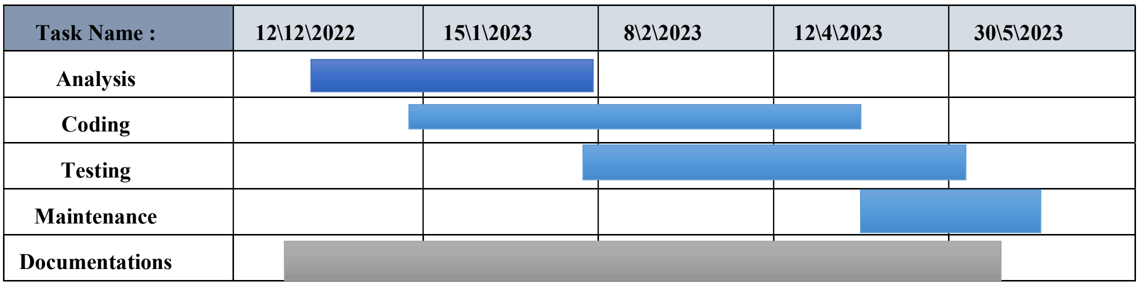
1.3 Objective :

Challenges facing Arabic Calligraphy Recognition is overwhelming like the different types of Handwriting and few public databases are available so the project Goals are:

1) Building a system capable of recognizing Arabic word and outperforming other modern models and researches by achieving high accuracy of results.
2) Make a research in the field of text/ word recognition.
3) Building and parameterization of CNN weights for handwriting recognition problem.
4) Testing of CNN on major text database, proving the effect of various parameterizations in improving text recognition.

1.4 Project Time domain table

Table 1 Time table for project scope



1.5 Problem Statements

The problem of finding a system that identifies handwritten Arabic words and text is still strong and reliable in its far-reaching performance for many reason, still many problems at handwritten. Despite the availability of computing power and progress made so far the capability of handwriting recognition system is still incomparable to human recognition. Because of many of the most important problems:

1) Handwritten problem: no two humans have exactly the same handwriting and even no two sets of handwritings of the same person for the same word exactly the same .

2) Segmentation problem: segment the connected object within image is a problem such as black and white colors for object and its background.

3) Data extraction problem: noise processing and distortion.

4) Language problem: The characteristics of Arabic Writing add difficulty to the process of segmentation and identification.

5) Pre-processed feature extraction problem: Once the initial text is pre-processed feature extraction is performed to identify key information such as loops, inflection points, aspect ratio etc. of an individual character.

6) Very little work and research

1.6 Motivations

There are many motivations that make us choice this project, they are in the following:

1) This project conforms to our collector's specialism that make us solve the problems that facing us.
2) The desire for increasing the generalization capacity and studies of Arabic handwritten word recognition.
3) Presenting research that able to reducing the challenges for make studies about Arabic handwritten word.

1.7 The Limitations and Scope of the Project :

The System is based on word recognition method instead of character recognition. Unlike the character recognition, which recognize the text by the recognition of words, text recognition has to trained with the whole word as input. So, the recognition of such model is constraint to the number of words in the dictionary because in such method we can cover all the words for recognition. Similarly, the accuracy of this method is very good because the model is trained with large no of datasets. The other constraint of this system is it does work offline, so, The primary goal of this project is to propose a handwritten text recognition system to recognize most of the Arabic words, that system should be able to handle cursive handwritten words. There are other limitations of this system such as:

1) Cover images with both clear as well as clear backgrounds
2) Text must be in horizontal orientation only
3) The system only recognize the Arabic alphabetical words
4) The font must be in naskh script.

1.8 Outline of the project

Chapter 1: It provides the overview that deals with the introduction and Problem statement, motivation for this project. It also contains project objective of this work, and method that has been used in this project and scope of study and limitation.

Chapter 2: It provides the overview that deals with the introduction for Arabic Handwriting system and Literature review. Then we study the Characteristics and methods of Handwriting recognition system, pattern recognition and then describe the general steps in the Arabic Handwritten word recognition system and discussion a different preprocessing techniques, then present some related work, then at the end we have the conclusion of the chapter.

Chapter 3: It provides the system specification and methodology, and building the model using ResNet , CTC, LSTM, greedy decoding , and image segmentation ,Proposed system consist of two major parts we will discuss in this section: text detection and text recognition.

Chapter 4: In this chapter we will study the flowchart of proposed system, and our system interface, Use Case Diagram, User model, preprocessing model, Segmentation model, Feature extraction model, Recognition model, and Sequences diagram of our model, and Activity diagram, and the implementation of study.

Chapter 5: In this chapter we view the results of the Arabic AHWS that are ready for assessment after the design and implementation phases are finished. The evaluation criterion for the majority of AHWS is based on the overall recognition percentage and future work and Discussion.

CHAPTER 2 :AHWRS PROCESS AND LITERATURE

2.1 Overview:

Arabic script is cursive (both handwritten and machine (printed text)). Table.(2.1) presents the Arabic alphabet and their different shapes. This makes handwriting recognition very challenging, as we need word segmentation before applying text recognition system. The segmentation methods reported in the literature are far from being robust and very accurate[26], Segmenting Arabic words to characters is very difficult[21]. There are two reasons for this difficulty. First Arabic letter shape is context sensitive. Some Arabic letters have four shapes according to their position in the word (see Table 2.1). Secondly, in Arabic writing dots are very important and they are not few as fifteen letters out of twenty-eight have dots above or below them. Arabic writers do not place these dots carefully on their proper place and this leads to much confusion. Although Arabic writing is horizontal from right to left some letters overlapped vertical. For illustration Fig. (2.1), contains an image of the Arabic word. The image shows three different handwriting. For comparison the Figure also shows some printing for the same word. This is an example for a word that expert Arabic reader recognizes it as one unit without trying to figure out each letter and its dot(s). An analytical automatic recognition system should find the proper segmentation first and then it should find the proper coupling of the dots with its corresponding letter.

Figure (2.1): This figure shows different handwriting and printing for a single Arabic word

Table (2.1) Arabic alphabet and their different shapes

Alone	Initial	Medial	Final	Name	No
ا	ئـ	ـئـ	أ	Hamza	1
ب	بـ	ـبـ	ـب	Beh	2
ة ت	تـ	ـتـ	ـة ـت	Teh	3
ث	ثـ	ـثـ	ـث	Tha	4
ج	جـ	ـجـ	ـج	Jim	5
ح	حـ	ـحـ	ـح	Ha	6
خ	خـ	ـخـ	ـخ	Kha	7
د	-	-	ـد	Dal	8
ذ	-	-	ـذ	Thal	9
ر	-	-	ـر	Ra`	10
ز	-	-	ـز	Za	11
س	سـ	ـسـ	ـس	Sin	12
ش	شـ	ـشـ	ـش	Shin	13
ص	صـ	ـصـ	ـص	Sad	14
ض	ضـ	ـضـ	ـض	Dad	15
ط	طـ	ـطـ	ـط	Ta	16
ظ	ظـ	ـظـ	ـظ	Za	17
ع	عـ	ـعـ	ـع	Ayn	18
غ	غـ	ـغـ	ـغ	Ghayn	19
ف	فـ	ـفـ	ـف	Fa	20
ق	قـ	ـقـ	ـق	Qaf	21
ك	كـ	ـكـ	ـك	Kaf	22
ل	لـ	ـلـ	ـل	Lam	23
م	مـ	ـمـ	ـم	Mim	24
ن	نـ	ـنـ	ـن	Nun	25
ه	هـ	ـهـ	ـه	Ha	26
و	-	-	ـو	Waw	27
ي	يـ	ـيـ	ـي	Ya	28

2.2 Literature Of Handwritten Arabic Recognition:

In the literature, there are good results for isolated Arabic character recognition (in some papers the recognition is more than (97%) [18][16][22]. However, publication on word recognition are few with low recognition accuracy rate[21], this low rate for word recognition accuracy is mainly due to the error in segmentation[26] . Arabic readers generally tend to recognize common Arabic words and names holistically (i.e.

without segmenting them). For new words or non-common ones, the Arabic readers identify the word letters and then recognize the word. In recent years, deep learning has gained great popularity in the pattern recognition filed[23]. It becomes the focus of many researchers since it represents the easiest way to deal with huge data and it automates the feature extraction task.

2.3 Characteristics Of Arabic Handwriting Script:

Arabic script consists of 28 basic letters, 12 additional special letters, and 8 diacritics. Arabic is written (machine printed and handwritten) in a cursive style from right to left. Most letters are written in four different letter shapes depending on their position in a word, e.g. the letter ع (Ain) appears as ع (isolated), عـ (initial), ـعـ (medial), and ـع (final). Among the basic letters, six are disconnected — ا (Alef), د (Dal), ذ (Thal), ر (Reh), ز (Zain) and و (Waw). Disconnected letters do not connect to the following letter and have only two letter shapes each. The presence of these letters interrupts the continuity of the graphic form of a word. We denote connected letters in a word, as a word-part. If a word-part is composed of only one letter, this letter is in its isolated shape. For example, the Arabic word مساحة (masahhat) "area" consists of 5 letters (from right to left): م(Meem),س (Sah), ا(Ahhh),ح (Haa), ة(ha), , which are realized initially مـ, medially ـسـ , finally ـا, medially حـ, connected ـة, respectively. This word has two word-parts (from right to left): حة ,and مسا .

Finally, in Arabic script a top-down writing style called vertical ligatures is very common — letters in a word may be written above their consequent letters. In this style, the position of letters cannot be predefined relative to the baseline of the word. This further complicates the recognition task, particularly in comparison in figure (2.2). Word (a1) (Kalem) "pencil" is a result of moving the dot to the left from word (a2) (Film) "Film". Word (b1) (Eyn) "Eye" is a result of eliminating the dot above the first letter from word (b2) (gaeen) "letter gaeen ". Additional case happens with the letter ججwhich can be confused with a set of two or three letters when the dots are misplaced. In (c1) and (c2) we can see a close example which is much more confusing with handwriting.

	a	b	c
1	قلم	عين	جديد
2	فلم	غين	حديد

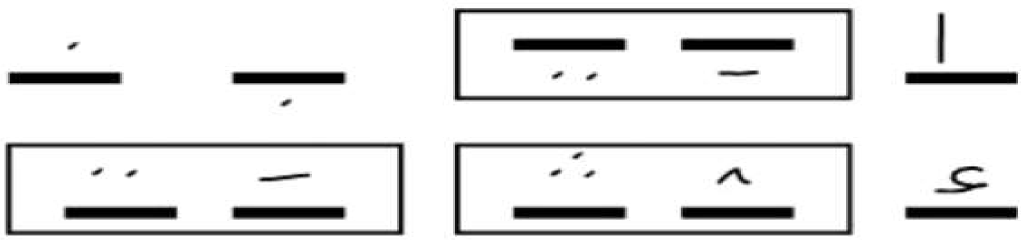


Figure (2.2) Delayed strokes in Arabic script may appear under or above the letter body.

2.4 Handwriting Recognition(HWR):

Handwriting Recognition (HWR) is a type of technology that can be used to “read” the writing in images of handwritten documents. Handwriting Recognition (HWR) is the capability of computers and mobile devices to receive and interpret handwritten inputs. The inputs might be **offline** (scanned from paper documents, images, etc.) or **online** (sensed from the movement of pens on a special digitizer), for example: A handwriting recognition system also includes formatting, segmentation into individual characters, and training a language model that learns to frame meaningful words and sentences. Let’s say you wrote an essay by hand, back when you were at school and you now want to have that essay as digital text on your computer. With the HWR software that illustrated in figure (2.3), you could take a photo of the essay, run it through the HWR software and get that same essay as a digital text file.

Figure (2.3): Offline handwriting system

The kind of handwriting recognition described in figure (2.3) is known as “offline handwriting recognition”. That is because it involves images of text that has already been written. There is also “online handwriting recognition”. This is software that generates digital text from handwriting as you write it, usually with a tablet and stylus.

2.5 Methods Of Handwriting Recognition:

There are two types of handwriting recognition depending on when the identification takes place, as they are illustrated in figure (2.4).

Figure (2.4): Online vs. Offline Handwriting recognition system

2.5.1 Online Handwriting Recognition

Online handwriting recognition involves the automatic conversion of text as it is written on a unique digitizer or digital pad with a sensor that picks up the pen tip movements and uses this dynamic data to evaluate the character and words as they're being written. The main features that make the online handwriting recognition system predict the text are:

1) Line quality.

2) Speed of writing/word.

3) Execution of letters.

2.5.2 Offline Handwriting Recognition

Offline handwriting recognition involves the automatic conversion of an image of text into letter codes usable within computers and text processing applications. The data obtained in this form is a static snapshot of the handwriting. Without information on pen pressure, stroke direction, etc. It's more difficult to achieve accuracy with offline recognition. However, it's still highly in demand, especially considering the need for digitizing existing historical and archival documents.

2.6 Handwriting recognition techniques:

There are several methods of recognizing human handwriting with machine learning, and new technologies are bound to emerge. Here, we'll summarize the most prominent handwriting recognition approaches and algorithms [26] .

2.6.1 Encoder-Decoder And Attention Networks

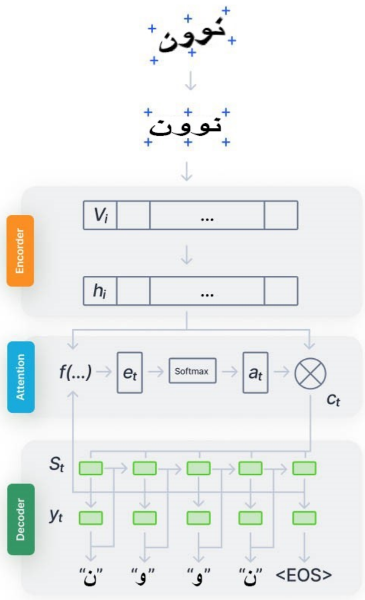


Figure (2.5): Architecture of encoder-decoder model coupled with attention network

The training handwriting recognition systems always suffers from the lack of training data, as it's impossible to create a set with all combinations of languages, stroke patterns, etc. To solve the problem, this method leverages the pre-trained feature vectors of text as a starting point. State-of-art models hint towards using an attention mechanism in combination with RNN (recurrent neural network) to focus on the useful features at each time stamp.

The complete model architecture can be divided into four stages:

1) Transformation

A CNN network is trained for localization. It takes an input image and learns the coordinates of fiducially points used to capture the shape of the text. Since handwritten words can be tilted, skewed, curved, or irregular, the input word images are normalized by applying some transformations.

2) **Feature extraction**

Features in the handwritten text include stroke angles, series of tilts, etc. for which a ResNet-type of architecture can be used to encode the normalized input image into a 2D visual feature map.

3) Sequence modeling

The features extracted in the previous step are used as a sequential frame (just like text from left to right). It is decoded using a Bidirectional LSTM for sequential modeling to retain the contextual information within a sequence from both sides and recognize each character independently while taking into account the higher-level abstractions.

4) **Prediction**

The output vectors containing the contextual information from the last decoder are transformed into words, first, the output vector needs to be fed into a fully-connected linear layer to get a vector of the size of the vocabulary, which is used to train the model. Then, the softmax function as an activation function is applied to this vector in order to get a probability score for each word in the vocabulary.

2.7 Pattern Recognition:

Handwriting recognition is an application in the field of pattern recognition. Automatic recognition, description, classification and grouping of patterns are important problems in many engineering and scientific disciplines. Pattern recognition (PR) is the study of how machines can observe the environment, learn to distinguish patterns of interest from their background, and make sound and reasonable decisions about the categories of the patterns. The primary goal of pattern recognition depends on whether it is a supervised classification or unsupervised classification. In supervised classification, the input pattern is identified as a member of a predefined class. In unsupervised classification the pattern is assigned to an unknown constructed class. Handwriting recognition really falls under supervised classifications as handwriting examples are used in building the recognizer.

A model for pattern recognition is shown in Figure (2.6). As can be seen it is operated in two modes: training (learning) and classification (testing). A PR system needs to be trained to obtain a recognition or classification model for use during classification. In training mode, features representing input patterns are extracted and used for training to partition the feature space. Some feedback from the learning stage allows the optimization of the preprocessing and feature extraction or selection strategies involved. At the end of training, parameter values for the classifier are obtained. In classification mode, the trained classifier is used to classify the input pattern into one of the pattern classes.

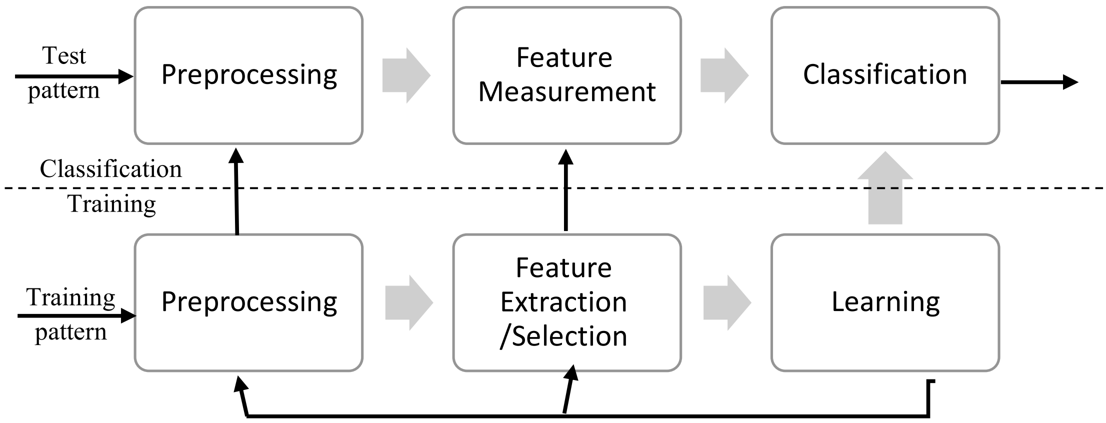


Figure (2.6): Model of Pattern Recognition System

Depending on complexity, some PR application might require extensive computation during training, especially in applications involving the processing of large data. There can also be huge data sets needed during the training stage of the systems. There are many approaches in PR, but it is important to note that there is no single optimal approach for all and multiple methods and approaches might need to be combined and used in a single system. This applies to handwriting recognition as well[45].

2.8 Arabic Handwritten Character Recognition Systems:

Character recognition is one of the leading applications of pattern recognition despite its many limitations. This process's main purpose is distinguishing between the individual characters and between the words. The general Arabic handwritten character recognition system processes according to some researches like[24],[6] ,[5] ,[10] and others, can be listed as shown in below in figure (2.7):

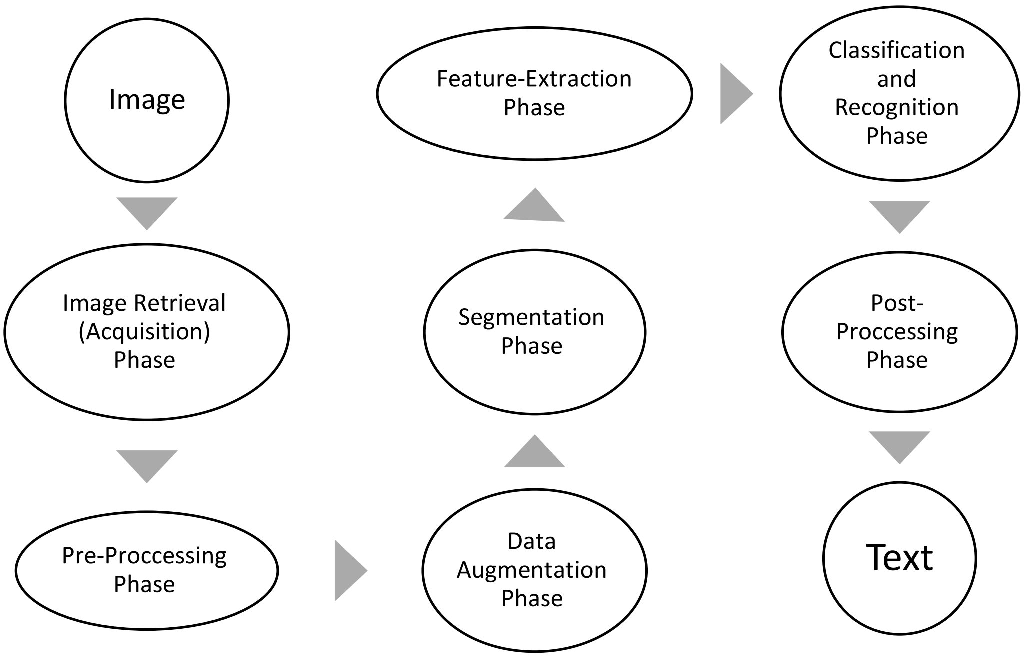


Figure (2.7): General steps for Arabic handwritten text/character recognition

2.8.1 Image acquisition phase :

This is the initial phase (task) in the Arabic handwritten text recognition system. The main objective of this phase is to retrieve the data (images) from different sources and convert them into the digitalized form. Digital cameras, scanners and tablets are examples of the text acquisition methods. To increase the speedup of the scanning process, it is required to select a suitable device (i.e., scanner) with high throughput, sensing tool, transport mechanism and document feeder.

2.8.2 Preprocessing phase :

The preprocessing phase is the second task which is very important in any recognition system[39]. The purpose of this step in handwritten text recognition is to improve and enhance the readability of the textual image and remove unnecessary details from it. The preprocessing phase usually includes several operations such as binarization, thinning, alignment, padding, centering, slant correction, noise removal, smoothing, baseline detection, skeletonization, skewing and normalization. The system can apply one or more from the preprocessing operations. It can also bypass this phase without applying any of the operations if the input data are previously preprocessed. Figure (2.8) shows a summary of the different preprocessing techniques.

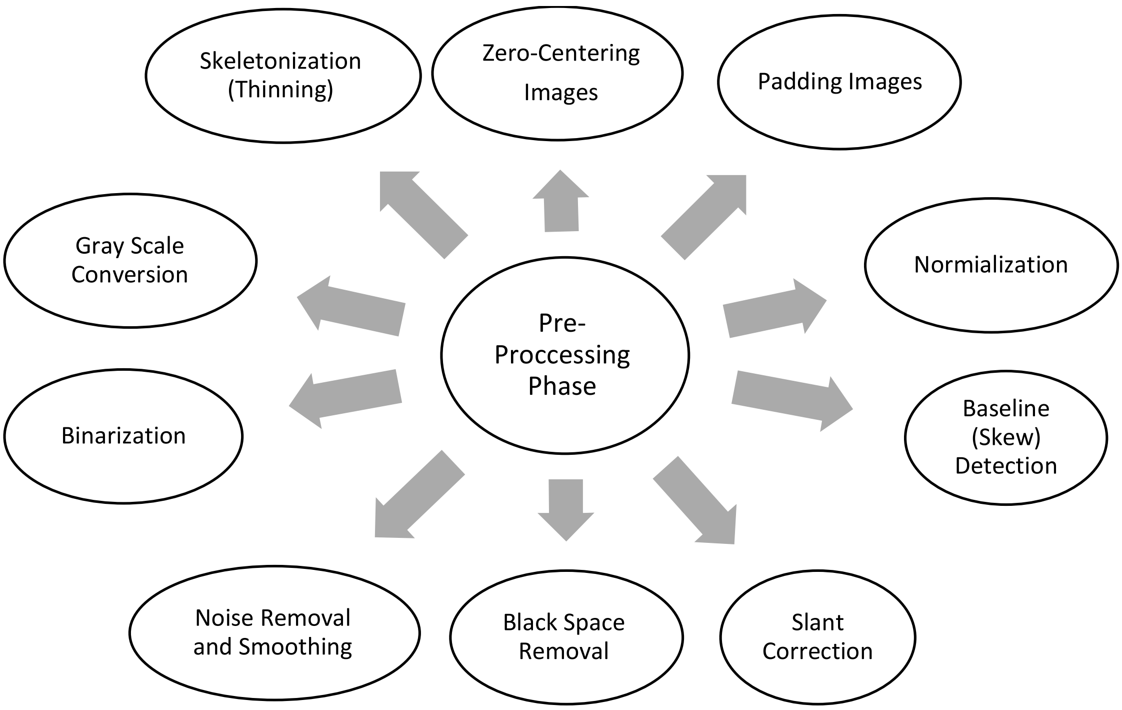


Figure (2.8): Summarization of the different preprocessing techniques

2.8.2.1 Grayscale conversion

The colored image in this process is converted to a grayscale image, So, instead of working with a three-channel image, we can work only with a one-channel image. There are multiple ways such as averaging, human eye correction (weighted), desaturation, and decomposition and single-color channel methods.

2.8.2.2 Binarization

It is also called thresholding which is the process of converting a text image into a binary (bi-level) format[44]. In other words, it is called a black and white image. The values of background pixels are 1 for white and0 for black. This process improves processing speed**.** They are many methods with different categories used into image binarization such as the fixed threshold method**,** mean value threshold method and Otsu method**.** Figure (2.9) shows two sample images after applying the three methods. The sample images after applying the three methods. An Original. B Using fixed threshold (threshold=100). C Using mean threshold, d Using Otsu threshold.

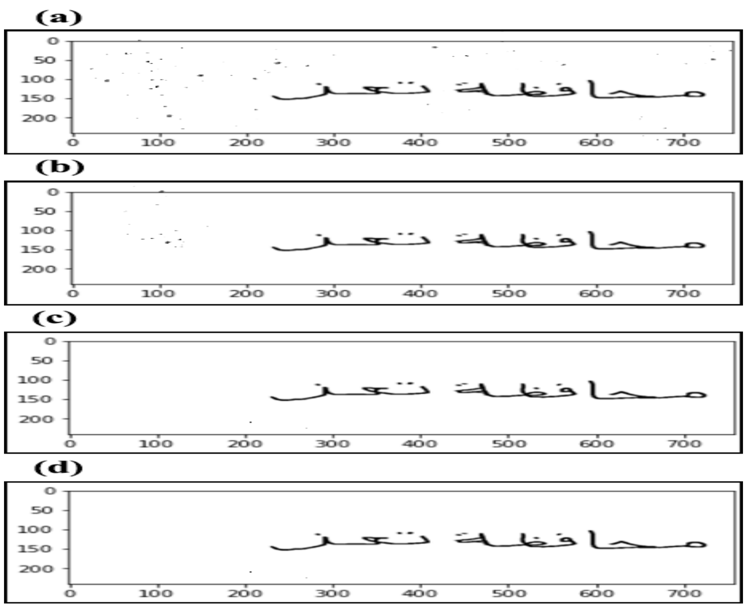


Figure (2.9): Fixed threshold method, mean value threshold method and Otsu method

The fixed threshold method deals with assigning a fixed value of threshold (i.e., *threshold* ¼ 100). If the pixels values are more than or equal to that threshold value, the output pixels value is 1 and 0 otherwise according to Eq. (2.1).

$$G(x, y) = 1 \rightarrow f(x, y) \geq \text{threshold and } 0 \rightarrow \text{otherwise.} \quad (2.1)$$

The mean value threshold method is similar to the previous one unless that the threshold is not predetermined by the user but is determined by taking the mean value of all of the pixels according to Eq. (2.2) .

$$\text{Threshold} = \frac{\sum_{i=0}^{widt} \sum_{j=0}^{height} f(i,j)}{(widt \ * heigh\ )} \quad (2.2)$$

The Otsu threshold method is used in the automatic binarization based on the histogram shape. It involves iterating through all of the possible threshold values and calculating the measure of spread for the pixel levels in each side of the threshold. The algorithm target is to find the threshold that minimizes the sum of the foreground and background spreads.

2.8.2.3 Noise removal and smoothing

All the data acquisition methods are affected by noise, hence, there is no ideal situation for no noise attached to data. Noise removal is the process of removing distortions and unrequired small objects that are not part of the writing[40]. Smoothing is the process of removing noise and data variations by using mathematical morphology operations[38]. There are multiple methods used to reduce image noise and perform smoothing such as filtering and morphological operations. In image processing, filters are used to suppress the high frequencies in the image (i.e., smoothing the input image) or the low frequencies (i.e., enhancing or detecting edges in the input image). Filtering is a neighborhood operation that uses the kernel for performing that. A kernel is a small window (matrix) that passes through all of the image pixels where the pixel in the center is replaced by a value according to the selected filter.

2.8.2.4 Black space removal

In this process, the unrequired black pixels are removed from the image. The count of black pixels is calculated from the image borders until the white pixel is found. The bounding box can be used to eliminate the space around the character image.

2.8.2.5 Slant correction

It is a way to correct and eliminate the different character slants[11]. A shear transformation is applied after the average slant of the character**.**

2.8.2.6 Baseline (skew) detection

The baseline is the imaginary horizontal line that connects the words characters together[31],[33] . This process is helpful in determining the words structural features such as dots, ascenders and descanters. There are multiple methods that can be used for detecting the baseline such as the horizontal project methods or the rotation angle. The rotated angle in the lines requires to be fixed, hence, baseline (skew) correction is used. Skew is considered one of the image distortions forms. Skew correction is used to correct the orientation angle of the image**.**

2.8.2.7 Normalization

Handwriting has different styles and sizes. Therefore, the normalization process is one of the most important tasks in the text recognition process. It is the process of reducing the variations between the images of the text and to adjust the size of the character or the word. Size normalization is commonly used to reduce size variation and adjust the character or word sizes to an identical size[32]. There is another techniques used to normalize the data such as Z-score normalization is a normalization technique, Batch normalization is a normalization and regularization technique.

2.8.2.8 Padding images

Images in many datasets are not the same size as the width, height or both may differ. The input images sizes to most of the recognizers in the different phases must be the same. The first approach is to crop large images to the size of the minimum image, but in this case, data will be lost. The second approach is to resize the image to a suitable aspect ratio, but in this case, large images data may be distorted. The third approach is to pad the image with white to the maximum width and height existing in the dataset. The white is added on both sides equally, so the image is stilling in the center. The latter approach is better than the first two approaches as it does not lose nor distort the data of the image.

2.8.2.9 Zero-centering images

Centering the image can be achieved by subtracting the dataset mean pixel values from the pixel values of the images as shown in the following two Equation (2.3), Equation (2.4):

$$x_{center} = x - \frac{1}{Heigh\ *Width} * \sum_{i=1}^{Hight} \sum_{j=i}^{Widt} x_{i,j} \quad (2.3)$$

$$y_{center} = y - \frac{1}{Heigh\ *Widt} * \sum_{i=1}^{Hight} \sum_{j=i}^{Width} y_{i,j} \quad (2.4)$$

2.8.2.10 Skeletonization (thinning)

The skeletonization (also known as thinning) [27]is one of the morphological operations that can be applied to images. It is based on reducing the foreground regions in a binary image. The pixels on the boundaries of the image are removed, but the main content of the image should outbreak apart.

2.8.3 Data augmentation :

Data augmentation is an approach that enables us to increase the diversity and amount of data available for training models without the collection of any new data. There are different techniques that can be applied such as rotating images, brightness change, zooming, shearing, shifting, cropping, padding and flipping[12].

2.8.3.1 Rotating images

A copy of the image is rotated toward the right or the left by a predefined small angle. Performing this to the whole dataset to make the dataset more robust to the issue of different writers can write the same word in different angles on blank papers.

2.8.3.2 Flipping images :

A copy of the image is flipped horizontally, vertically or both according to a predefined boolean flag (true or false). This can affect some of the symmetric characters positively such as “ب ". This may also affect some of the characters negatively such as “ح “which will be meaningless if it is flipped horizontally. Hence, a special treatment should be considered using the flipping technique.

2.8.3.3 Cropping images :

A cropped image copy is generated and rescaled to the original shape to match other images shapes. Applying cropping and rescaling on an image can be called zooming.

2.8.3.4 Generative adversarial networks (GANs)

Generative adversarial networks can be also used in data augmentation. They have been proved in many cases to be able to produce augmented data that are like the real data. They also have been successfully applied to various image generation tasks as a useful approach for data augmentation.

Figure (2.10) shows a simple appliance of rotation, flipping and cropping (with zooming) on the “ح “character. It showed that the horizontal flipping changed the meaning of the character, while zooming increased the number of black pixels that can be used in feature extraction.

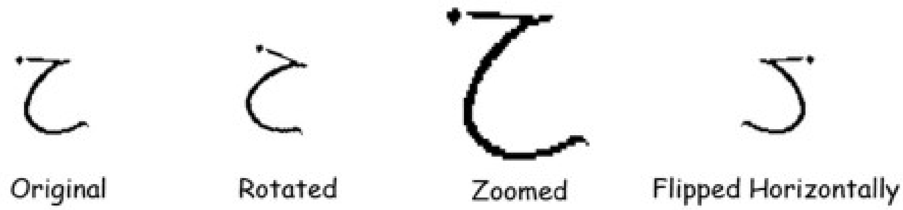


Figure (2.10): Rotation, flipping and zooming appliance on the “ح“character

2.8.4 Segmentation phase :

The segmentation phase performs segmentation of texts into smaller units. They can be lines, words and characters. It is an important phase as it can effect the recognition accuracy rate. There are many types of segmentations such as segmenting a page into line, segmenting a line into words and segmenting a word into characters. Figure (2.11) shows the different steps in segmenting a page.

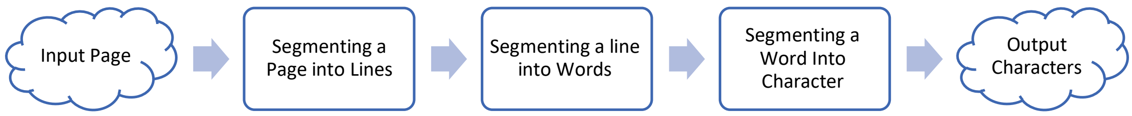


Figure (2.11): Steps of segmenting a page

2.8.4.1 Segmenting a page into lines

This process is used to divide the whole text into separated simpler lines. In general, the paragraph or page consists of several lines. In text recognition, the text has many lines segmenting into separated lines. There are certain methods used for this purpose, such as horizontal projection [43] .

2.8.4.2 Segmenting a Line into Words

After segmenting lines, the line is segmented into words. Segmenting the line into words depends on the space between the words. However, there are some characters which do not connect to a subsequent character in a word and this causes a separation of the word into sub-words. Longer spaces separate words while short spaces separate sub-words. Therefore, most researchers assume in their technique that the space between words is bigger than the space between sub-words.

2.8.4.3 Segmenting a Word into Characters

This part of segmentation involves segmenting the word into individual characters. As described in[41]. The segmentation points are identified at the end of a character and at the beginning of the next one. The cursive nature of the Arabic language makes segmentation of words into individual characters a difficult task.

Techniques Applied For Character Segmentation

1. **Segmentation Technique based on Vertical Projection:** The vertical projection technique reduces two-dimensional information into one dimension. This technique is based on the fact that the thickness of the connection stroke between the characters in the word is always less than the other parts. The vertical projection is useful for segmenting the words, sub-words and characters, while the horizontal projection is usually used for line segmentation and baseline extraction.

2. **Segmentation Technique based on Thinning:** In character recognition, the skeleton of a character provides the essential information about the shape of the character.

3. **Segmentation Technique based on Contour Tracing:** Another technique based on contour tracing of a word is also used for segmentation. This technique is achieved by tracing the outer contour of the main body of the word.

4. **Segmentation Technique based on Artificial Neural Networks:** ANNs are used to verify the valid segmentation points in Hamid and Haraty [42] proposed a technique for segmenting handwritten Arabic texts based on ANNs. They identified pre-segmentation points by using topographic features of each connected block of characters, such as black pixel density and holes. The ANNs are then used to verify these segmentation points. The potential points are manually classified into valid and invalid segmentation points.

2.8.5 Feature Extraction Phase :

In handwritten texts, the features capture the information extracted from the text image. This information should have the essential characteristics of the character or the word which make it different from another, in other words, filtering out all attributes and preserving the properties that make one character or word different from another. This information is passed onto the classifier to assist in the classification process. Feature extraction techniques differ from one application to another. Techniques that succeed in one application, may not be successful for other applications. Therefore, the selection of the method for feature extraction remains the most important step for achieving high recognition accuracy. Features of handwritten texts can be classified into the following categories:

2.8.5.1 Structural Features:

Structural features are the most common features used by researchers. Structural features depend on the category of the pattern to be classified. For an Arabic text, for example, the features include dots and their position, strokes, width and height of the stroke, directions, intersection of line segments and loops.

2.8.5.2 Statistical Features:

Statistical features analyze the spatial distribution of pixels by counting local features at each pixel and deriving a set of statistics from the distributions of the local features. The major statistical features of the character include zoning, where the character divides into overlapping or non-overlapping zones and the density distribution of character pixels in different regions is analyzed.

2.8.5.3 Global Transformation:

Global transformation techniques have the ability to convert pixel representation into a more compact form. These techniques can represent the signal by a linear combination of a series of simpler well defined functions. The series expansion provides a compact encoding by the coefficients of the linear combination.

2.8.6 Classification phase :

Classification is the process of identifying and recognizing the object by comparing its features to one of the classes. It is assumed that the classes and model are inputs as the classification is a supervised learning approach. The training process is performed to teach the classifier model on the given features. The output model should have the ability to recognize unseen objects. Classification methods can be based on one of the following: machine learning techniques, graph-theoretic methods such as decision trees, syntactical methods, statistical methods or mathematical methods. The classification methods can be divided into three types: structural, statistical and neural network classifier.

2.8.6.1 Structural Classifier

The input pattern is classified based on its components or primitives[37]. The classifier identifies the primitives of the character first and then identifies the character by a set of primitives.

2.8.6.2 Statistical Classifier

Statistical classification methods map a fixed length of feature vectors with a partitioned space [25][28].one of the most efficient statistical classification methods is the hidden Markov models (HMMs). HMMs are applied successfully in speech recognition application then the researchers implement HMMs in character recognition.

2.8.6.3 Neural Network Classifier

Artificial Neural network (ANNs) is a widely used classifier in the pattern recognition field[29]. ANNs are a non-linear system and may be characterized according to a particular network topology, characteristics of the artificial network and learning algorithm used. The architecture of ANNs used with Arabic HWCR systems is generally broken into three layer: input, hidden and output. The network is trained to learn the correct classification output for each training example.

2.8.7 Post-processing phase :

The post-preprocessing stage improves the output from the previous stage by refining the taken decisions. This can be done by using the context. This stage is responsible for outputting the best solution and is often implemented as a set of techniques and methods that rely mainly on the character dictionary, frequencies and other information. Grouping, error detection and error correction can also be applied in the post-preprocessing phase.

2.9 Related Work:

Handwritten Arabic character recognition systems were extensively studied and developed for many years. However, consistent comparison of different proposed solutions remains difficult, because in practice, diverse private unpublished databases are used, where different number of handwritten words and characters are stored [34] [30] [35][36]. Usually, a word recognition system is based on three main steps: preprocessing, features extraction, and recognition (classification).

In [8]: The Authors deployed a deep learning approach based on Multi-Dimensional Long Short-Term Memory (MDLSTM) networks and Connectionist Temporal Classification (CTC). The MDLSTM has the advantage of scanning the Arabic text-lines in all directions (horizontal and vertical) to cover dots, diacritics, strokes. In this work the final results of handwritten Arabic text lines recognition rate is up to 80.02% on KHATT dataset.

In [20]: The Authors proposed segmenting the word based on a vertical smoothed histogram projection . After that, they extracted the characteristics of each cell using the Zernike and HU moments, which are invariant to rotation, translation and scaling. hen, the sub-character is estimated at the lowest level of the Bayesian Network (BN) and the character is estimated at the highest level of the BN. The developed system was tested on the IFN/ENIT database and achieved a word recognition accuracy of 78.5%.

In [9]: The presented model trained for recognize Handwritten Arabic names with the SUST-ARG dataset. The SUST-ARG dataset contains 8028 names, written by 2007 students. The achieved result of training a Conventional Neural Network (CNN) is up to 99.14%.

The work of[4]: Presented A handwriting recognition system for Arabic words using a few important structural features and based on a Radial Basis Function (RBF) neural networks is proposed. The recognition system evaluated using a development database, which contains 24 words of Algerian city names were written by fifteen writers. The Authors used the dataset that consist of 7200 samples and achieved The accuracy of the developed system is over 95%.

In [1]: The Authors used The transfer learning technique to develop the Offline Arabic handwritten word recognition by using AlexU-W dataset and IFN/ENIT dataset. AlexU-W consist of 25,114 images were collected from 109 unique words written by 907 different writers. IFN/ENIT consist of 32,432 images for 937 unique names written by 500 different people. So the accuracy achieved in this work is up to 96.11%.

In [15]: The Authors presented a model CNN based Hidden Markov Models(HMM) for Arabic handwriting word recognition. So they used the IFN/ENIT database that consists of 26,459 images for 946 handwritten Tunisian city names written by 411 people. This model achieved recognition rate of 89.23%.

In [2]: The proposed system based on deep Convolutional & Recurrent Neural Network CNN/RNN, and the Authors trained their system on IFN/ENIT extended database(augmentation database) in order to improve their results in proposed system. The system achieved accuracy of training up to 91.49%.

In [13]: The Authors presented an approach for extracting features and preprocessing for image text. The proposed model gets a single line of Arabic text, which convert and segments into words and then segments into letters. The model achieved an accuracy up to 83%.

Khemiri et al. in [14] discussed that Bayesian Network (BN) architectures and Convolutional Neural Network (CNN) are different types of ML in the process of defining the distinctive features of the dataset. BN needs the domain experts to manually define the features to be used as input to the NB algorithm. While CNNs learn the feature automatically by training the model on the training data. Therefore, they investigated the performance of these two architectures and combinations of the architectures in recognizing the images of Arabic words in IFN/ENIT. They train and evaluate five BN architectures e.g., Naïve Bayes (NB) and Dynamic and Naïve Bayes (DNB) and 6 CNN architectures. The DNB model achieved the best performance in the BN category with accuracy of 88.27%. The best result achieved by the CNN model was 92.7%. The combination of the previous best models achieved an accuracy of 95.20%, improving the performance of the original two models.

In [3]: The Authors applied a series of preprocessing techniques to 100 classes selected from the handwritten Arabic database of (IFN/ENIT). Then, they trained the k-nearest neighbor's algorithm (k-NN) algorithm to generate the best model for each feature extraction descriptor. The best k-NN model, according to common performance evaluation metrics, is used to classify Arabic handwritten images according to their classes. The model achieved an accuracy up to 99.88%.

In [17]: The Authors proposed several combination rules between multilayer perceptron (MLP), support vector machine (SVM) and Extreme Learning Machine classifiers trained with Chebyshev moments (CM) and Statistical and Contour-based Features (SCF) features. They consider a second level of combination that merges three best rules among the proposed ones. The system is evaluated on the IFN/ENIT database and achieved an accuracy up to 96.82%.

The authors in[19]: introduced a model for recognizing Arabic printed text using linear and nonlinear regression. In that work, text in images were initially thinned and segmented into sub words. Next, the relations between word segments were represented using a numerical coding scheme that represented characters as a sequence of points, lines, ellipses and curves. Using that scheme, a unique code was established for each character form and a unique list of codes were used to recognize each font type. Finally, linear regression technique was used to validate the representations against a ground truth table using distance measures. The model was evaluated using (14000) words samples and it has achieved an accuracy rate of (86%).

2.10 Conclusions:

This chapter presents the general overview for extracting features and preprocessing for text images to study the previous accuracy of recognition handwritten Arabic text. Many algorithms that used during the preprocessing phase comprehensive binarization and noise removing using median filter and thinning using morphological filter, and tackle some of image suffering from an extreme case of nonuniform illumination by using adaptive binarization technique. It furthermore overcomes many of handwritten word variations. Choosing suitable features for recognizing handwritten Arabic word will give best recognition accuracies, despite the fact that, there are several challenges with some characters, however, the overall recognition rate is good particularly when compared to other handwritten Arabic word`s systems. After checking the recognition accuracy of word, and text in other systems it is found that the recognition rate up to (99.88%) [3] **,** they have used The IFN/ENIT dataset and it is not balanced, and it contains only number of town names and is not suitable for general handwritten text recognition , also MDLSTM networks are more complex compared to traditional LSTM networks, which can make them computationally expensive and slower to train, they can have various characteristics from one to other writing style. Letters with dots furthermore tend to have low recognition accuracies because the differences in drawing the dots (dots may be linked to the main body of the letter or even can be removed in other writing styles) give inaccuracies. After we studied the several of systems to Arabic Handwriting recognition we found the ratio of accuracy up to (99.88%) [3], from these systems build to recognize the Handwriting for words, so we presented some steps and suggestions to create system that able to recognition Arabic words that write by hand. The first step of our project to improve the studies for Arabic handwritten text recognition we generate the dataset contains more then Million words,then we tried to applying these steps to create the system and presents to our university as graduation project.

CHAPTER 3: SYSTEM SPECIFICATION AND METHODOL

3.1 Overview :

This chapter presents the methodology of the proposed approach in this study, which consists of some stages, by using some techniques those described in chapter 2. Proposed system consist of two major parts we will discuss in this section: **text detection** and **text recognition**.

Text Detection: Text detection model are designed to identify areas that are likely to contain text, disregarding non-textual regions, Text detection refers to the process of locating and identifying regions within an image that contain textual content. The goal is to identify the bounding boxes or outlines around text regions in order to extract and isolate them, in this study we are using craft model to text detection, and detection relied on several predetermined steps, as Data Preprocessing which is noise removal

Text Recognition: We used transfer learning using a ResNet model to train our own database of more than a million Arabic words, and we also use (CNN) such as ResNet, Long Short-Term Memory (LSTM), and Connectionist Temporal Classification (CTC), By combining CNNs for feature extraction, LSTM networks for sequence modeling, and CTC for handling variable-length outputs, the text recognition system can effectively process input images and generate accurate text predictions. This approach has demonstrated good performance in various text recognition tasks .

3.2 Project Roadmap:

A recognition system of handwritten Arabic words is proposed where three main phases are included: preprocessing, feature extraction, and classification. In an HWRS the recognition rate depends on a number of factors. Two very important factors are the quality of input images and effectiveness of pre-processing. Once the sample image is acquired, pre-processing is required to enhance the signal for better performance. After pre-processing, features are extracted using overlapping blocks for each word image. Means computed for each block constitutes the feature vector and ending up with a feature matrix for all the word images. Finally, the CNN classifier is applied to decide to which class of an unknown word belongs. Figure (3.1) illustrates the block diagram of our proposed system.

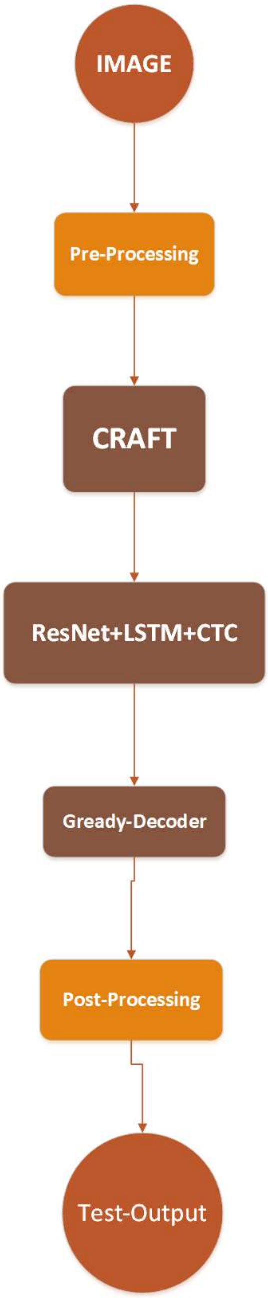


Figure (3.1): Block diagram of our proposal system

3.3 DATASET :

CNN requires a large dataset, so we used more than one different dataset. Some of them consist of characters only, and some of them consist of words and texts, the table (3.1) shows that accessibility to most of the available databases.

Table (3.1) Arabic handwriting databases

Reference	Databases	Type (size)	Availability	Remark	Accuracy Results
Altwaijry and Al-Turaiki [7]	Hijja	character (47,434) images	Freely Accessible	Written by children and also includes each character in their connected form	88.5%
[46]	UHWR	Character (175500) Images	Freely Accessible	There are 39 alphabets in Urdu language, We have collected it and extracted only the letters of the Arabic language	97.24%

- **Hijaa**: is a dataset for handwritten Arabic letters collected from Arabic-speaking school children between the ages of 7 and 12. Data were collected in Riyadh, Saudi Arabia from January to April, 2019. It represents a total of 47,434 characters written by 591 participants in different forms, contains of the various letter forms for each letter [7]. Figure (3.2): show the size of hijaa rows & column.

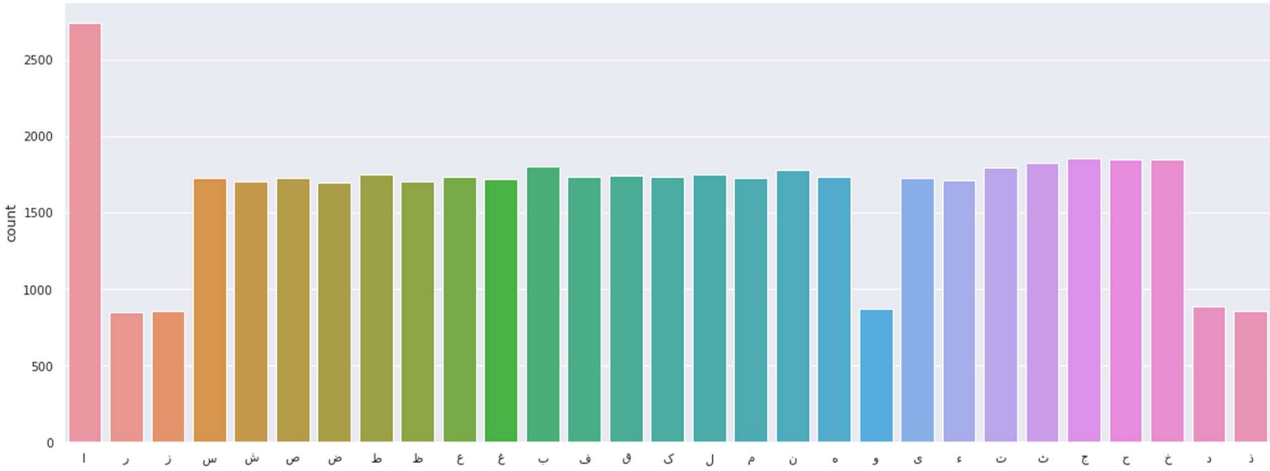


Figure (3.2): Show the number of hijaa characters

By using this Hijja dataset we have achieved a recognition accuracy of (88.24%) after (50) epochs, and using data augmentation.

- **UHWLR:** Urdu Handwritten Letters Recognizer, There are 39 alphabets in Urdu language, This Experiment is done using 175500 images, each character contains 4500 images, The dimension of the image is 1x50x50x1, After we processed this data and extracted the Arabic characters only, the number of images became 134999, and the number of characters became 30 characters, as ۃ separate character and ء separate letter to increase accuracy.

	255.0	255.0.1	255.0.2	255.0.3	255.0.4	255.0.5	255.0.6	255.0.7	255.0.8	255.0.9	...	255.0.1664	255.0
0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	...	255.0	
1	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	...	255.0	
2	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	...	255.0	
3	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	...	255.0	
4	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	...	255.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
134994	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	...	255.0	
134995	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	...	255.0	
134996	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	...	255.0	
134997	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	...	255.0	
134998	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	255.0	...	255.0	

134999 rows × 2501 columns

Figure (3.3): Shows number of column and rows inside the UHWLR dataset

Table (3.2): Shows characters and their count inside the dataset

character	ا	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض
counts	4499	4500	4500	4500	4500	4500	4500	4501	4500	4500	4500	4500	4500	4500	4500
character	ط	ظ	ع	غ	ف	ق	ك	ل	م	ن	ه	و	ي	ة	ء
counts	4500	4500	4500	4500	4500	4500	4499	4500	4500	4500	4500	4499	4501	4500	4500

by using this dataset we have achieved a recognition accuracy of (97.24%) after (100) epochs, and using data augmentation .when no using data augmentation we have achieved a recognition accuracy of (99.9%) , but in testing we have achieved a recognition accuracy of (94.34%) this is called overfitting .

- **KHATT: KFUPM Handwritten Arabic Text Database :**

The database includes 2000 similar-text paragraph images and 2000 unique-text paragraph images and their extracted text line images. The images are accompanied with manually verified ground-truth and Latin representation of the ground-truth.

- **Our dataset used In this study we are using :**

The main objective of this work was to recognizing Arabic Word handwriting that was printed using fonts that mimic Arabic handwriting styles. For that purpose, no suitable dataset was found and consequently a number of custom datasets were compiled to serve the purpose i.e. Arabic Multi-Fonts Dataset (AMFDS). These dataset were prepared using the (7) fonts depicted in Fig. 3.4 below. In this respect, a custom toolkit for generating the datasets was prepared. This toolkit can be used to generate any number of text image samples using any required font type. It can also be configured to generate samples as separate image files or as a single binary repository for all the samples. Table (3.3): shows the main characteristics of each generated dataset. As presented in the table, the ((A) Arslan Wessam A (A) Arslan Wessam A) font was selected to generate the single-font datasets. This font type was selected because its printing style clearly exhibits cursive structures that mimic Arabic hand writing script. Similarly, the (18 Khebrat Musamim Regular) and the (AAAGoldenLotus Stg1_Ver1

Regular) font's types were selected to generate the two-font's datasets. Finally, datasets from (1) to (7) in Table (3.3) were generated using the font types that are presented in Fig. (3.4).

Font Type	**Font sample**
(A) Arslan Wessam A (A) Arslan Wessam A	مرحبا بكم في جامعة تعز
18 Khebrat Musamim Regular	مرحبا بكم في جامعة تعز
AAAGoldenLotus Stg1_Ver1 Regular	مرحبا بكم في جامعة تعز
Bahij_Myriad_Arabic-Bold	مرحبا بكم في جامعة تعز
Janna LT Bold	مرحبا بكم في جامعة تعز
KFGQPC Uthmanic Script HAFS Regular	مرحبا بكم فى جامعة تعز
Times New Roman	مرحبا بكم فى جامعة تعز

Figure (3.4): The Set of the Selected Fonts

Table (3.3): The Prepared Datasets

Dataset #	Number of simples	Dataset size	Fonts	Augmentation
1	1276654	242MB	(A) Arslan Wessam A (A) Arslan Wessam A	Curve ,rotate,zoom
2	1276654	242MB	18 Khebrat Musamim Regular	Curve ,rotate,zoom
3	1276654	206MB	AAAGoldenLotus Stg1_Ver1 Regular	Curve ,rotate,zoom
4	1276654	213MB	Bahij_Myriad_Arabic-Bold	Curve ,rotate,zoom
5	1276654	219MB	Janna LT Bold	Curve ,rotate,zoom
6	1276654	242MB	KFGQPC Uthmanic Script HAFS Regular	Curve ,rotate,zoom
7	1276654	158MB	Times New Roman	Curve ,rotate,zoom

Further, each dataset is comprised of two main data files: a labels file and binary file. The labels file is a normal text file that contains details about the word samples, this includes the Arabic word represented by the image, the font type, the font style, the font size and a value that represent the starting index of that image in the binary file. Hence, the byte stream of the designated image begins at the starting index and spans to a length equals to the image’s size (in bytes). This binary file represents a single repository for all the images in the dataset. Unlike the common adopted approaches of using single image file for each word sample, the format presented in this work is more appropriate for addressing large data files with large number of samples and it is more scalable as it facilitates moving datasets around different execution environments i.e. cloud based environments, it also facilitates the processing of image data in terms of loading, preprocessing and training.

تصنيف	وفي
التي	عام
كانت	حتى

Figure (3.5): Images from our dataset

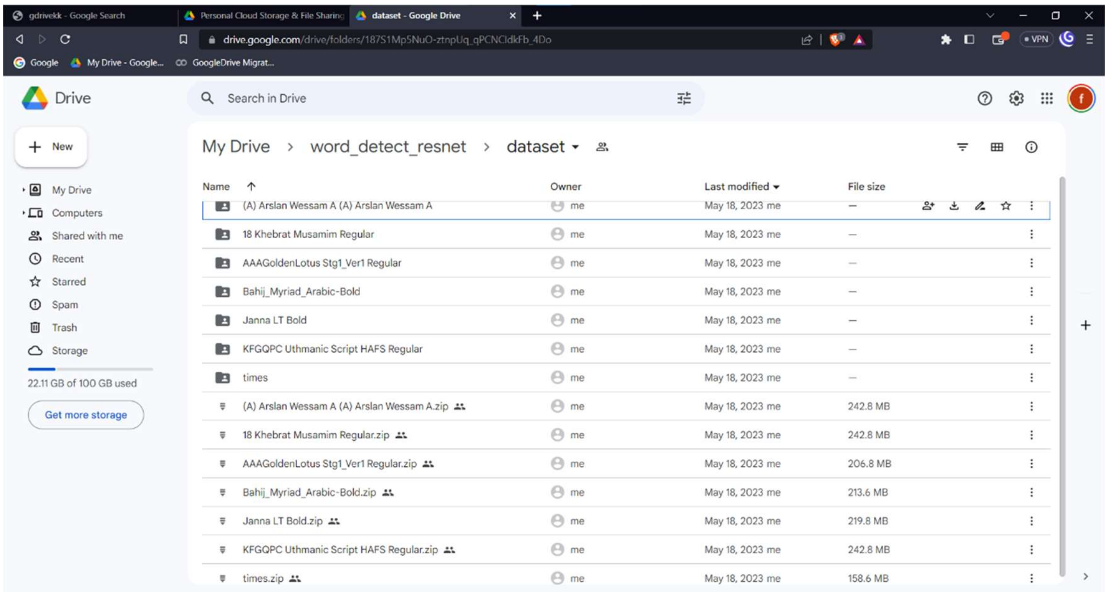


Figure (3.6): Image of dataset in google drive

3.4 Requirement Analysis :

Requirement Analysis is critical to the success or failure of a systems or software project.

3.4.1 Hardware Requirements:

- High Capabilities Computer.
- Scanner Device.

3.4.2 Software Requirements:

- Windows 7/8/10 (32 or 64 bit) operating system.
- Jupyter notebook or other python IDE.
- Python V.3 program or other new version.
- Pytorch and ternserflow
- Google Coleb
- PyCharm.
- Visual studio Code.

3.4.3 Functional Requirements:

Functional requirements are the requirements or functions a software or systems must perform. These are the functional requirements for our proposed system.

- The system must process images that the user enters with extension (JPG, PNG.).
- The system should display an error message when the user inserts an image whose extension is not in the required format.
- The system should allow users to copy extracted text from image.
- The system should allow users to upload images to perform Detection and recognition.

3.4.4 Nonfunctional Requirements:

Non-Functional Requirements are the requirements which ensure the usability and effectiveness of the entire software system.

- **Performance**: This system will be able to recognize our model with fast recognition performance.
- **Functionality**: This system will depend on the functional requirements mentioned in this document.
- **Availability**: this system will recognize the image of the input text if it is written in Arabic language.
- **Flexibility**: This system helps the user to recognize text easily.

- **Reliability**: This software will work reliably for low resolution images and for colored images.
- **UI :** UI page should be simple and user friendly.
- The system should display image to the users when the users clicked upload button and extract the text from image and displayed on GUI.

3.5 Proposed Approach :

3.5.1 Image Pre-processing :

It is the performing of different operation on the input image. It helps to remove noise from image, make character clear and it basically enhances the image rendering suitable for Training. Preprocessing has various task are such as converting gray scale, binarization, and normalization, Encoding Categorical.

3.5.1.1. Grayscale conversion

In this step we used Opencv to converting an image from other color spaces e.g. RGB, CMYK, HSV, etc. to shades of gray. It varies between complete black and complete white. Furthermore we must use this step to apply Dimension reduction For example, In RGB images there are three color channels and three dimensions while grayscale images are single-dimensional.

3.5.1.2. Binarization

In this step we are converting a grayscale image into binary (bi-level). And in this step we used an Adaptive thresholding method and Otsu method to segmentation the object from image by representation the value of background pixels with 1 for white and 0 for black.

3.5.1.3. Normalization

The normalization process is one of the most important tasks in our approach. In this step we tried to reducing the variations between pixels in the images of the text and adjust the size of the text images to identical size by divided each pixel with 255.

3.6 Text Detection :

Before the text recognition process, the text in the image must be identified and its location, either each row must be identified separately or each word selected, detection can be done in two ways, either by training a CNN model such as CRAFT, or by image processing operations traditional

3.6.1 CRAFT

(Character Region Awareness For Text detection) is used as detector to detection the Arabic text in an images. CRAFT is a deep neural network that uses the convolutional neural network model to calculate :

Region Scores: is used to localize word regions, and the inference is done by providing text-level bounding boxes as illustrate in figure(3.7).

Affinity Scores: is used to group the words into text.

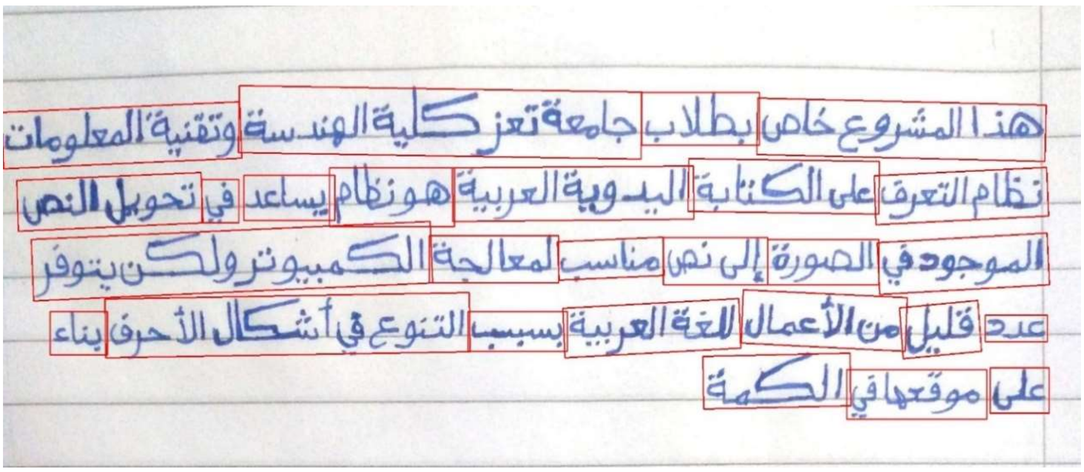


Figure (3.7): The Original outputs of Text detection model

By observed figure (3.7) The original outputs of CRAFT text detection model at various scales, so need to merge some of these boxes to reduce the processing , so the predefined parameter in **detect** method works with others to perform this operation , boxes with different level should not be merged, boxes with very different text size should not be merged, now by observe the outputs of this method we will find horizontal list of rectangle text boxes. The format is [[x1,y1],[x2,y2],[x3,y3],[x4,y4]] as illustrate in figure(3.8).

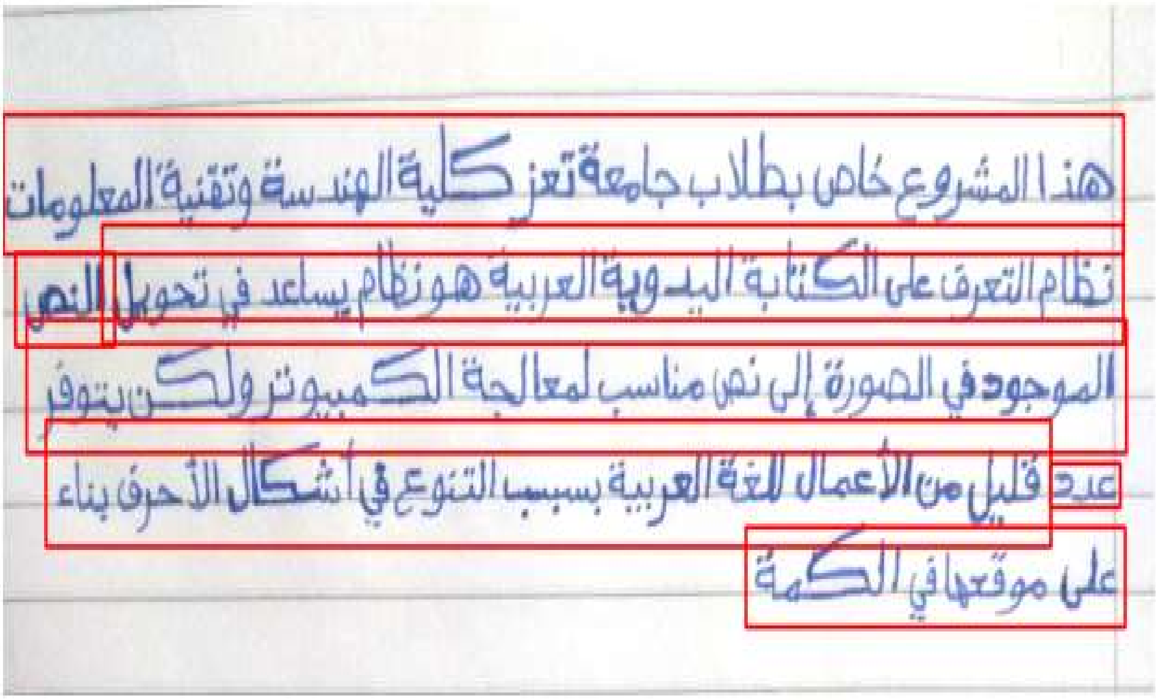


Figure (3.8) :Outputs of text boxes.

At last by crop the horizontal list of text boxes as in figure(3.9), for several separated images to pass after that to recognizer and the model of recognizer deal with each one of these separated images as a single input to predict it and return the appropriate text for that text written on image.

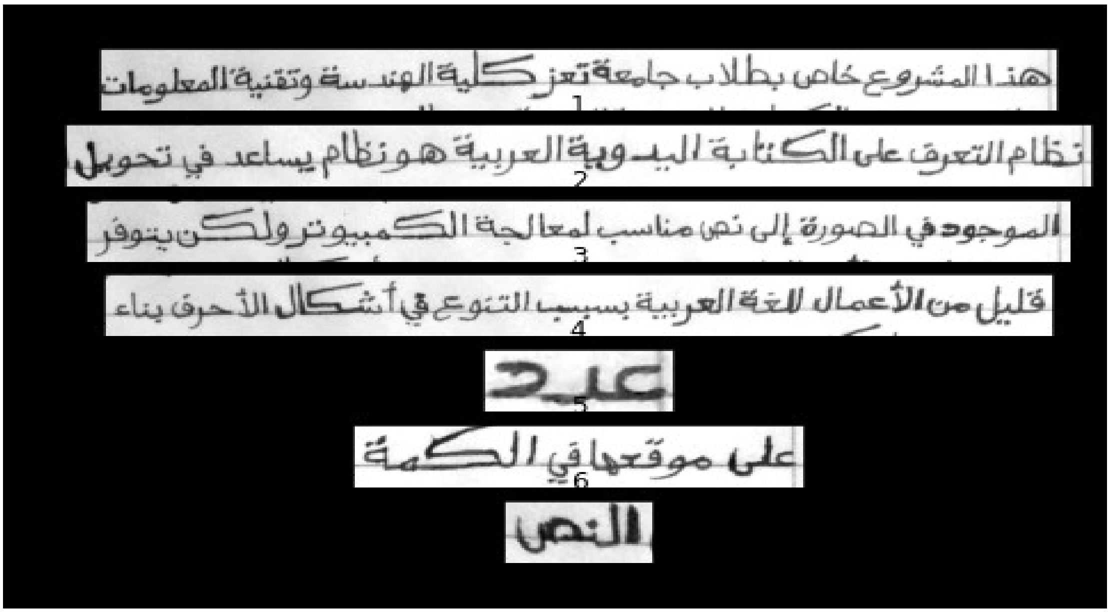


Figure (3.9): Crop the horizontal list of text boxes

3.6.2 Text Detection by using image processing (Segmentation) :

Segmentation is one of the most important steps in any handwriting recognition system, there are several types of segmentation:

- **Segmentation of a page into lines:**

The page consists of several lines, there are certain methods used for Segmentation of a page into lines such as Method based on the horizontal projection and the classification of the lines according to their heights or Method based on cutting by a sliding column of a fixed size. In this study we are using horizontal projection and line classification techniques, First, the paragraph image is preprocessed to remove any noise and enhance the contrast of the text. Then, horizontal projection is used to identify the positions of each line of text in the image. Horizontal projection involves summing the pixel values of each row of the image, which results in a graph where the peaks correspond to the lines of text in the image, in the following steps, we will explain all the operations necessary to divide the image into lines:

- **Image acquisition**

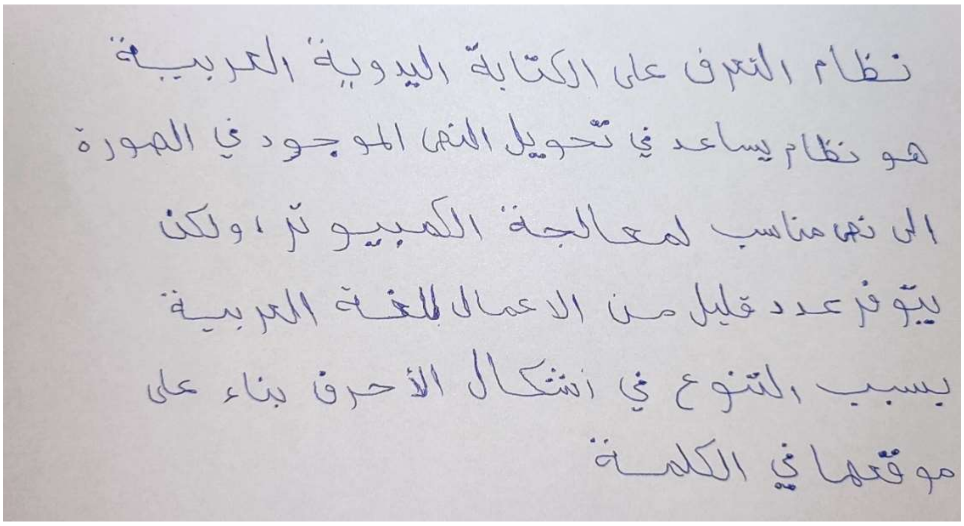


Figure (3.10): Original Image

- **Image thresholding(Binarization):**

We need a way to automatically determine the threshold, Due to the difference in the quality and lighting of the input images, The Otsu algorithm is used to automatically determine the threshold value that separates the foreground from the background. It calculates an optimal threshold value based on the image histogram, which is a plot of the distribution of pixel intensities in the image.

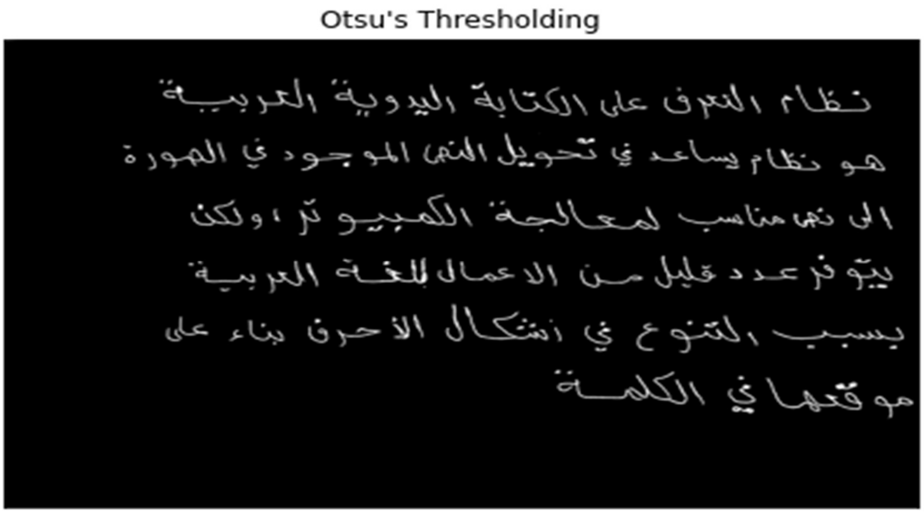


Figure (3.11) : Otsu's Thresholding

- ### **Image horizontal projection:**

Summing all pixels in each line, to detect the location of a horizontal object in image, by summing the pixel values of each line, we can also calculate the total brightness or intensity of an image

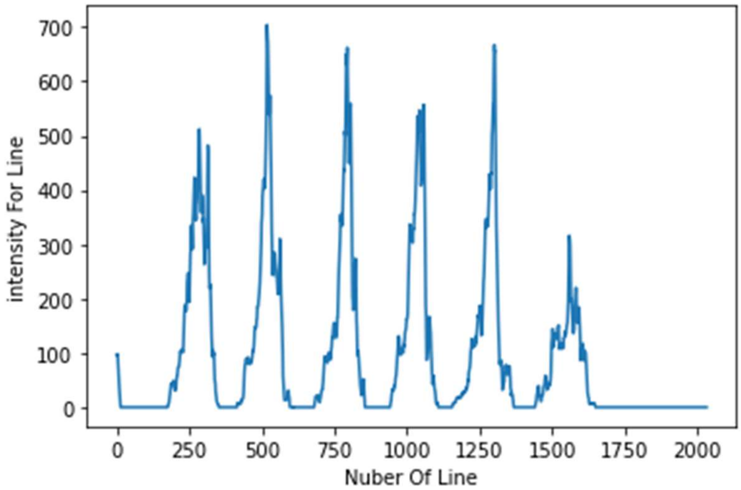


Figure (3.12): Image horizontal projection

- ### **Smooth image:**

Applying the moving average filter can be used to smooth out the pixel values in an image and reduce image noise. This improve the image quality and make it easier to detect features or edges in the image.

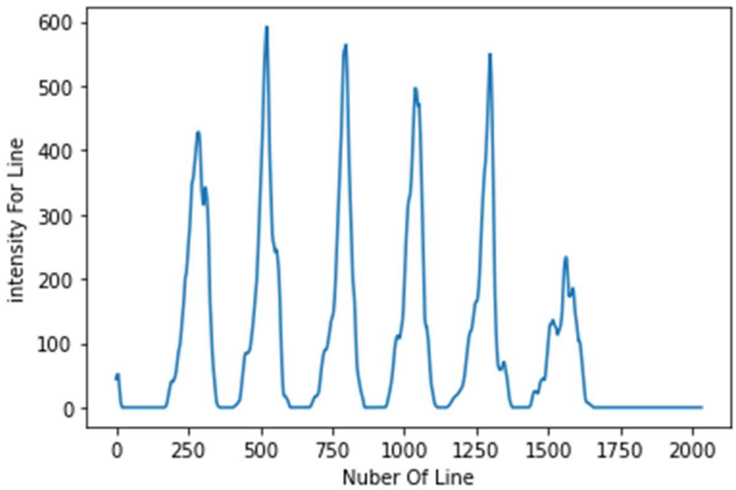


Figure (3.13): Smooth image

- **Resampling and peak detection**:

Is the process of changing the resolution of an image, Up-sampling increases the resolution of an image by adding more pixels, The goal of resampling is to change the size of the image or to adjust the pixel density .

Given an intensity histogram of an image, this function increases the resolution of the histogram (increases number of row), by interpolation and then finds the sharp peaks in this histogram, this values represent the peaks of histogram [28400 28700 52200 79700 103800 129700 156000]

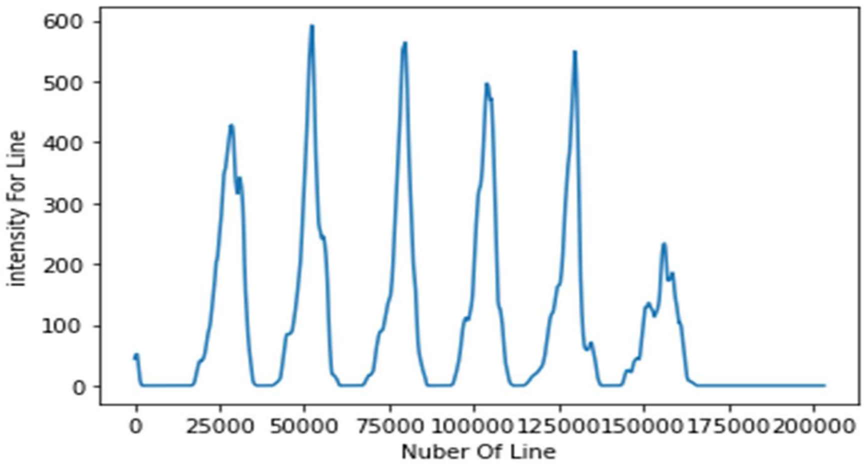


Figure (3.14): Resampling and peak detection

- **Calculate Gradients:**

The gradient represents the rate of change of pixel values in the image, and is high in regions where there is a rapid change in pixel values such as edges and boundaries, By computing the gradient at each pixel in the image, we can identify the location and direction of edges in the image Since we counted all the pixels for each row and then put them as a single value inside an array, we will use the gradient to define the edges of each row. If the gradient output is greater than 0, the value is 1, and if it is less than 0, the value is 0.

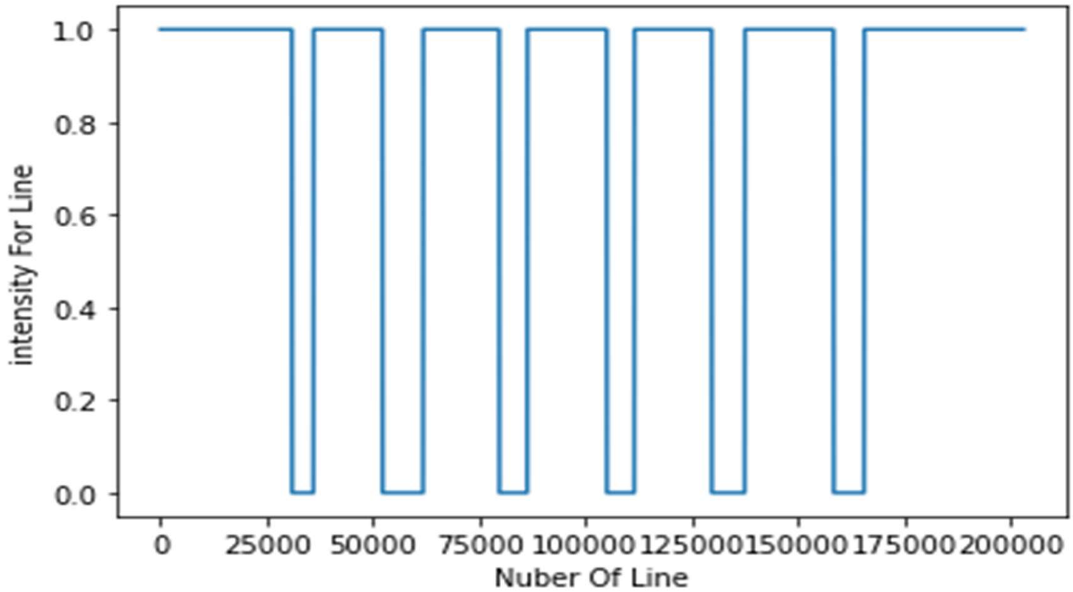


Figure (3.15): Numbers of line In image

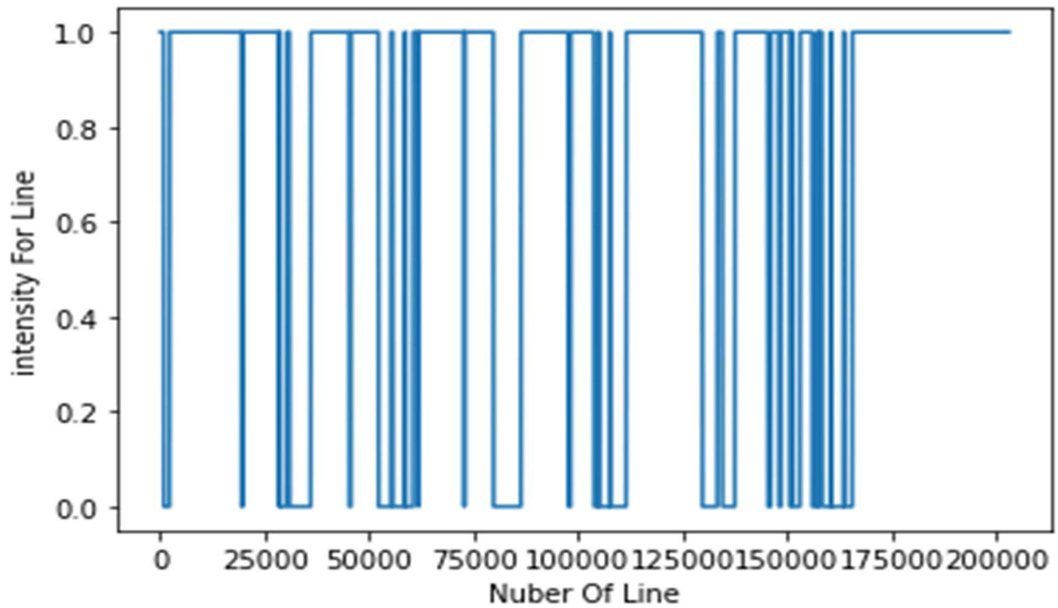
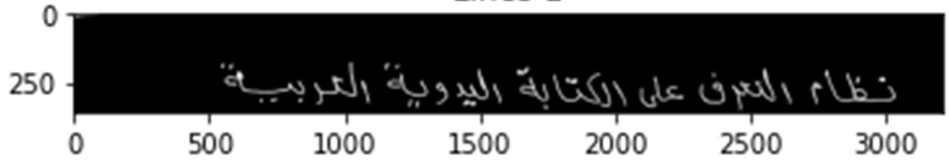
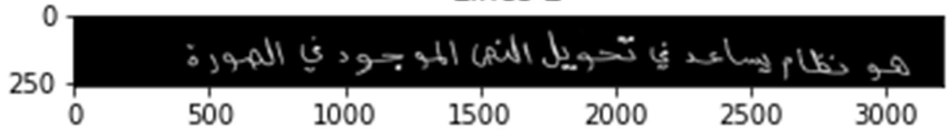


Figure (3.16): After Applying Gradients

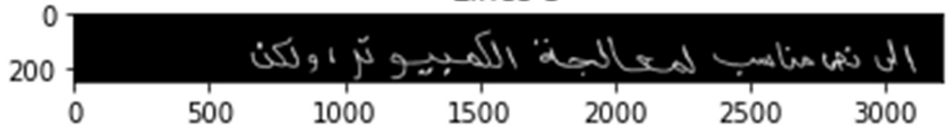
Lines 1



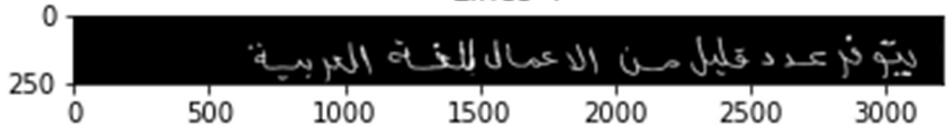
Lines 2



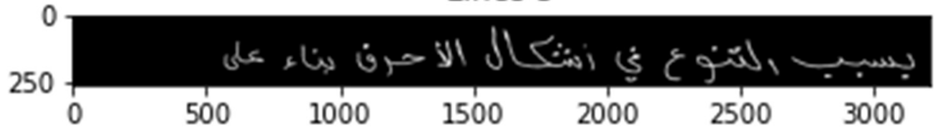
Lines 3



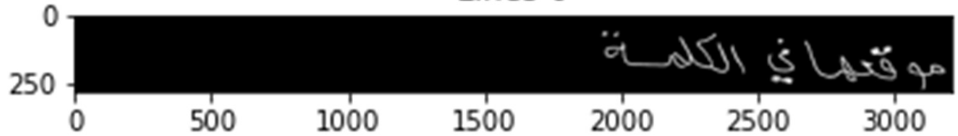
Lines 4



Lines 5



Lines 6



▪ Segmentation of a lines into words:

In this step we used the algorithm that called word extraction. The algorithm used in the input image is a text image that has already segmented into lines ,we have to browse all the lines of text where each line is segmented into words by applying multiple processing, the first step we browse the image lines for example we browsed the image line_1 as illustrate in figure (3.17). Then Compute the vertical projection histogram and get the indexes of zeros in this histogram. After this step we are segmenting the lines into connected components in order to compute the distances between these components, we compute threshold of the zeros sequences whether it is big enough to consider it as an inter word spacing or small enough to consider it as an inter word spacing by using the average of all these distances to classify these distances according to this threshold. The result of the algorithm is the number of columns of the between-word spaces, in the figure (3.18) we present the image of the result.

- **Image line 1 browsing :**

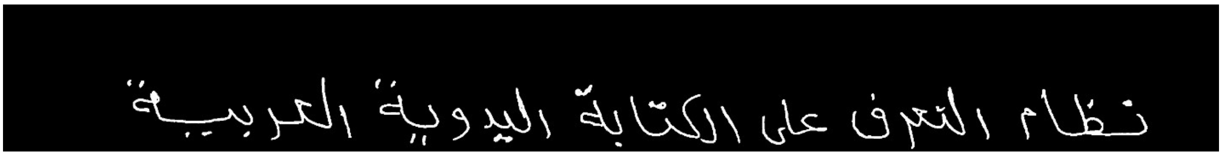


Figure (3.17): Line_1

Figure (3.18) : Result of the word extraction

- **Image line 1 vertical projection:**

Summing all pixels in each column, to detect the location of a vertical object in image, by summing the pixel values of each column, we can also calculate the total brightness or intensity of an image.

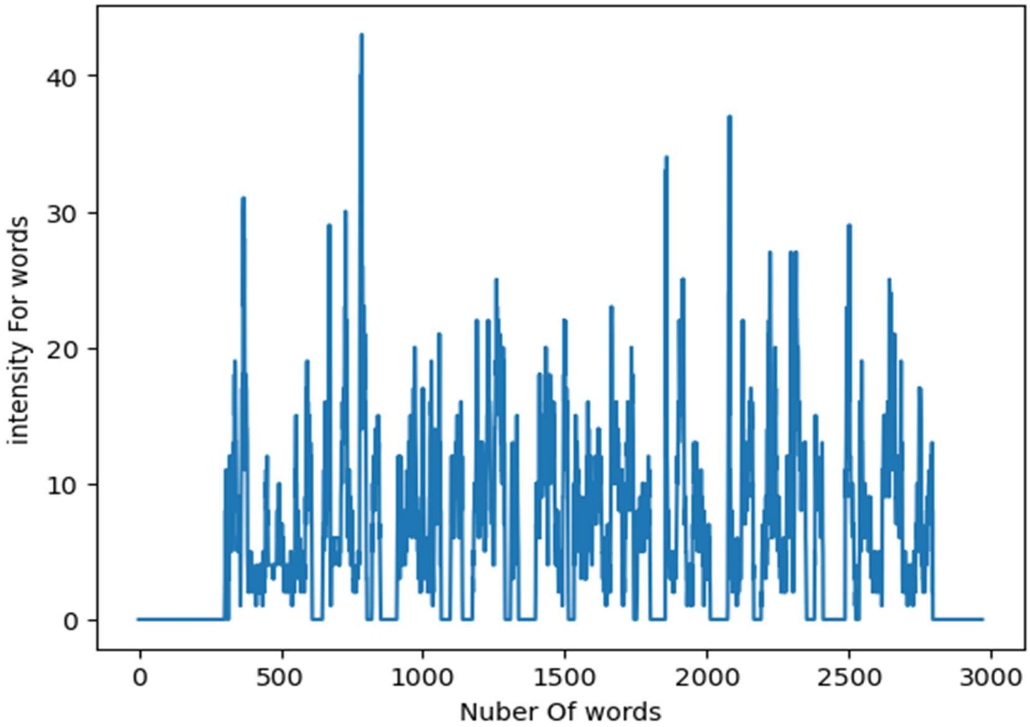


Figure (3.19: Image line 1 vertical projection

- **Smooth image line 1:**

Applying the moving average filter can be used to smooth out the pixel values in an image line 1 and reduce image noise.

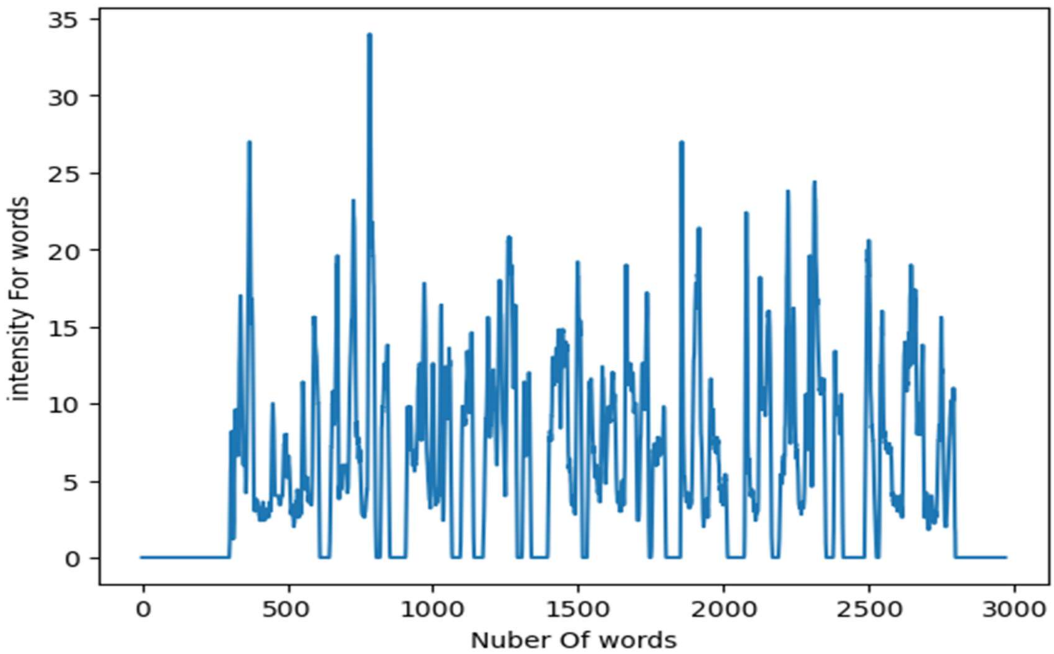


Figure (3.20): Smooth image line 1 vertical projection

- **Segmentation image line 1 into connected components:**

  The algorithm used in the processed image by vertical projection and smoothing. In this step each line is segmented into connected components in order to determine the distances between these connected components. The algorithm applied to extract words that finding in the image line as we illustrated this operation in figure (3.21):

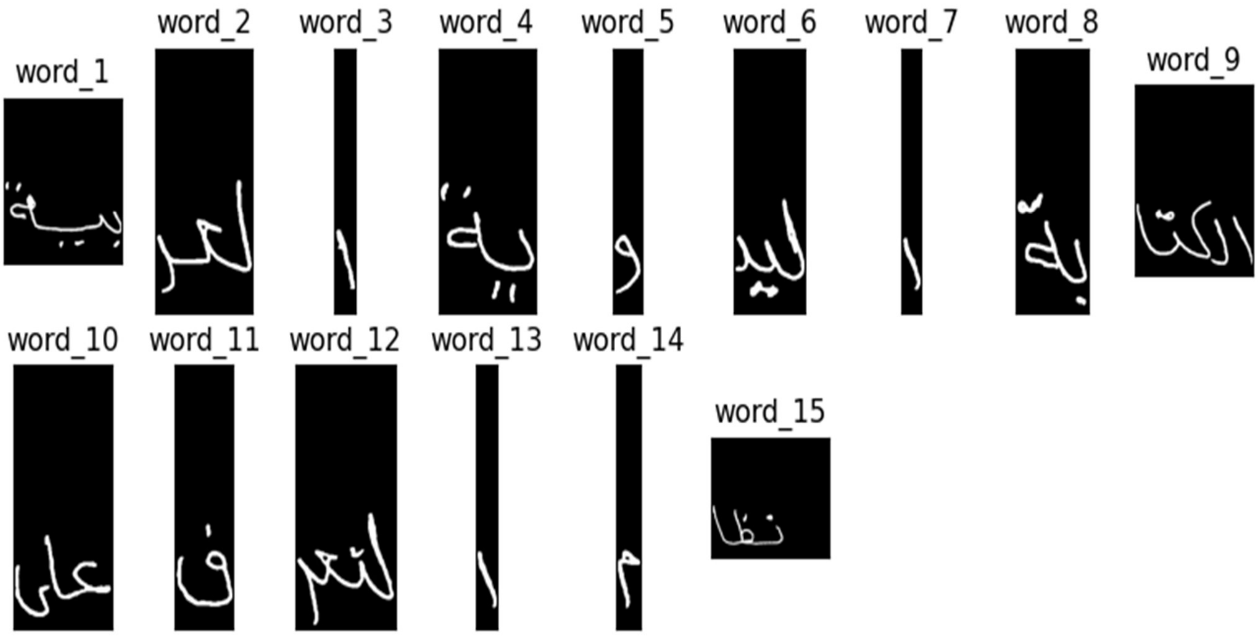


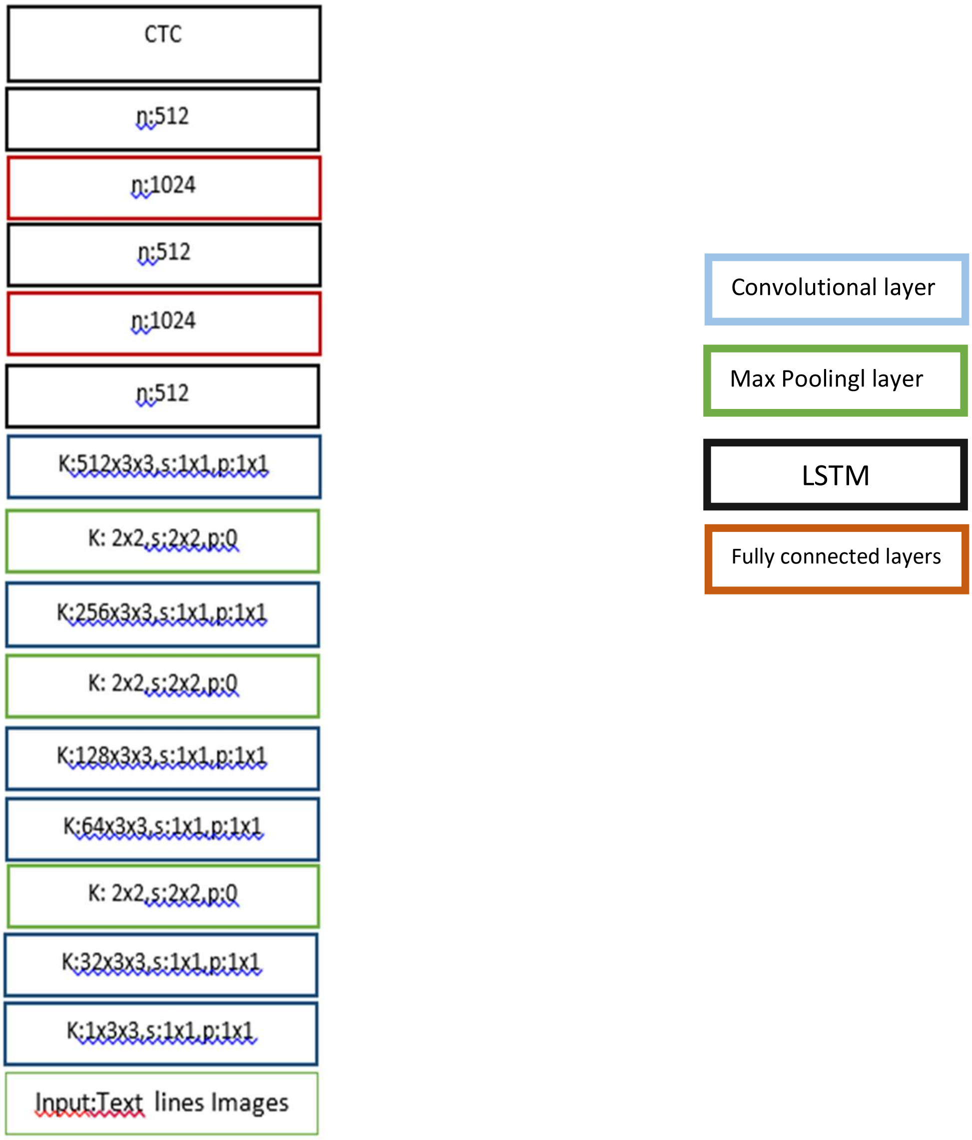
Figure (3.21) : Connected component word in image_line_1

- **Add the segmented words to our model to predict:** After finishing from applying the algorithms to extraction each words in the line , we then send the segmented image to the ResNet model .

3.7 Text Recognition:

- ### ResNet :

The ResNet architecture is considered to be among the most popular Convolutional Neural Network architectures around. Introduced by Microsoft Research in 2015. The ResNet Model for specific our project is a model which has 7 Convolution layers with 3 MaxPool and 1 AdaptiveAvgPool layer, also it has a Sequential modeling in BidirectionalLSTM with Linear and LSTM (long short term memory). It is a widely used with our project, to know more about it see the table.



Layer (type (var_name))	**Param #**
Model (Model)	--
├─ResNet_FeatureExtractor (FeatureExtraction)	--
│ └─ResNet (ConvNet)	--
│ │ └─Conv2d (conv0_1)	288
│ │ └─BatchNorm2d (bn0_1)	64
│ │ └─Conv2d (conv0_2)	18,432
│ │ └─BatchNorm2d (bn0_2)	128
│ │ └─ReLU (relu)	--
│ │ └─MaxPool2d (maxpool1)	--
│ │ └─Sequential (layer1)	230,144
│ │ └─Conv2d (conv1)	147,456
│ │ └─BatchNorm2d (bn1)	256
│ │ └─MaxPool2d (maxpool2)	--
│ │ └─Sequential (layer2)	2,099,712
│ │ └─Conv2d (conv2)	589,824
│ │ └─BatchNorm2d (bn2)	512
│ │ └─MaxPool2d (maxpool3)	--
│ │ └─Sequential (layer3)	22,555,648
│ │ └─Conv2d (conv3)	2,359,296
│ │ └─BatchNorm2d (bn3)	1,024
│ │ └─Sequential (layer4)	14,161,920
│ │ └─Conv2d (conv4_1)	1,048,576
│ │ └─BatchNorm2d (bn4_1)	1,024
│ │ └─Conv2d (conv4_2)	1,048,576
│ │ └─BatchNorm2d (bn4_2)	1,024
├─AdaptiveAvgPool2d (AdaptiveAvgPool)	--
├─Sequential (SequenceModeling)	--
│ └─BidirectionalLSTM (0)	--
│ │ └─LSTM (rnn)	--
│ │ └─Linear (linear)	--
│ └─BidirectionalLSTM (1)	--
│ │ └─LSTM (rnn)	--
│ │ └─Linear (linear)	--
├─Linear (Prediction)	--
│ └─LinearPackedParams (_packed_params)	--

Total params: 44,263,904

Trainable params: 44,263,904

Non-trainable params: 0

▪ LSTM :

After using a ResNet CNN architecture for features are then fed into a sequence modeling module, which consists of two bidirectional LSTMs. LSTMs are a type of

RNN that are designed to handle sequential text with time-series data by capturing long-term dependencies text in the input. The bidirectional nature of the LSTMs allows the model to take into account both past and future context when predicting the next character in the sequence.

Then the output of the sequence modeling module is fed into a linear layer to predict the final output, which in this case is the text contained in the image. The model is trained end-to-end using backpropagation to minimize a loss function that measures the difference between the predicted text and the ground truth text. The model has a total of 44,263,904 parameters, all of which are trainable.

- **CTC :**

CTC (Connectionist Temporal Classification) is a decoding algorithm used in AHWS to improve text recognition . that uses CTC as one of its main components for sequence labeling and decoding. It is built on top of PyTorch, ResNet, CTC, and beam-search-based decoding .

3.7.1 Gready-Decoder :

- A greedy search decoder is a decoding algorithm for natural language processing tasks that involve generating sequences of words, such as caption generation, text summarization, and machine translation.
- The use of a greedy search decoder is to select the most likely word at each step in the output sequence based on the probability distribution output by a model.
- The method of work of a greedy search decoder is to iterate over the output sequence length and choose the word with the highest probability as the next word in the sequence.
- The advantages of a greedy search decoder are that it is very fast and simple to implement.

3.7.2 Hyperparameter Tuning:

Optimizers: Optimizers are algorithms or methods used to minimize an error function (loss function) or to maximize the efficiency of production. Optimizers are mathematical functions which are dependent on model's learnable i.e Weights & Biases. Optimizers help to know how to change weights and learning rate of neural network to reduce the losses. The optimizer used in our model we built is Adadelta .

Adadelta is an extension of Adagrad and it also tries to reduce Adagrad's aggressive, monotonically reducing the learning rate and remove decaying learning rate problem.

Loss function: The CTC loss function is typically used in conjunction with an optimizer, such as stochastic gradient descent (SGD) or Adam, to minimize the loss during training. By minimizing the CTC loss, the model learns to align the predicted output sequence with the target sequence and improve its accuracy in sequence labeling tasks.

Batch and Epoch : we have used a 32 batch and an 100 epoch .

3.7.3 Post-Processing :

We uses post-processing techniques to refine our HWR model output. This includes spell-checking, punctuation correction, and word dehyphenation to improve the result's quality.

- Detects bounding boxes and polygons by applying a combination of morphological and probabilistic techniques to the score and link maps, based on text and link threshold values.
- Adjusts the coordinates of bounding boxes and polygons (scaling and coordinate mapping) to match the original image scale.
- Returns the final list of bounding boxes and polygons coordinates of the detected texts.

- test the input image with the trained model, which returns a pair of boxes (word in boxes)
- Converts the polygon data and converts it into the desired format: list of lists containing polygon coordinates.
- Appends these polygons to a results list.
- Returns the final list as output.

3.8 Risk and Hardship:

The variability of Arabic handwriting is a significant risk to the accuracy of the recognition system. The Arabic script is cursive, and its letters can connect to each other in different ways depending on their position within a word. Arabic handwriting can also vary significantly from one writer to another, and even from one sample to another for the same writer. This variability poses a significant risk to the accuracy of the recognition system.

Another significant challenge in developing AHWRS is the limited availability of annotated data. Unlike printed Arabic text, which has been extensively studied and annotated, there is a shortage of annotated data for Arabic handwriting. This makes it challenging to train and test the recognition system.

Need a lot of computational processes and a clear handwritten and a bright background without noise.

3.9 Summary :

In this chapter we have explained the structure of our system and the system requirement, non-requirement, software requirement and then we explain the varies datasets we used in building our system then we explained our system approach.

CHAPTER 4: SYSTEM DEISGN AND TESTING

4.1 Overview:

In this section, we will study the flowchart of proposed system, and our system interface, Use Case Diagram, User model, preprocessing model, Segmentation model, Feature extraction model, Recognition model, and Sequences diagram of our model, and Activity diagram, and the implementation of study and the report and measurements results of our model and the performance of our model.

4.2 Proposed Method:

The proposed system takes the text image from the GUI that Upload by the user, then the system deals with as the system inputs to extract the handwritten text from it, and shows the recognized text as system outputs. Using this method, the uploaded image, that need to read the text from it. It need some steps to easy deals with it such as preprocessing to resized in a global size and applying some filters to reduce noise and deals with the intensity, contrast. Then by using detector model for detection text in the image by identify bounding boxes around each word or instance to perform text segmentation by crop the whole bounding boxes in the image and local save its as a list of several cropped image to send after that to the next stage. the recognize stage will receive the cropped image and then start deal with individually cropped images to enable the recognizer model to examine text segmented and recognition it and return the appropriate text, eventually the text that returned from the system should present on the user's screen.

4.3 Use case diagram:

Is a tool used in software development to illustrate a system's components, users and how they interact for given situations. For HWRS, it can be used to illustrate how the system accesses an image, processes it, and produces an output.

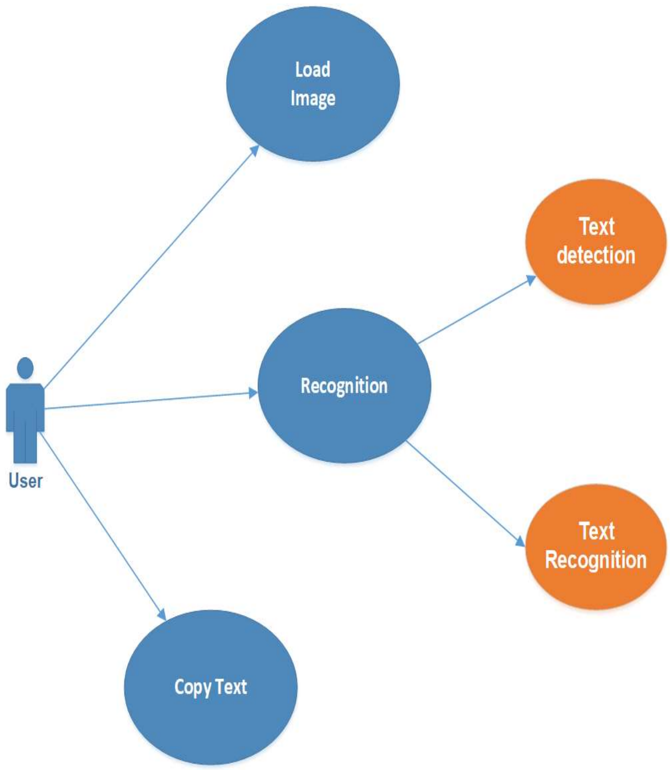


Figure (4.1): Use case diagram (User)

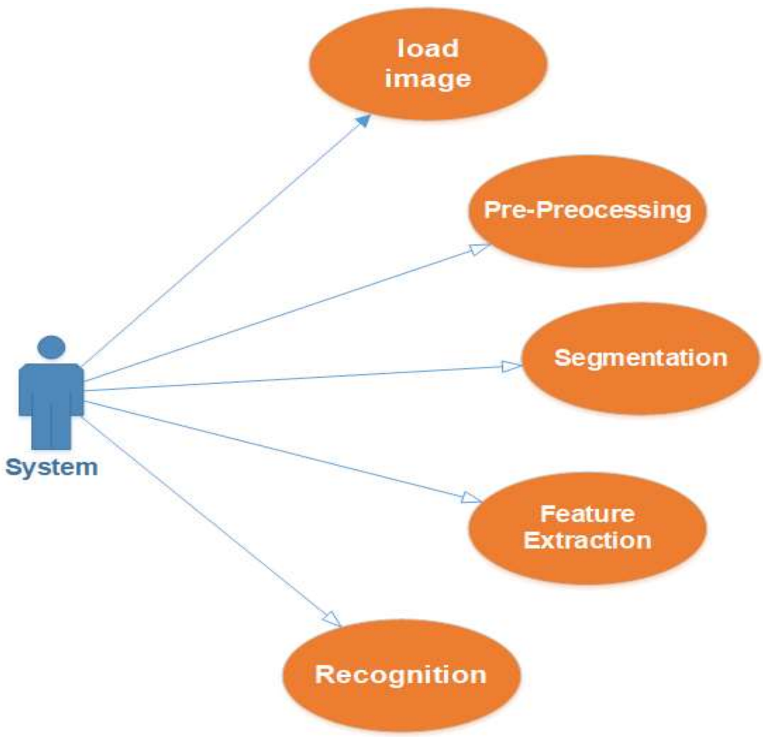


Figure (4.2): Use case diagram (system)

4.3.1 User model:

It includes information about the user, including the font style or size of the text, the image background, and any distortions present. This model is used to create a profile or specification of the user's needs and preferences, which is used in the preprocessing model's selection.

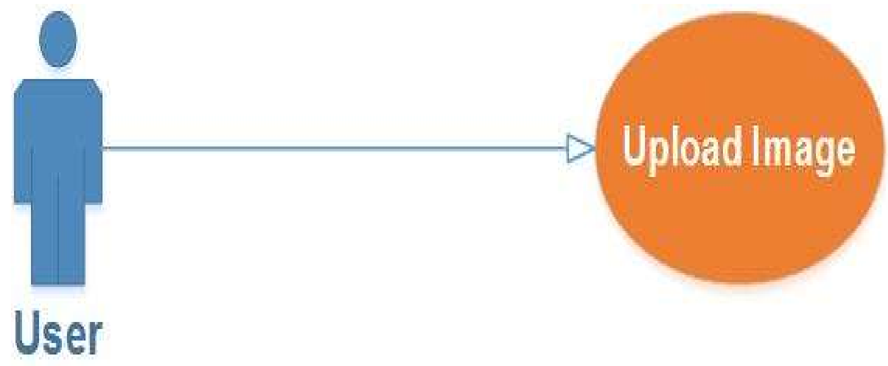


Figure (4.3): Use case for user model

Use Case	Description
Actor	User
Condition	Image should be available and clear white background
Scenarios	User upload image ,
Action	Image uploaded successfully

4.3.2 Preprocessing model:

It is used to clean up the image and enhance its quality before proceeding further. This model performs operations like contrast enhancement, noise removal, skew correction, and binarization.

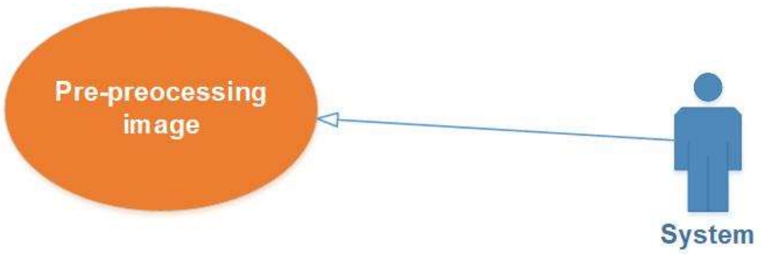


Figure (4.4): Pre-Processing module

Use case	Description
Actor	System
Condition	Upload input image
Scenario	The image is converted from RGB to binary representation during pre-processing.
Action	Extraction character before segmentation

4.3.3 Segmentation model:

It divides an image into several segments or regions that include text and non-text components. This model is used to identify and isolate textual content from an image.

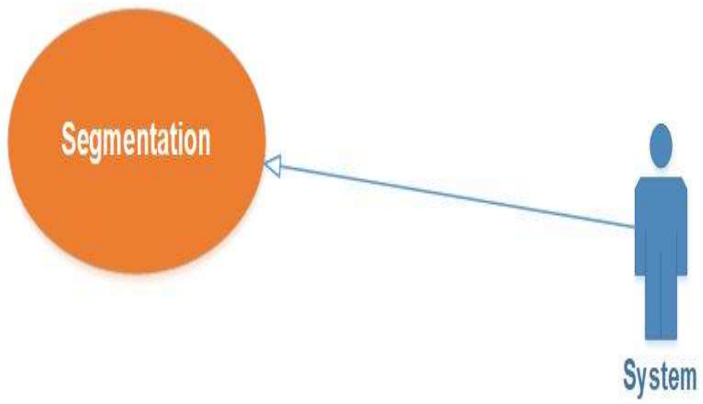


Figure (4.5): User case for segmentation module

Use case	Description
Actor	System
Condition	Pre-processing image should be available
Scenarios	The Pre-Processing input image is segmented into isolated word by assign a number to each word using a labeling process, this labeling provides information about number of word in the image
Action	Image segmented successfully

4.3.4 Feature extraction model:

It involves identifying critical features of the text such as lines, curves and edges. The extracted features are then used to transform the text into machine-interpretable data that may be processed in the recognition phase.

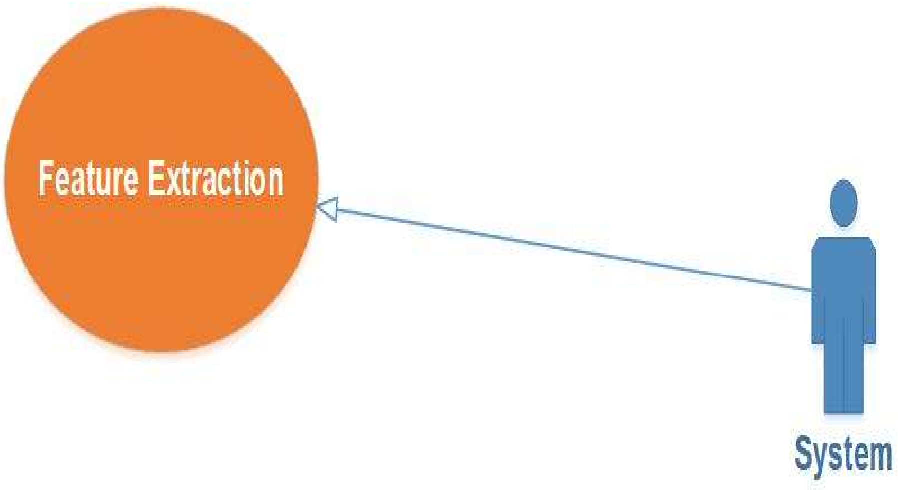


Figure (4.6): Feature extraction system model

Use case	Description
Actor	System
Condition	Segmentation input image should be available
Scenarios	We use the freeman ResNet after obtaining the image's word. The coordinates of the boundary pixels are obtained first, based on these coordinates the coordinates code of the character image is found. The coordinates code is converted to a two-dimensional matrix to produce the normalized code. The code value can be found in the first row of this matrix, and its frequency of occurrence can be found in the second row.
Action	word successfully extract-feature

4.3.5 Recognition model :

It is used to match the features and patterns extracted from the image to the available font styles and templates to output machine-encoded text. This model is built over Machine Learning algorithms or heuristic-rule-based approaches.

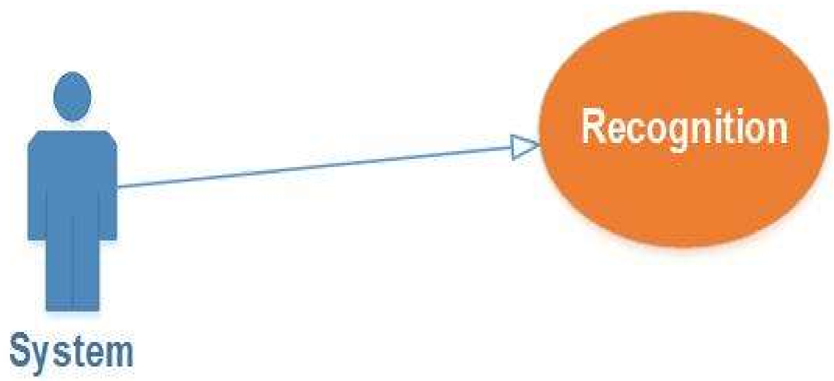


Figure (4.7): System use case for recognition module

Use case	Description
Actor	System
Condition	Feature extraction for character must be available
Scenarios	We used transfer learning using a ResNet model to train our own database of more than a million Arabic words
Action	Your text from image is recognized successfully

4.4 Sequential Diagram :

Sequence diagrams are sometimes called event-trace diagrams, event scenarios, and timing diagrams. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

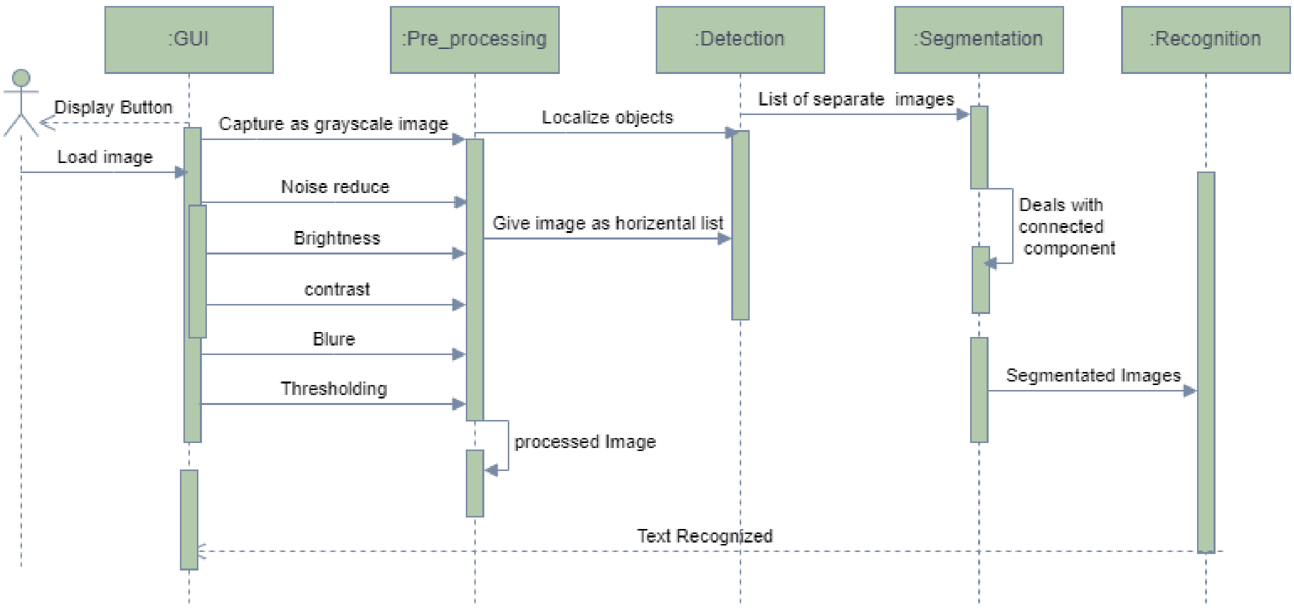


Figure (4.8): Sequential Diagram

4.5 Activity Diagram :

Activity diagram is the diagram that support the logical modeling of system processes an workflows. Activity diagrams are used to model the behavior in the system process independent of objects. An activity diagram is a way of describing system flow, with the possibility of expressing how actions are taken, what they do(change of object states), when they take place (action sequence), and where they take place(activity partitions). In the UML activity diagrams can be used to describe the business and operational step-by-step workflows of components in the system. An activity diagram shows the overall flow of control.

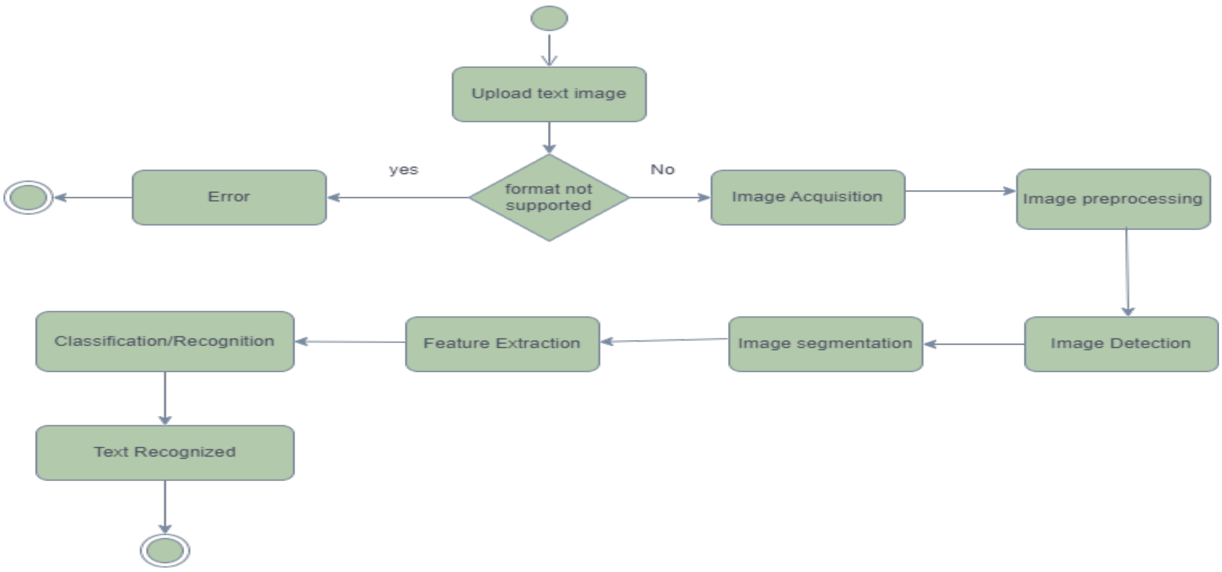


Figure (4.9): Activity Diagram

4.6 System Architecture :

The system architecture describes the major components with their relationships and how they interact with each other. The Architecture for our proposed system is given below.

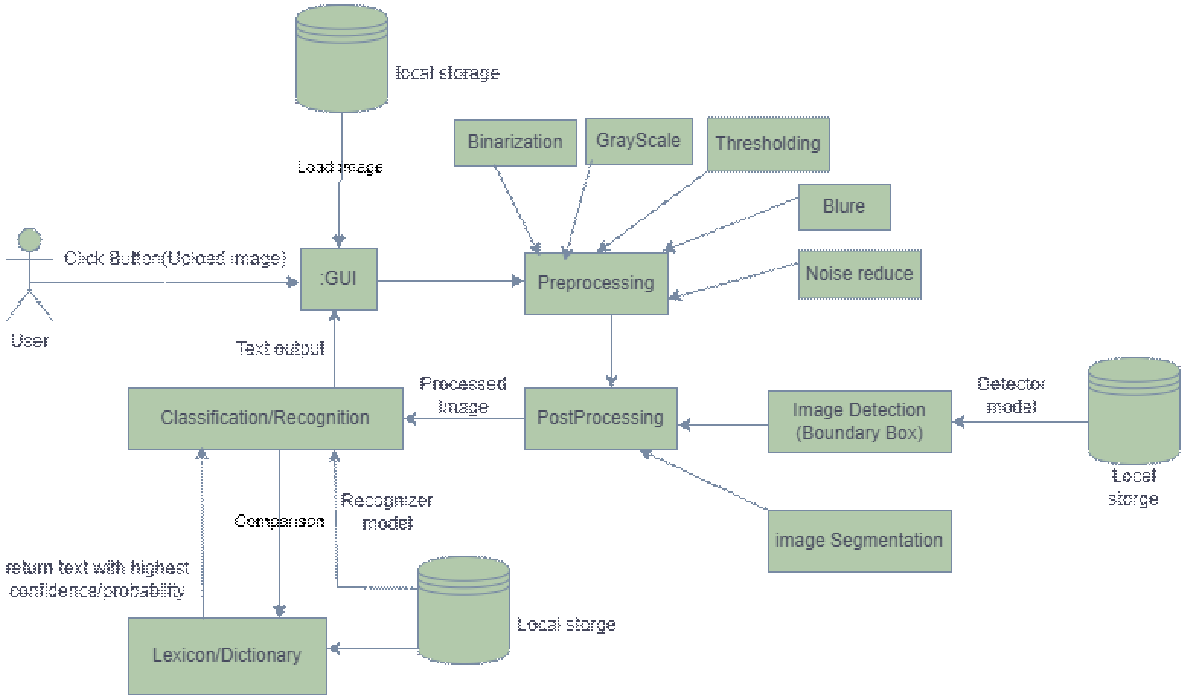


Figure (4.10): System Architecture

4.7 System Implementation :

A web page with an HTML, CSS form that allows users to upload an image. Use JavaScript to handle the form submission and send the image to the Flask backend. Flask Python Backend: a Flask server to receive HTTP requests from the web page, and Pass the image to your image recognition model for processing, finally return text.

1- Click on Icon to display a file upload dialog, and load image

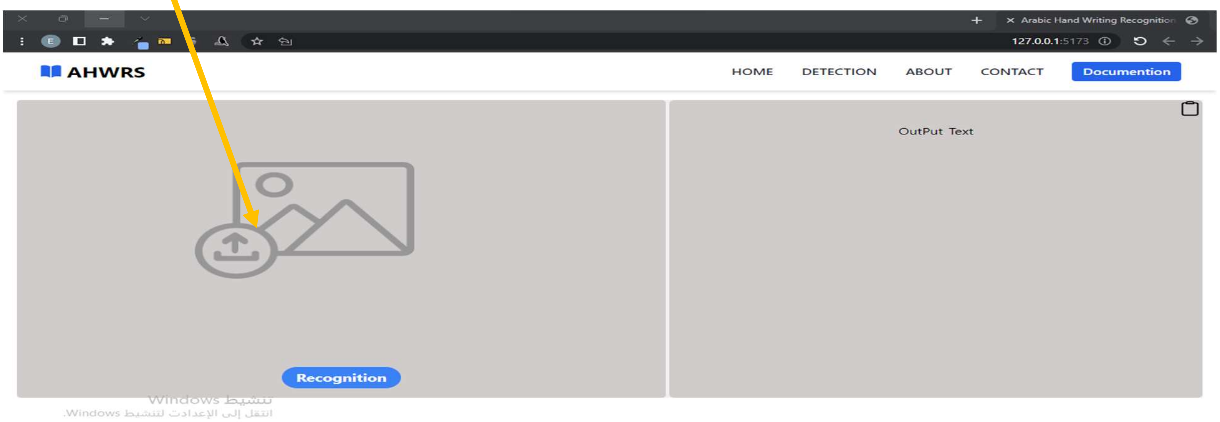
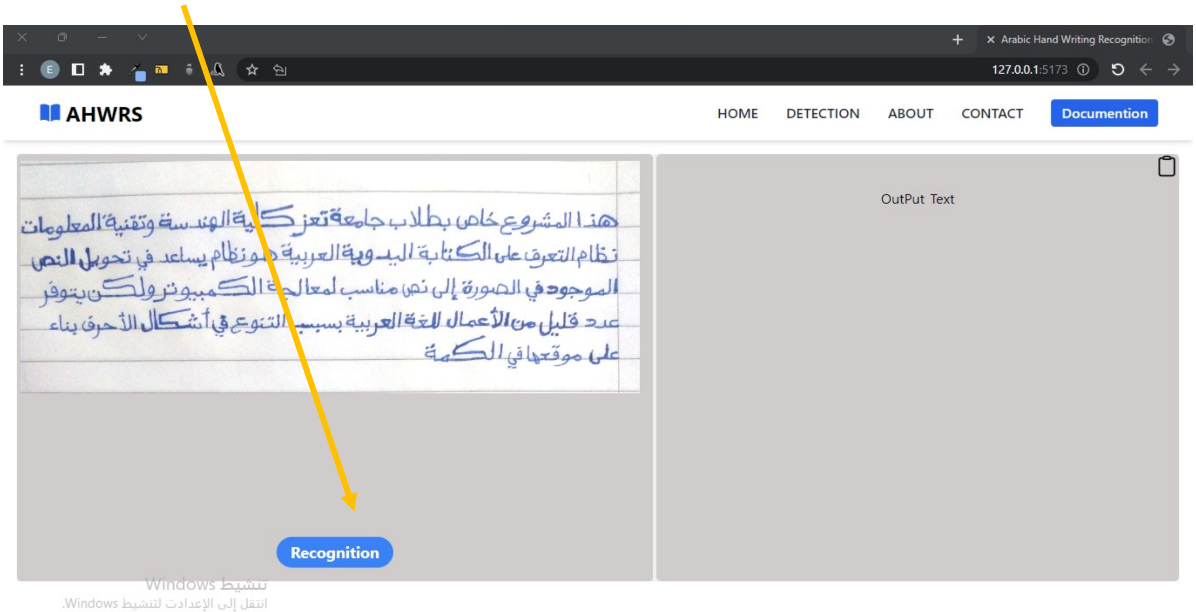
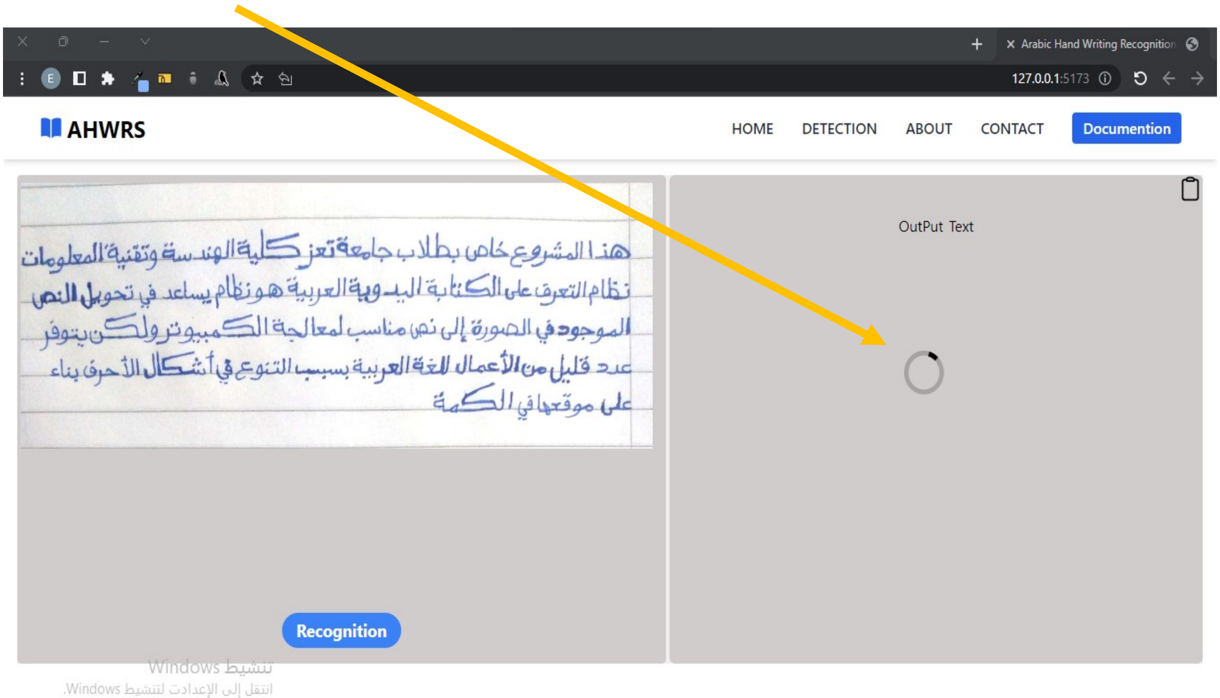


Figure (4.11): Main interface of AHWRS System

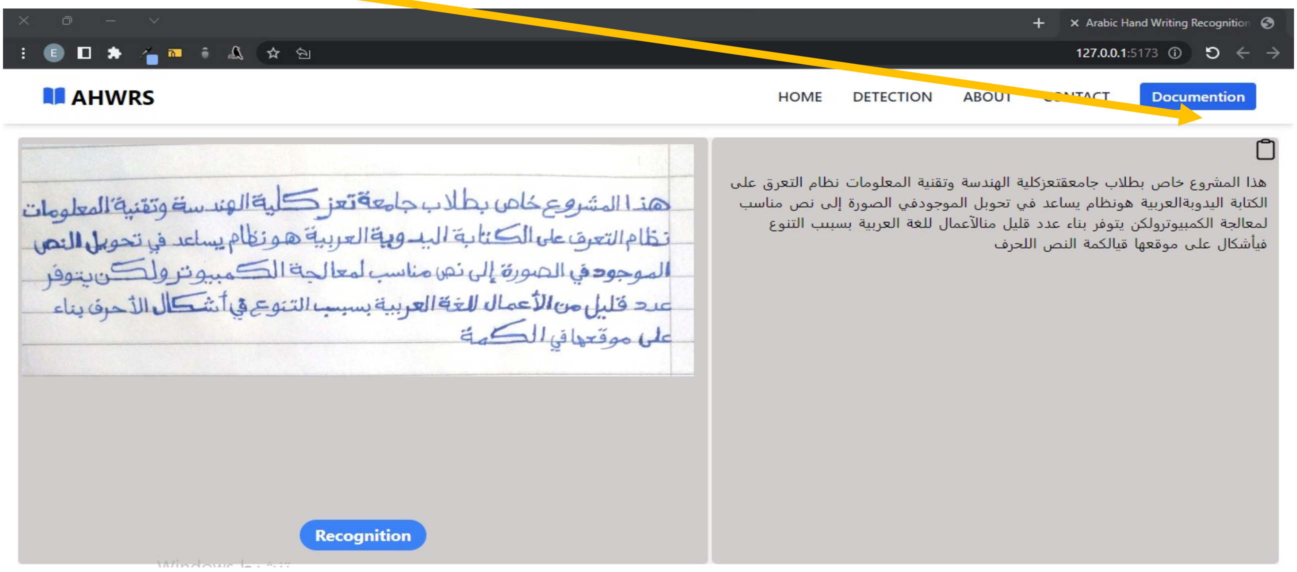
2- Click on Recognition Button To send image to Backend and show texet



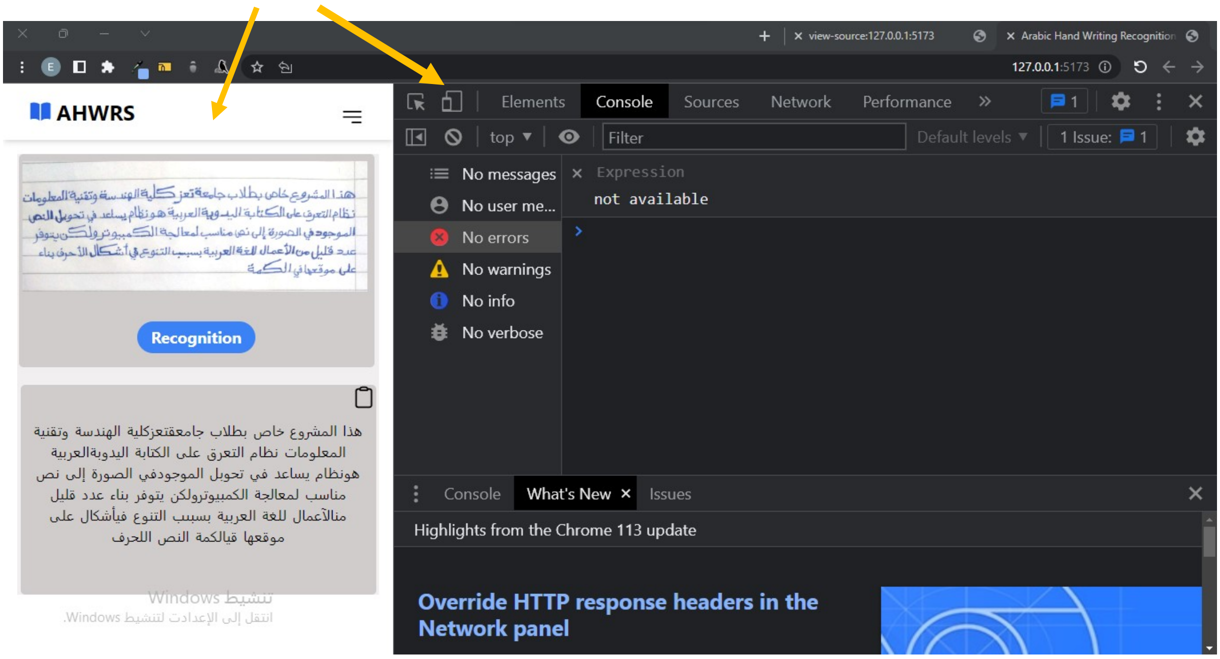
3- Wait to load model and recognize image , Waiting time depends on the size of the image and the number of words in the image



4- Click on this icon to copy text to the clipboard



5- This site is responsive and can be used on mobile phones, tablets or computers



CHAPTER 5 : CONCLUSION AND DISCUSSION

5.1. Overview :

In this chapter we view the results of the Arabic AHWS that are ready for assessment after the design and implementation phases are finished. The evaluation criterion for the majority of AHWS is based on the overall recognition percentage and future work and Discussion. In order to assess the effectiveness of our AHWS, the time that passed for each stage is typically calculated. The recognition rate of the various fonts is calculated in this chapter. The used fonts ((A) Arslan Wessam A (A) Arslan Wessam A, 18 Khebrat Musamim Regular, AAAGoldenLotus Stg1_Ver1 Regular, Bahij_Myriad_Arabic-Bold, Janna LT Bold, KFGQPC Uthmanic Script HAFS Regular, and Times New Roman) . The amount of time that has passed during the image pre-processing, feature-extraction, image-segmentation, post-processing,and classification stages is used to evaluate the system's performance and our system accuracy.

5.2. Result :

After training and evaluating the result of our model for the our dataset and validation accuracy, we can come to the conclusion that even thought that our model has reached the accuracy 95.3% ,it cannot be improved anymore in the current configuration.

- The performance of the AHWS system was trained on a our custom dataset seven font which we build and use for training our model , consisting of 1276654 simples per font with a total of 1276654 image and labels. The dataset was pre-processed by removing noise and skew correction.
- The AHWS system was trained using a combination of Convolutional Neural Networks (CNN) and Long Short-term Memory (LSTM). The trained model was able to recognize printed and handwritten text accurately, with an average accuracy of 95.3%. The system performed better on printed text compared to handwritten text .due to the noise and variation present in the handwriting.

- Builted website is use to upload image that contain Arabic text and send to backend Recognize it, Then show text that is editable and more useful in many different cases.
- The Arabic HWRS component of this model has been applied on the synthetic dataset by seven fonts ((A) Arslan Wessam A (A) Arslan Wessam A, 18 Khebrat Musamim Regular, AAAGoldenLotus Stg1_Ver1 Regular, Bahij_Myriad_Arabic-Bold, Janna LT Bold, KFGQPC Uthmanic Script HAFS Regular, and Times New Roman).

5.3. Discussion :

the discussion regarding our proposed system is presented as complex tasks, such as detecting and recognizing diacritical image texts at the image that contains text with its diacritical in Arabic HWR, that have not received much attention. This can be the current research direction in preparing Arabic dataset and present some propose methods about how dealing with problems in Arabic language. Furthermore to the best of our knowledge, there are no publicly available datasets for a diacritical line dataset or Quranic image dataset for text recognition purposes. our dataset deals with the images text, however it is not available to researchers. In addition, it is synthetically generated at word level. We worked on RNNs, CNNs because it is considered the state-of-the-art system in the Arabic text recognition domain. but many researchers use only CNNs in the Arabic domain for different task like speech recognition and in Arabic neural machine translation, Arabic named entity recognition and Arabic discretization. Meanwhile for complex tasks, such as recognizing diacritical image texts (Quranic text) at word or line level has not received much attention.

5.4. Conclusion :

The project was initiated with the aim of building platform for the users where they can perform Arabic handwriting recognition. The proposed project presents the Arabic handwriting recognition system where system extract the words from image in editable form and displayed it on GUI. Many components were integrated to build a complete system. First the dataset was generated and built the model and finally implemented the flask for the integration of front end and back end. It was done by using many technologies and algorithms such as Pytorch, CRAFT, NumPy for building the CNNs, RNNs, OpenCV for the preprocessing and postprocessing stages respectively, and HTML, CSS, React js for front end. The accuracy rate achieved by the model was **95.3%**. The entire system was built in agile methodology as it is very flexible and easily adaptable in any projects. The project was planned to breakdown the system into many of artefacts and allocated time for each task. Different methods were used for the project for gathering the requirements such as read papers research, Observation, Document review etc. The requirements gathered from this method were successfully built in the system. Research had been done on similar systems for finding the information on the tools and techniques used on their system. Finally, the system was built in a way that it can fulfill the aims and objectives and answer the academic questions such as the questions in the following.

1) **How will the system work**:

The proposed system works by specific method i.e. take the whole words as input and recognize it without detecting each character like in character recognition. Users upload the image from storage disk into the GUI, the backend will save the image into its own local storage. With that image, the preprocessor stage will grayscale and binarize it, and the detector create a temporary copy of the image for word detection. For every word detected, the detector will crop the words out and save it for classification. After the words detection is complete, the server will remove the temporary image. With the cropped word images, the classifier will create a list of word results. For each word image, the classifier will reshape the image and send it through its network for prediction. Each text classification will have a list of words and their resulting confidences. The classifier will extract the highest confidence and its respective word and append it to the word list. When all of words have been classified, the system will return the list of words to the GUI front to be displayed as a sentence of classified words.

2) **How will the users get benefited by this system**:

The system was built for the convenience of the users. Before the concept of OCR and HWR came, people were compelled to store their data and information in hard documents. In corporate field, they had to copy the whole documents while storing the information. But with the HWRS, people can store in the electronic from without going through the hassle of copying the whole information in the form of text. So, this proposed system will help them to store data by extracting words from images in editable form.

5.5. Future work:

1. An improved model for text detection:

The currently proposed system uses the CRAFT model for text detection. In the future, it will be useful to explore and try other state-of-the-art text detection models, such as EAST (Efficient and Accurate Scene Text Detector) or TextBoxes++ (One-shot Scene Text Detector).

2. Multilingual text recognition:

The system is currently focusing on Arabic text recognition. However, to make it more versatile and useful, it can be extended to support text recognition in multiple languages. This may involve training the model on datasets containing scripts from different languages and implementing language detection mechanisms to handle multilingual inputs.

3. Integration of user feedback:

Incorporating user feedback and integrating the iterative training process can help improve the accuracy and performance of the text recognition system over time. Allowing users to provide feedback about the recognized text and using that feedback to retrain the model can lead to continuous improvements.

4. Uploaded our website to hosting services:

Once our website files have been uploaded to hosting services, our website will be accessible to the public. It's the best way to make our website available to the public and to ensure that it is always up and running.

5. Mobile Integration:

We will consider developing mobile and web versions of the system to make it available across different platforms. This will increase its reach and ease of use, allowing users to easily perform text detection and recognition tasks from their mobile devices or web browsers.

6. UI Improvements :

Continuous improvement of the user interface based on user feedback and usability studies can greatly enhance the overall user experience. We will incorporate intuitive design elements, clear instructions, and visual cues to guide users through the process of uploading images, extracting text, and interacting with the system.

5.6. Recommendations :

1. Image Format Validation:

Instead of just displaying an error message when a user inserts an image with an unsupported format, it would be very handy to provide an option to convert the image format automatically. For example, the system can prompt the user to convert the image to a supported format or provide a link to an online image converter.

2. Progress indicators:

When processing large or complex images, it would be useful to incorporate progress indicators or load animations to provide feedback to the user that the system is actively working on an image recognition task. This will improve the user experience by reducing uncertainty and by communicating that the system is working as intended.

3. Fault handling and robustness:

The system must be designed to handle exceptional cases gracefully. For example, if the image quality is too low for accurate text recognition, the system can make suggestions for improving the image quality or offer alternative ways to enter text.

Reference:

[1] Awni, Mohamed, Mahmoud I. Khalil, and Hazem M. Abbas. "Offline Arabic handwritten word recognition: A transfer learning approach." Journal of King Saud University-Computer and Information Sciences 34.10 (2022): 9654-9661.

[2] Boualam, Manal, et al. "Arabic handwriting word recognition based on convolutional recurrent neural network." WITS 2020: Proceedings of the 6th International Conference on Wireless Technologies, Embedded, and Intelligent Systems. Springer Singapore, 2022.

[3] Hamida, Soufiane, et al. "Efficient feature descriptor selection for improved Arabic handwritten words recognition." International Journal of Electrical & Computer Engineering (2088-8708) 12.5 (2022).

[4] Saidi, Abderrahmane, Abdelmouneim Moulay Lakhdar, and Mohammed Beladgham. "Recognition of offline handwritten Arabic words using a few structural features." Computers, Materials and Continua 66.3 (2021): 2875-2889.

[5] F. U. F. &. M. M. Qomariyah, "Handwriting Arabic Character Recognition Using Features Combination," in IJID (International Journal on Informatics for Development), 2021, p. 62–71.

[6] H. A. H. &. B. M. Balaha, " Automatic recognition of handwritten Arabic characters: a comprehensive review," in Neural Comput & Applic 33, 2021, p. 3011–3034.

[7] N. A.-T. Altwaijry, "Arabic handwriting recognition system using convolutional neural network," in Neural Comput. Appl., 2020.

[8] Ahmad, Riaz, et al. "A deep learning based Arabic script recognition system: benchmark on KHAT." Int. Arab J. Inf. Technol. 17.3 (2020): 299-305.

[9] Mustafa, Mohamed Elhafiz, and Murtada Khalafallah Elbashir. "A deep learning approach for handwritten Arabic names recognition." International Journal of Advanced Computer Science and Applications 11.1 (2020).

[10] A. M. a. M. Sadoon, "Developing an Arabic Handwritten Recognition System by Means of Artificial Neural Network," in Journal of Engineering and Applied Sciences, 2020, pp. 1-3.

[11] C. S. a. D. Si, "Skew and slant correction for document images using gradient direction," in Proceedings of the Fourth International Conference on Document Analysis and Recognition, Ulm, Germany, 2019, pp. 142-146.

[12] K. T. Shorten C, "A survey on image data augmentation for deep learning," in J Big Data, 2019, p. 60.

[13] Mohsin, Ali, and Mohammed Sadoon. "Developing an Arabic handwritten recognition system by means of artificial neural network." J. Eng. Appl. Sci 15.1 (2019): 1-3.

[14] Khémiri, Akram, Afef Kacem Echi, and Mourad Elloumi. "Bayesian versus convolutional networks for Arabic handwriting recognition." Arabian Journal for Science and Engineering 44 (2019): 9301-9319.

[15] Amrouch, Mustapha, Mouhcine Rabi, and Youssef Es-Saady. "Convolutional feature learning and CNN based HMM for Arabic handwriting recognition." Image and Signal Processing: 8th International Conference, ICISP 2018, Cherbourg, France, July 2-4, 2018, Proceedings 8. Springer International Publishing, 2018.

[16] M. K. E. a. M. E. Mustafa, "International Journal of Advanced Research in Computer and Communication Engineering," in Convolutional neural network model for arabic handwritten characters recognition, 2018, p. 11.

[17] Tamen, Zahia, Habiba Drias, and Dalila Boughaci. "An efficient multiple classifier system for Arabic handwritten words recognition." Pattern Recognition Letters 93 (2017): 123-132.

[18]. C. B. a. M. Batouche, "Investigation on deep learning for off-line handwritten arabic character recognition using theano research platform," in Intelligent Systems and Computer Vision (ISCV), 2017, pp. 1-6.

[19] A. A. Shahin, Printed Arabic Text Recognition using Linear and Nonlinear Regression, Int. J. Adv. Comput. Sci. Appl., vol. 8, 2017.

[20] Jayech, Khaoula, Mohamed Ali Mahjoub, and Najoua Essoukri Ben Amara. "Arabic handwritten word recognition based on dynamic bayesian network." Int. Arab J. Inf. Technol. 13.6B (2016): 1024-1031.

[21] M. T. P. a. S. A. Mahmoud, "Offline arabic handwritten text recognition," in a survey. ACM Computing Surveys (CSUR), 2016, p. 23.

[22] Mohamed Elleuch, Najiba Tagougui, and Monji Kherallah, "International Multi-Conference on Systems, Signals & Devices (SSD15)," in Arabic handwritten characters recognition using deep belief neural networks, 2016, pp. 1-5.

[23] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning. nature," 2016, p. 426.

[24] L. A, "A survey on Arabic character recognition," in Int J Signal Process Image Process Pattern Recognit, 2015, pp. 401-426.

[25] G. A. T. Q. ,. E. A. Abandah, "Recognition handwriting Arabic words using grapheme segmentation and neural networks," in international Journal on Document Analysis and Recognition (IJDAR), 2015, pp. 275-291.

[26] Y. M. Alginahi, " International Journal on Document Analysis and Recognition (IJDAR)," in A survey on arabic character segmentation, 2013, pp. 105-126.

[27] A. S. B. B. A.-A. T. O. K. Abu-Ain W, "Skeletonization algorithm for binary images," in Procedia Technol, 2013, p. 704–709.

[28] S.-S. ..-Z. K. A.M.AL-Shatnawi, "Offline arabic text recognition an overview," in World of computer Science and information Technology Journal (WCSIT), 2013, pp. 184-192.

[29] M. L. a. M. k. Zied T., "Arabic Cursive Character Distribution Recognition using the DTW Algorithm on BOINIC: Performance Analysis (IJACSA)," in International Journal of Advanced Computer Science and Applications, 2013, pp. 75-79.

[30] M. D. S. M. Aljuaid H, "Evaluation approach," in Int J Comput Vis Image Process, 2013, p. 58–77.

[31] P. &. A. A. Nagabhushan, " International Journal on Computer Science and Engineering.," in Tracing and Straightening the Baseline in Handwritten Persian/Arabic Text-line: A New Approach Based on Painting-technique, 2012.

[32] J. C. a. R. B. Z. Q. Liu, "Springer," in Handwriting Recognition Soft Computing and Probabilistic Approaches, 2012.

[33] O. K. Atallah A-S, "the state of art. IJCSNS," in Methods of Arabic language baseline detection, 2012, p. 137.

[34] Y. K. Abandah GA, "Handwritten Arabic character recognition using multiple classifiers based on letter form," in In Proceedings of the 5th conference on signal processing. Pattern recognition application, Innsbruck, Austria, 2012, p. 128–133.

[35] E. M. R. A. M. D. Amrouch M, "Off-line Arabic handwritten characters based on hidden Markov models," in Proceedings of the 3rd international conference on image and, Cherbourg-Octeville, France, 2011, p. 447–454.

[36] E. Y. M. M. D. Rachidi A, "A pre-topological approach for handwritten isolated Arabic character recognition," in In Proceedings of the second international symposium on communication, 2010.

[37] N. N. Amor, "Multifont Arabic characters Recognition Using Hough transform and HMM/ANN Classifications," in Journal of multimedia, Val 1,No. 2, 2010.

[38] M. Roushdy, "Comparative study of edge detection algorithms applying on the grayscale noisy image using morphological filter," in International Journal of Graphics, Vision, and Image Processing GVIP, 2009, pp. 17-23.

[39] V. G. a. M. P. F. Farooq, "Pre-processing methods for handwritten Arabic documents," in Eighth International Conference on Document Analysis and Recognition (ICDAR'05), Seoul, Korea (South), 2009, p. 191.

[40] B. C. a. J. .. -M. M. A. Buades, "A non-local algorithm for image denoising," in Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2009, pp. 60-65.

[41] L. L. a. V. Govindaraju, "Segmentation and Pre-recognition of Arabic Handwriting," in In Proceedings of the Eighth International Conference on Document Analysis and Recognition, vol. 2, 2002, pp. 605-609.

[42] A. H. a. R. Haraty, "A Neuro-heuristic Approach for Segmenting Handwritten Arabic Text," in In International Conference on Computer Systems and Applications, 2001, pp. 110-113.

[43] R. P. a. S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," in in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, 2000, pp. 63-84.

[44] T. T. Trier OD, "Evaluation of binarization methods for document images," in IEEE Trans Pattern Anal Mach Intell, 2000, p. 312–315.

[45] P. Baheti, "1," v7labs, 31 1 2023. [Online]. Available: https://www.v7labs.com/blog/handwriting-recognition-guide. [Accessed 1 2 2023].

[46] M. Abdullah, 19 Nov 2022. [Online]. Available: https://github.com/ranahaani/htlr.git [Accessed 21 2 2023].