Adaptive OCR Technique for Arabic Characters

Anwar H. Al-Saleh

Computer Department, College of Science, Mustansiriyah University, Baghdad, Iraq

anwar.h.m@uomustansiriyah.edu.iq

Abstract

OCR is an artificial intelligence, recognition English characters have been solved by a huge number of studies last 3 decades, but Arabic letters which used by at least one billion people have not been definitively resolved yet, therefore In this study proposes an effective approach to the recognition of off-line of 121 Arabic characters which is involves of creating database (DB), contains standard image for 59 main character target and Characters Indexing Table (CIT), contains indices of 62 main character target to be used in recognition process. The (DB) is designed to identify the main body of the 59 unknown characters. The (CIT) is responsible of identifying the secondary components, the number of dots or hamza that is associated with the body of character and their position. The mean square error classifier (MSE) is used in the recognition stage, where each character is assigned the best 6 matches with the lowest MSE. The lowest MSE value is 0.025 and the highest value is 0.٠٩.

Key-Words: - Mean Square Error MSE, Character Recognition CR, Arabic Character Recognition AOCR, and Pattern Recognition.

1 Introduction

Continuous growth in information technologies, new development tools and methodologies are used to identify isolated characters. Recognition of isolated characters is the process of identifying individual characters. It is useful in wide range of real world problems like documentation verification, interpreting mailing addresses, documentation analysis, handling bank checks

and many others. Because of recognition's applications, it is one of the most challenging areas in pattern recognition[1]. The main goal of isolated character recognition is to simulate human reading abilities. Character recognition systems can be improved the interaction between human and machine in many applications[2, 3]. The recognition of Arabic characters are very difficult compared to the recognition characters of other scripts [4, 5]. English text characters are separated from one to another with spaces and don't have the extra complexities associated with Arabic characters. So, the most commercially available OCR products are for the typed English text, and that is the reason of why English OCR systems are easier and well developed than Arabic OCR systems[6]. The recognition of character is a process to detect isolated and identifying characters from text image. More precisely, character recognition is a process to detect and recognize characters from the input text image and converts it to ASCII code, for Information Inter-change or other equivalent machine editable forms[7]. It contributes significantly to advance of automation process and improving the interface process. Character recognition is one of the most fascinating and interesting fields in artificial intelligence and pattern recognition[8, 9]. It is getting more and more attention where for previous decade due to its wide range of applications. Different techniques are used for designing of OCR based on their characteristics: feature extraction, neural networks, matrix matching, and fuzzy logic.

2 Related works

Various researchers have worked in character recognition approach using different methods, some of these works as follow:

El-Sheikh and El-Taweel (1990) proposed a segmentation approach for OCR system based on dividing the Arabic characters into 4 groups of letters (initial, meddle, final and isolated) and implement segmentation based on the stroke. This approach extremely sensitive to noisy data in terms of number of strokes since the recognition system was built on counting the exact number of stroke[10].

Al-Yousefi and Udpa (1992) introduced a statistical approach to recognize isolated printed and handwriting Arabic letters. It includes the segmentation of each letter into two parts: primary and secondary (such as dots and small markings) and normalization by moments of horizontal and vertical projections. The features were skew, nine measurements of kurtosis and relationships of moments, the recognizer was a quadratic Bayesian classifier[11].

Mozaffari et al. (2006) presented a recognition system of Arabic/Farsi handwritten-numeric characters which is used the statistical and the structural features. They used the nearest neighbor classifier for recognition, and the classification accuracy obtained was 94.44% [12].

Abdurazzag A. Aburas and Salem A. Rehiel (2008) presented a structure of off-line OCR system, which benefited from the (JPEG) image compression property that is high compression ratio and produces minimum compressed image size with every compressed image has a unique vector which helps to recognize each character. The result was considerably high in terms of accuracy and recognition rate[13].

Razzak, et al. (2010) proposed an Arabic OCR system for Urdu script that is written in Naskh and Nasta'liq styles based on a novel technique for segmentation and recognition. First, form the raw input strokes, the secondary strokes are segmented. Second, using the horizontal projection on ghost shapes the primary baseline is extracted. Then, the local baseline of each ligature is estimated based on extracted features and the estimated primary baseline. This approach obtained good results because of the utilization of baseline estimation over global reduction of diacritical marks. The proposed approach obtained an accuracy of 91.7% and 80.3% for Naskh and Nasta'liq font, respectively[14].

Al-Shatnawi et al. (2014) they are estimating and comparing skeleton of Arabic character extraction techniques. By reviewing the main stages of hand-written Arabic OCR systems, and presenting the major techno challenges of

these systems. Also, provided a comprehensive review of the works done in this field[15].

Mansoor Alghamdi (2017), produce a study in Arabic OCR systems, in his study provides the practical protocol with a set of quantitative performance measures that enables the effectiveness of different OCR systems to be compared. Where the recognition accuracy rates for separated characters are higher than for initial, middle and end characters. Also, the recognition accuracy rates with no dot characters are significantly better compared to the characters with dots[16].

Farhan M. A. Nashwan (2017) he introduced efficient holistic Arabic OCR system. A lexicon reduction method based on clustering similar to shape words was used to reduce time of recognition. Using global word level discrete cosine transform based features in combination with local block based features. He generalized for new font sizes that were not included in the training data. The results evaluation for the introduced method using different test sets from modern and historical Arabic books are promising compared with the state of Arabic OCR systems[17].

Nabeel M. Mirza (2018) portrays recognition strategies, these strategies depend on new pre-processing processes, extraction the structural and numerical features to build databases for printed alphabetical Arabic characters. The database information that obtained from features extracted was applied in recognition stage. Minimum Distance Classifier technique (MDCT) was used to classify and train the classes of characters. The procedure of one character against all characters was used in determining the rate of recognition. The suggested approaches have yielded unique and encouraging results in terms of accuracy in which the recognition rate reached to 97.28%[18].

Arabic characters in their different shapes in different locations exceed 121 cases, which are very similar in shape, making the recognition process of them by extracting features very difficult process and takes longer time. So in this study adopting OCR method includes MSE measure to recognizing separated printed Arabic letters after performs some preprocessing operations including:

noise removal, binarization and normalization. Firstly create main database (DB) for similar main body letters (contains 59 classes). Secondly create Characters Indexing Table (CIT) for the differences letters of same main body (contains 62 classes), so 121 different characters recognized.

3. Arabic Characters Properties

Arabic text is written from right to left and is always cursive in hand-written and machine printed text [19]. Arabic characters are more difficult than others languages, such as Chinese, Hindi, and Latin [20]. In contrast to English, Arabic characters are not divided into upper and lower case categories. Also, the Arabic character might have several shapes depending on its relative position in a word. Arabic character is written as a single main target (MT), and in most cases are completed with other complementary targets such as dot(s), **hamza**, **mada**, (In fact, 5 cases for different letters with same main target), see Table (1). The complementary targets might be placed above, or below of the main target (ف ك ب ق). Some Arabic characters have a unique main target (as shown in Table (1); case1, from id=1 to id=21); other Arabic characters have same main target but they are distinguished from each other only by the presence/absence, position or number of complementary targets, (as shown in Table (1); case2 id=22 to 45, case3 id=46 to 53, case4 id=54 to 55, case5 id=56 to 59). An example of different characters with similar main target is (ح خ ج), (ط ظ), (س ش), (و ؤ). Ambiguous writing of these complementary targets sometimes causes a word image to be read in many various forms with completely different meanings, like (جمل حمل). These complementary targets make the recognition of Arabic characters more difficult.

Table (1) Main character target (MT), similar targets, and number of Cases for each Arabic

Index (id)	Main Char. Target (MCT)	characters of Same Main Target (CSMT)	Case No. (CN)	Index (id)	Main Char. Target (MCT)	characters of Same Main Target (CSMT)	Case No. (CN)	Index (id)	Main Char. Target (MCT)	Characters of Same Main Target (CSMT)	Case No. (CN)	Index (id)	Main Char. Target (MCT)	characters of Same Main Target(CSMT)	Case No. (CN)
1	ء	ء	1	16	ک	ک	1	31	س	ش س	2	46	ى	ي ئ ى	3
2	ھ	ھ	1	17	ك	ك	1	32	ـسـ	ـشـ ـسـ	2	47	ی	ي ئ ی	3
3	ھ	ھ	1	18	ق	ق	1	33	ـسـ	ـشـ ـسـ	2	48	ج	ج خ ج	3
4	ں	ن	1	19	ق	ق	1	34	ـو	ـؤ ـو	2	49	ح	ج خ ح	3
5	ں	ن	1	20	ف	ف	1	35	و	ؤ و	2	50	ـحـ	ـجـ ـخـ ـحـ	3
6	م	م	1	21	ف	ف	1	36	ـه	ـة ـه	2	51	ـح	ـج ـخ ـح	3
7	م	م	1	22	ع	غ ع	2	37	ه	ة ه	2	52	ـب	ـت ـث ـب	3
8	ـمـ	ـمـ	1	23	ع	غ ع	2	38	ـف	ـقـ ـفـ	2	53	ب	ت ث ب	3
9	ـم	ـم	1	24	ـع	ـغـ ـعـ	2	39	ف	ـقـ ـفـ	2	54	ـب	ـپـ ـتـ ـثـ ـنـ ـبـ	5
10	ل	ل	1	25	ع	غ ع	2	40	ط	ظ ط	2	55	ـب	ـپـ ـتـ ـثـ ـنـ ـبـ	5
11	ل	ل	1	26	ص	ض ص	2	41	ط	ظ ط	2	56	ـلا	ـلإ ـلأ ـلآ ـلا	4
12	ـلـ	ـلـ	1	27	ص	ض ص	2	42	ر	ز ر	2	57	لا	لإ لأ لآ لا	4
13	ـل	ـل	1	28	ـصـ	ـضـ ـصـ	2	43	ر	ز ر	2	58	ـا	ـإ ـأ ـآ ـا	4
14	ـك	ك	1	29	ـص	ض ص	2	44	ـد	ـذ ـد	2	59	ا	إ أ آ ا	4
15	ك	ك	1	30	س	ش س	2	45	د	ذ د	2				

The shape of an Arabic character changed if it is located at the beginning, middle, end of the word or separated. For example, (ع ـع ـعـ عـ), (ب ـب ـبـ بـ), & (ه ـه ـهـ هـ). In addition to the 29 characters, Arabic has additional non basic characters such as **hamza** (ء) and short **Taa** (ة ـة). **Hamza** can be shown separated or in upper/lower of some letters (إ أ ؤ ئ ء). Short **Taa** is a special form of the letter **Taa** (ت) that only appears at the end of words. Arabic language incorporates some ligatures such as (لا) which actually consists of two letters (ل & ا), but when connected produce another character that different in shape for both merged letters (ل & ا). This new character together with the combination of **hamza** and **madda** (ء ~) increase the number of classes to 121 (لا ـلا لأ ـلأ لإ ـلإ لآ ـلآ). In this study all the 121 possible classes of Arabic characters are adopted.

Table (2) shows the list of 121 Arabic letters in their different shapes in different locations (each table cell contains one character with different shapes).

Table (2) The 121 Arabic Characters in their different shapes

ا ـا أ ـأ إ ـإ آ ـآ	خ ـخ ـخ	ش ـش ش ـش	غ ـغ ـغ	ن ـن ـن	ؤ ـؤ
ب ـب ـب ـب	د ـد	ص ـص ص ـص	ف ـف ـف	ه ـه ـه	لا ـلا لأ ـلأ لإ ـلإ آ ـلآ
ت ـت ـت ـت	ذ ـذ	ض ـض ض ـض	ق ـق ق ـق	و ـو	ة ـة
ث ـث ـث ـث	ر ـر	ط ـط	ك ـك ك ـك	ي ـي ي ـي	ء
ج ـج ـج ـج	ز ـز	ظ ـظ	ل ـل ل ـل	ى ـى	
ح ـح ـح ـح	س ـس س ـس	ع ـع ع ـع	م ـم م ـم	ئ ـئ ئ ـئ	

4 The proposed Arabic Characters Recognition Technique

The similarity among each group of different characters and the large number of patterns of Arabic characters, causes large number of classes that complicate the recognition system. Also, this problem makes the recognition system unable to absolutely decide what character is under recognition. In order to solve these problems, this study proposes a recognition technique that is divided into two parts: first part create database (DB) contains 59 main target of all Arabic characters, and save standard image for each of 59 main character target (MCT). Second part create Characters Indexing Table (CIT) contains indices of 62 MCT to be used in recognition process. Figure (1) shows the block diagram of the proposed system, and illustrate both input images to create BD and CIT, and Table (3): (a) shows the database (DB) of standard image for all 59 main target, (b) the Characters Indexing Table (CIT) of standard images for all 62 main target, and (c) the estimated id for input image contains some characters (لإ بـ ـب ى غ ـه) either from DB of 59 indexing or from CIT of 62 indexing.

4.١ Preprocessing and Extracting Main Character Target Image (MCT)

preprocessing for input characters' image is necessary to prepare the extracted character image for remainder stages. In order to extract the main target of each character in the input image, it should first be converted to binary image. For binarization the image a threshold value (th) is required. the MATLAB function (im2bw) is used, and th = 0.9. So, if the value of the pixels in the

image is less than the threshold value, it is expressed as "0"; and if it is greater than the threshold value, it is expressed as "1".

There are several preprocessing steps used for the Arabic characters' images before performing OCR, these preprocessing steps as follows:

a- The MATLAB function regionprops perform scanning processing vertically for the 59 binary image from top to down and from left to right to extract main target of the characters, without dots or **hamza**. To reduce processing time and simplify recognition procedure, the size of segmented target must be resizing into 32×32 pixels as a standard size of the main target. Then save standard image for each main target to create DB, see Table (3a). the input image is of font Times New Roman, size 22, th=0.9, repeat this step to the same image using two other thresholds (0.8, 0.7), then re-index the database from 1 to 59, so the size of database become 177 (59×3) main target images. To increase the accuracy of the recognition, input two images for the same font but with different sizes (24, 26) and three thresholds (0.9, 0.8, 0.7), in this case the final size of database is 531 (177×3) main target images, also re-index the database from 1 to 59.

b- Again used MATLAB function regionprops to scan image vertically now for the 62 binary image from top to down and from left to right to extract main target of the characters. Also extract number, position, and area of complementary targets in each character image, and the total area of all complementary targets. The size of segmented main target must be resizing to a fixed size (32×32 pixels) as a standard size of the main target, then save standard image for each main target to create CIT, see Table (3b).

4.٢ Create Database (DB) and Characters Indexing Table (CIT)

The first step of propose recognition system is to create the database DB contains 59 main characters target and introduce CIT contains 62 main characters target using input images that contains these characters, as shown in Figure (1) and Table (3 a and b). In this study arranged the Arabic characters in utilized input images to build DB and CIT must be compatible with analysis conditions in recognition algorithm.

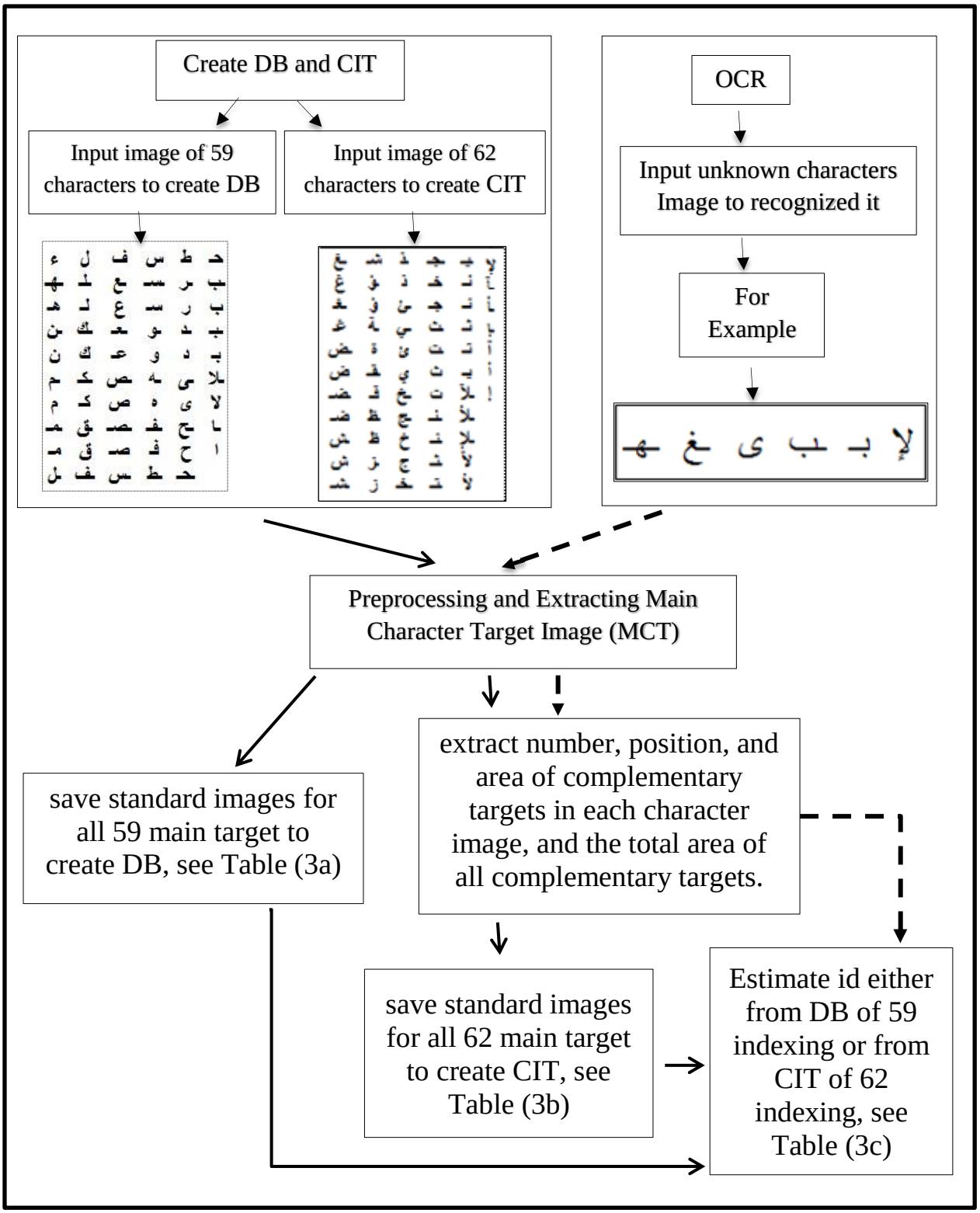


Figure (1) the block diagram of the proposed system, where ⟶ represent the path of creating BD & CIT and - - ➔ represent the path of OCR result for Input unknown characters Image.

Table (3) (a) The database (DB) of standard images for all 59 main target, (b) The Characters Indexing Table (CIT) of standard images for all 62 main target, (c) The estimated id for input image contains characters (ﻫـ ﻎ ى ﺐ ﺑ لإ) either from DB or from CIT.

(a) 59 DB

id	Img. Char.	The Text	id	Img. Char.	The Text	id	Img. Char.	The Text
1		ء	21		ف	41		ط
2		ﻫـ	22		ع	42		ر
3		ﻫ	23		ع	43		ر
4		ن	24		ﻋـ	44		ﺪ
5		ن	25		ﻋ	45		د
6		ﻡ	26		ﺺ	46		ى
7		م	27		ص	47		ى
8		ﻤـ	28		ﺼـ	48		ح
9		ﻤ	29		ﺼ	49		ح
10		ل	30		س	50		ﺤـ
11		ل	31		س	51		ﺣ
12		ﻠـ	32		ﺴـ	52		ﺐ
13		ﻠ	33		ﺴ	53		ب
14		ك	34		ﻮ	54		ﺒـ
15		ك	35		و	55		ﺑ
16		ﻜـ	36		ﻪ	56		لا
17		ﻛ	37		ه	57		لا
18		ق	38		ﻔـ	58		ﺎ
19		ق	39		ف	59		ا
20		ﻓ	40		ط			

(b) 62 CIT

id	Img. Char	The Text	id	Img. Char	The Text	id	Img. Char	The Text
1		غ	22		ز	43		ﺛ
2		غ	23		ذ	44		ﺗ
3		ﻏـ	24		ذ	45		ﻴ
4		غ	25		ئ	46		ﺌ
5		ﺿ	26		ي	47		ﻧ
6		ض	27		ئ	48		ﺛ
7		ﻀـ	28		ي	49		ﺗ
8		ﺿ	29		ﺧ	50		ﻴ
9		ش	30		ج	51		ﻟآ
10		ش	31		خ	52		ﻟأ
11		ﺸـ	32		ج	53		ﻟإ
12		ش	33		ﺨـ	54		لآ
13		ﺆ	34		ﺟـ	55		لأ
14		ﺆ	35		ﺧ	56		لإ
15		ﺔ	36		ﺟ	57		آ
16		ة	37		ث	58		أ
17		ﻘـ	38		ت	59		إ
18		ﻗ	39		ث	60		آ
19		ظ	40		ت	61		أ
20		ظ	41		ﺋـ	62		إ
21		ز	42		ﺋ	63		

(c) OCR result

Input Img. Char.	id	Img. Char.	Char.	Estimated id from
	2			DB
	1			CIT
	47			DB
	52			DB
	55			DB
	56			CIT

4.3 The Suggested OCR System

The final stage of proposed system is recognizing of each unknown character that extracted from input test image. In this system Mean Square Error technique (MSE) was used for OCR process. This technique basically matches between the input character main target image after resizing into 32×32 pixels and the character main target images in DB and used CIT. The character identity is assigned based on the similarity value. The input of unknown character image, for example image input in Figure (1), contains characters (لإ ﻫـ ﻎ ى ﺐ ﺑ), preprocessing like in section 4.2 (a) for each unknown character estimate main character target from perform suggested OCR method

with DB and estimate the index (id) for the correct main character in DB. Then estimate the correct character using 5 cases conditions.
Figure (**2**) shows the processing of extracting the main target of the characters in the unknown input image.

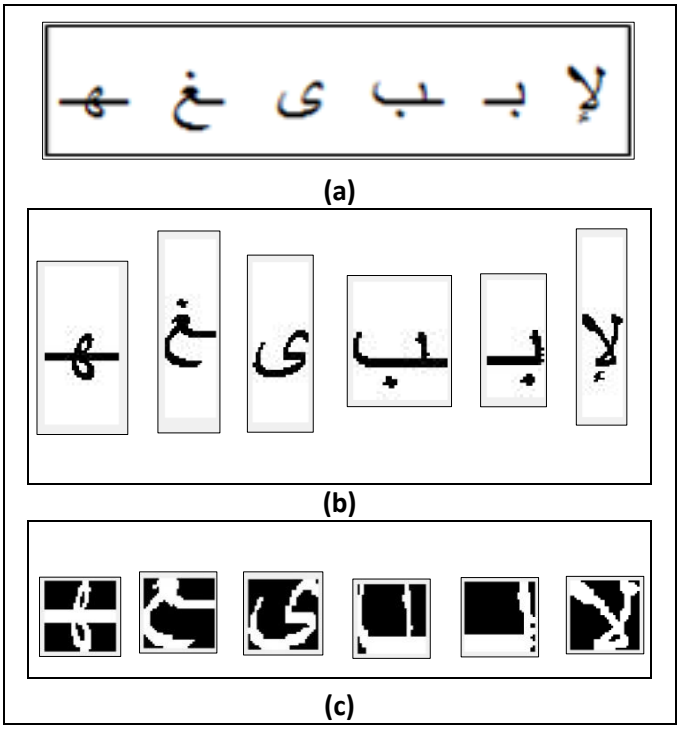


Figure (2) (a) Samples of input characters' images, (b) input character's images after binarization process, (c) resizing process.

4.4 Experimental Results

The process of recognizer cases is done in two stages: Stage 1: the extracted main target of the unknown input image character was comparing with each main target in DB, and OCR used minimum MSE to indicate the corrected Class Number (CN), the output index in DB, where the return value of the MSE process is used to identify the character, that is, return values of the comparison approaching to less value (approaches to zero) are assumed as good match, Stage 2: after the main target has been identified from the DB, the complementary of the character (dot, two dots, three dots or a **hamza**) are

now identified as well as their position (bottom, top or inside the character), used CIT. The recognition process is done through 5 checkup cases, as shown in Table (1). Each of these cases will be discussed independently as follow. Finally recognized characters are displayed. Figure (3) shows the block diagram of suggested OCR technique and all cases of recognition process.

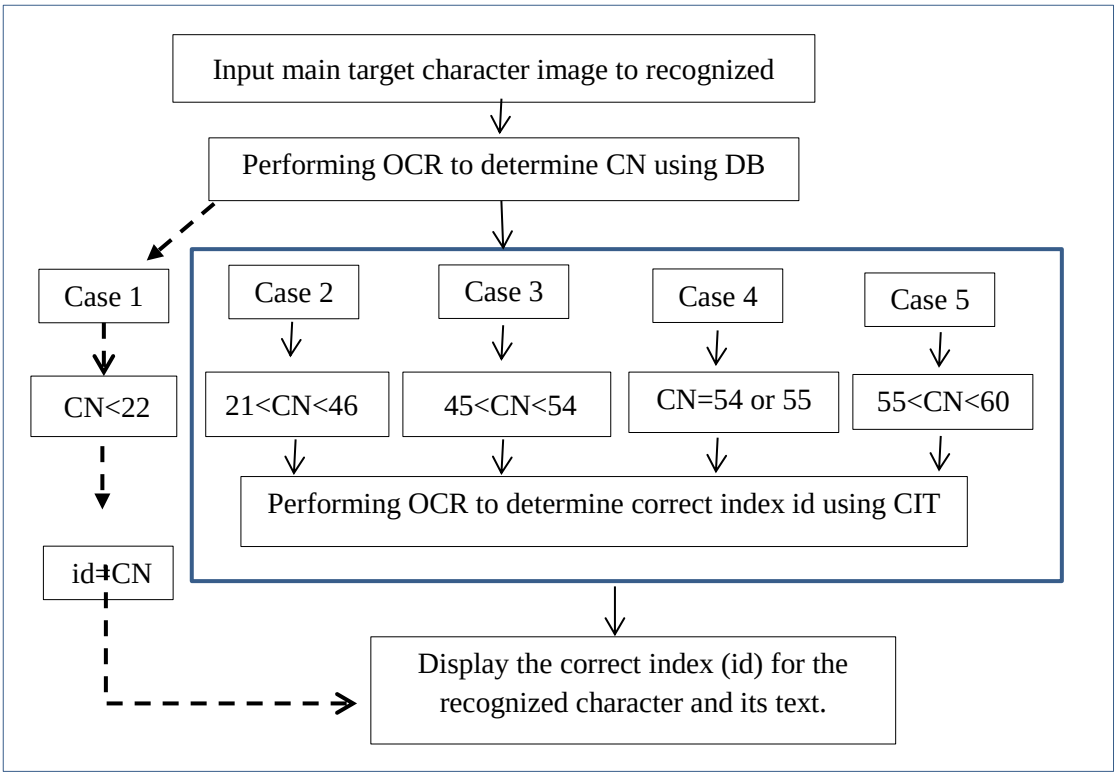


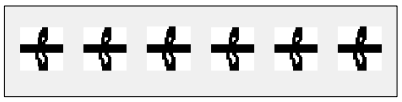
Figure (3) the block diagram of OCR technique and all cases of recognition process, where ⇢ path identification from only DB and → path identification from DB & CIT.

Case 1: if the output class number in stage 1 is less than 22, (CN<22), then the character of index CN is displayed, the correct index (id) of the input character is (id = CN), where id represent the output class index in the recognition system, so in this case don't need used CIT. Figure (4) shows example of case 1 to recognize character (ع), where can be shown that the matching process gives the best 6 matching results in DB, Figure (4b), in order to ensure the validity and conformity of recognition, then choose the least value of MSE (first of them) to be the recognized output.

input main target image

(a)

The 6 best matching OCR results with DB and its CN number & MSE's value for each one.



CN	2	2	2	2	2	2
MSE	0.025	0.026	0.033	0.034	0.034	0.039

(b)

Output index from DB

id=CN=2

(c)

Figure (4) Example of case 1, (a) the input character image to recognition system, (b) The 6 best matching OCR results with DB and its CN number & MSE's value for each one, (c) the output of the recognition system.

Case 2: if the output class number CN in stage 1 is in the range (22 - 45), i.e. (21< CN <46), then in this case to get final recognition result there are two possibilities, the recognized character have or not have one dot, (غ - ع), (ـص - ـض), (ذ - ـذ), two dots, (ـة - ـه), (ة – ه), three dots, (ش – س), (ـشـ - ـسـ), or **hamza**,(ؤ - و), (ـؤ - ـو). So to solve this problem, a simple condition is added, as shown below:

```
If  NT=1 then      // where NT represented the no. of targets in input char. Image.
   id=CN          // estimate index no. of this character is like case1, from DB.
Else              // the input character image contains more than one target.
   id =CN-22+1 // CN in this equation to a correct character index id, from CIT.
End               // end condition.
```

As shown in Figure (5), for example, if CN=22 then the recognizer in DB is the main target for character (ـع), in this case there are only two possibilities, if input image is (ـع) then NT=1, so id=CN=22 in DB table (Table (3a), and if input image is (ـغ), in this case NT>1, (NT=2), then id= CN-22+1=1 (Table (3b), the output class number from CIT.

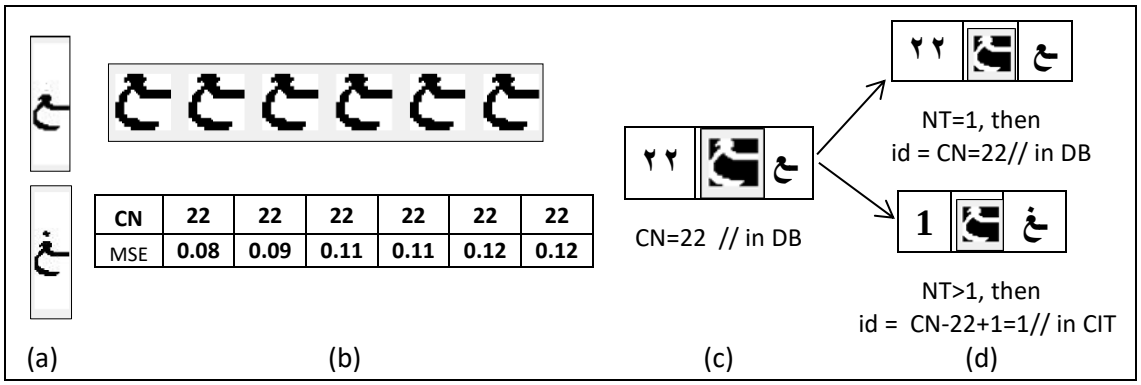
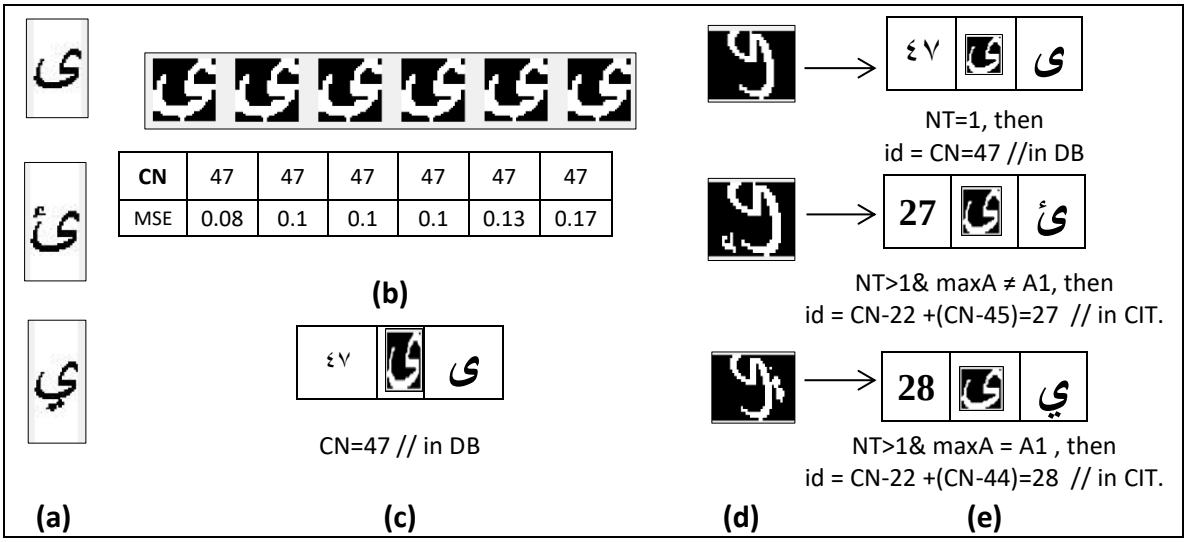


Figure (5) Example of case 2, (a) the input character images to recognition system, (b) the 6 best matching OCR result and table of its indexes MSE's values, (c) the output of the recognition system from DB and CIT.

Case 3: in this case the output OCR number CN from stage 1 is in the range (46 - 53), i.e. 45<CN<54, the characters in this range are (ى), (ح) and (ب) with its different positions (begin, medal, or separated). Then in this case to get final recognition index number id; there are three possibilities for each character, the recognized character has two dots down (ي), **hamza** up (ئ) or without dot (ى). Also there are three possibilities to the second character, where the recognized character has one dot up (خ), one dot within character (ج), or without dot (ح), and the three possibilities for third character is one dot down (ب), two dots up (ت), or three dots up (ث). So to solve this problem, the character image will be rotate by (90^o), then scan output image from left to right to determine the area of each target in the rotated image, taking into consideration the maximum area maxA and area of first target A1, finally a following condition is added:

```
If NT=1 then  // NT is the number of targets in the input character image.
  id=CN       // index output from DB.
Else
   If maxA ≠ A1 // Where maxA and A1 represent the maximum area and area of
                   first target in the rotated character image, respectively.
       id = CN-22 +(CN-45)   // index output from CIT.
   Else// in this case the area of first target in the rotated image is the maximum.
       id = CN-22 +(CN-44)   // index output from CIT.
   End
End
```

As shown in Figure(6), for example, if CN=47, (ى), then the recognizer in DB is the main target for character (ى), in this case there are three possibilities, the recognize character is (ى), where NT=1, then (id = CN =47), the output index in DB, or (ئ), due to NT>1 and the area of second target is the maximum area (maxA ≠ A1) then id=27, the index output from CIT. Finally the recognize character is (ي), NT>1 and the area of first target is the maximum area (maxA = A1), then id=28, the index output from CIT, Figure(6e).



Figure(6) (a) the input character images to recognition system, (b) the 6 best matching OCR result and table of its indexes MSE's values, (c) the output in DB, (d) rotated input images by 90°, (e) the output of the recognition system.

The above condition is very suitable solution in the case of the letters (ى) and (ح). But it is not appropriate in the case of the character (ب), because in both cases (ت) and (ث) the first target is not the maximum area (maxA ≠A1) in the rotated image. So another condition should be added to recognized (ت) from (ث). In this case, calculate the area of all dots in the character image (Sa), where clear that the area of the two dots is smaller than the area of three dots, so can be used the condition below to solve this problem:

```
if CN>51 and CN<54   // to recognize (ب ـب) (ت ـت) or (ث ـث).
    if maxA = A1      // that mean input image character is (ب) or (ـب).
       id =CN         // index output from DB.
    else              // maxA ≠ A1; so must recognized (ت ـت) from (ث ـث)
```

```
        if Sa>th1 // where Sa=(area of character image) - (area of maxA),  Sa=sum(A)-
                  maxA, i.e. it represents the area of all secondary strokes and in this
                  case it represents the area of three dots of character (ث ـث), th1=95,
                  (limit area in pixels).
          id=CN-22+(CN-45) // index output from CIT, recognition of characters (ث ـث).
        else  // i.e. Sa represents the area of two dots of character (ت ـت).
          id= CN-22 +(CN-44) //index output from CIT, recognition of characters (ت ـت).
        end
      end
  end
```

Figure (7), illustrate the recognition between (ـب), (ـث), and (ـت), where CN=52 then the OCR in DB is the main target for character (ـب), the recognize character is (ـب) if the area of first target is maximum area (maxA =A1), then (id=CN =52). The recognize character is (ـث) if the maximum area not first target (maxA ≠A1), and the area of all dots greater than th1=95, (Sa>95), then id= 37. Finally, the recognize character is (ـت) if condition1 is true but condition2 is not, i.e. the area of all dots lees than th1, (Sa<95), then id= 38, Figure (7e).

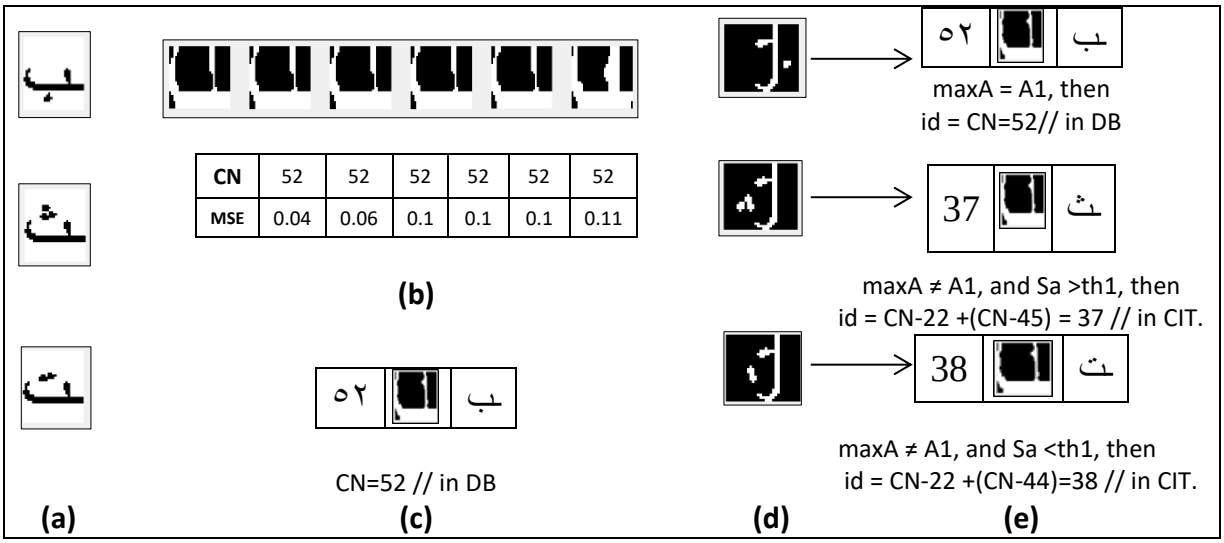


Figure (7) (a) the input character images to recognition system, (b) the 6 best matching OCR result and table of its indexes MSE's values, (c) the output in DB, (d) rotated input image by 90°, (e) the output of the recognition system.

Case 4: in this case the output CN in stage 1 is in the range (54 - 55), the characters in this range are (ـبـ, ـتـ). This case requires six checks, depending on whether dots or **hamza** are up or down the main target of the character as well

as the number of dots (ـبـ, ـيـ, ـتـ, ـثـ, ـنـ, ـئـ), (ـبـ, ـيـ, ـتـ, ـثـ, ـنـ, ـئـ). In the same way in case 3, rotate the character at an angle of 90°, where the rotation process will recognize (ـيـ, ـبـ) from (ـتـ, ـثـ, ـنـ, ـئـ), and (ـيـ, ـبـ) from (ـتـ, ـثـ, ـنـ, ـئـ), where the area of the first target A1 is the maximum. Now it is easy to distinguish between (ـبـ & ـيـ) and (ـبـ & ـيـ) depending on the total area of dots, where it is clear that the area of one dot (ـبـ ـبـ) is smaller than the area of the two dots (ـيـ ـيـ). So must be used the following condition:

```
if maxA =A1  //   (ـيـ ـيـ ـبـ ـبـ)
        if CN=54 and|Sa-B1| < th2     // to recognize (ـيـ), where B1=125 and th2=40,
                                                    Sa=sum(A)-maxA.
           id= CN-22 +(CN-41)    // in CIT.
        elseif CN=55 and | Sa-B1 |< th2   // to recognize (ـيـ)
           id= CN-22 +(CN-38)   //  in CIT.
        else                                    // to recognize (ـبـ, ـبـ).
           id= CN    //   in DB.
        end
 end
```

As shown in Figure (8).

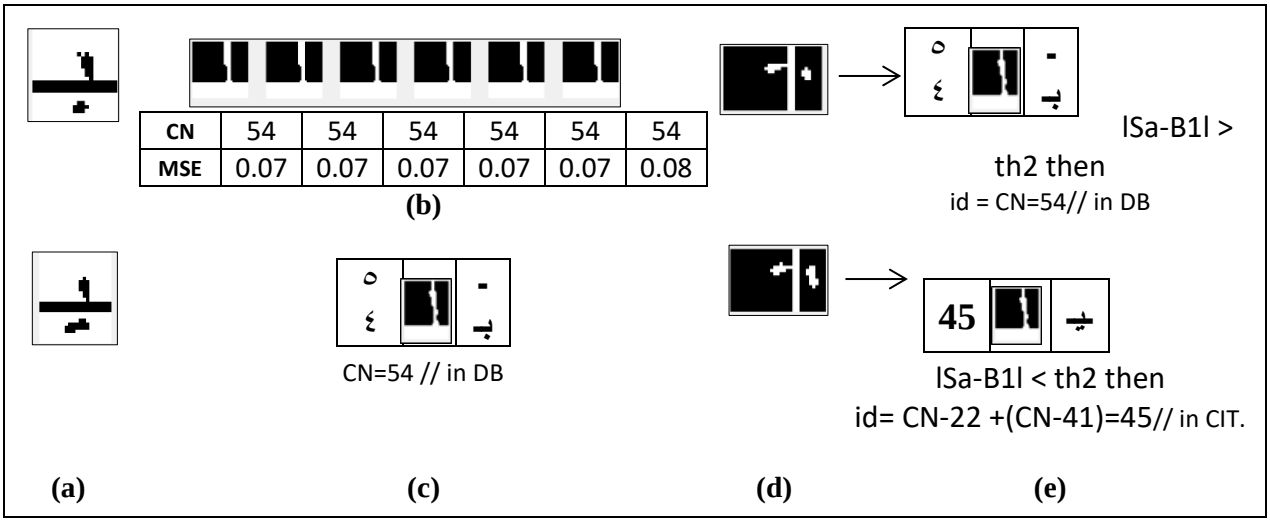


Figure (8) (a) is the input character images to recognition system, (b) The 6 best matching OCR results with DB and its CN number & MSE's value for each one., (c) the output in DB, (d) rotated input image by 90°, (e) the output of the recognition system.

On the other hand, if the first target A1 is not with the maximum area, then the characters to be recognized are one of (تـ, ثـ, نـ, ئـ) or one of (تـ, ثـ, نـ, ئـ). The condition has been used to recognized (تـ, تـ, ثـ, ثـ) is as follow:

```
if CN=54 and maxA ≠ A1 // (تـ ثـ); the first target A1 is not the maximum
        if |Sa-B2|<th3          //  B2=90, th3=15.
          id=CN-22 +(CN-42)  // تـ
        elseif |Sa-B3|<th3      //  B3=137.
          id=CN-22 +(CN-43)  // ثـ
elseif CN=55  and maxA ≠ A1    // (تـ ثـ)
        if |Sa-B4|<th3          //  B4=130.
          id=CN-22 +(CN-39) // تـ
        elseif abs(Sa-B5)<th3 // B5=187.
          id=CN-22 +(CN-40) // ثـ
 end
```

Figure (9) illustrates how the system recognized between tow input images of characters (تـ ثـ):

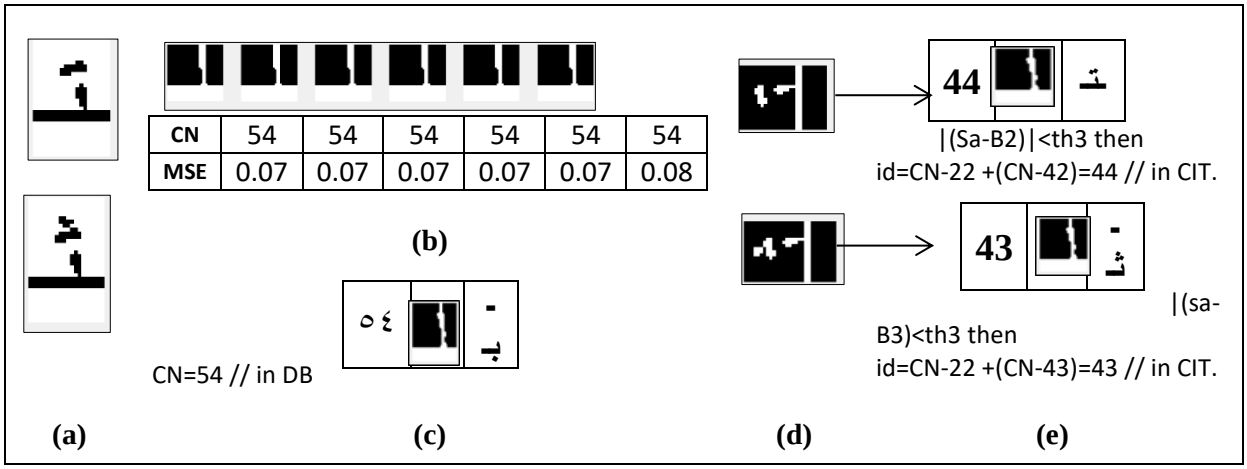


Figure (9) (a) is the input character images to recognition system, (b) The 6 best matching OCR results with DB and its CN number & MSE's value for each one., (c) the output in DB, (d) rotated input image by 90°, (e) the output of the recognition system.

Because of the great similarity between the two characters (نـ ـنـ) & (ئـ ـئـ) have need to use some features to distinguish these two characters. So used the following processing steps to distinguish these characters:

1- Identify and extract the secondary part of each character, **hamza** from (ـئـ ـئ), and dot from (ـنـ ـن), Figure (10b).
2- Extracts only the secondary targets from the image in a new image and resize the output image into (100×60), Figure (10c i & ii).
3- For each row in extracted secondary part image calculate the length of the straight line connecting between the first and last pixels not equal to zero, as shown in Figure(10c).
4- Determine the longest straight line and locate its position (the row number), and save the output features into feature-vector V, (V= [L, L_{pos}, $|L-L_{pos}|$]), where L represent the length of longest straight length, L_{pos} its position, and $|L-L_{pos}|$ is the absolute value of the difference between line length and its position, in Figure (10c) the red line illustrate the position of connected longest straight length.

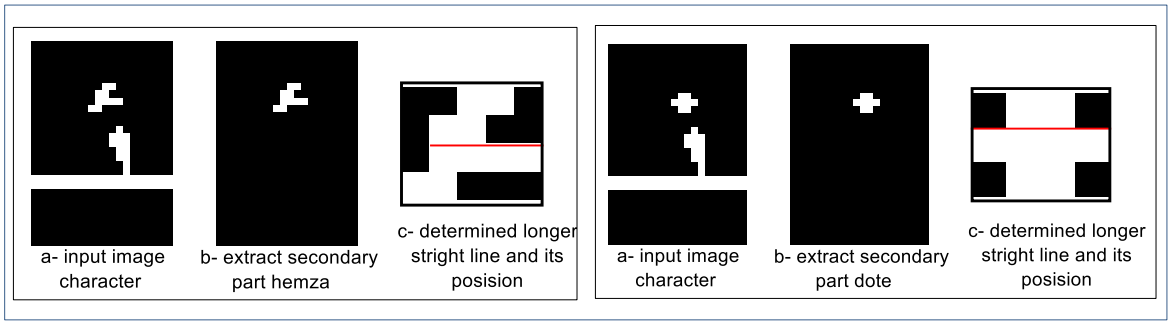


Figure (10) steps for extracted secondary parts (i) hamza and (ii) dot from character (ـئـ) and (ـنـ) respectively.

By extracting the features above the process of recognize is become easy, by adding the condition below:

```
if CN=54                                  // ـبـ in DB.
   if (V(2)-B6)<th4    // V(2) is The second value in vector V (Lpos), B6=15, th4=4.
        id=CN-22 +(CN-44)          //ـنـ in CIT.
   elseif
        id=CN-22 +(CN-45)          //ـئـ in CIT.
   end
if CN=55                                  // ـبـ in DB.
```

```
if (V(2)-B6)<th4
    id=CN-22 +(CN-41)    //ـنـ in CIT.
  elseif
    id=CN-22 +(CN-42)   //ـئـ in CIT.
  end
end
```

Figure 11) to describe an example of recognizing (ـنـ ـئـ):

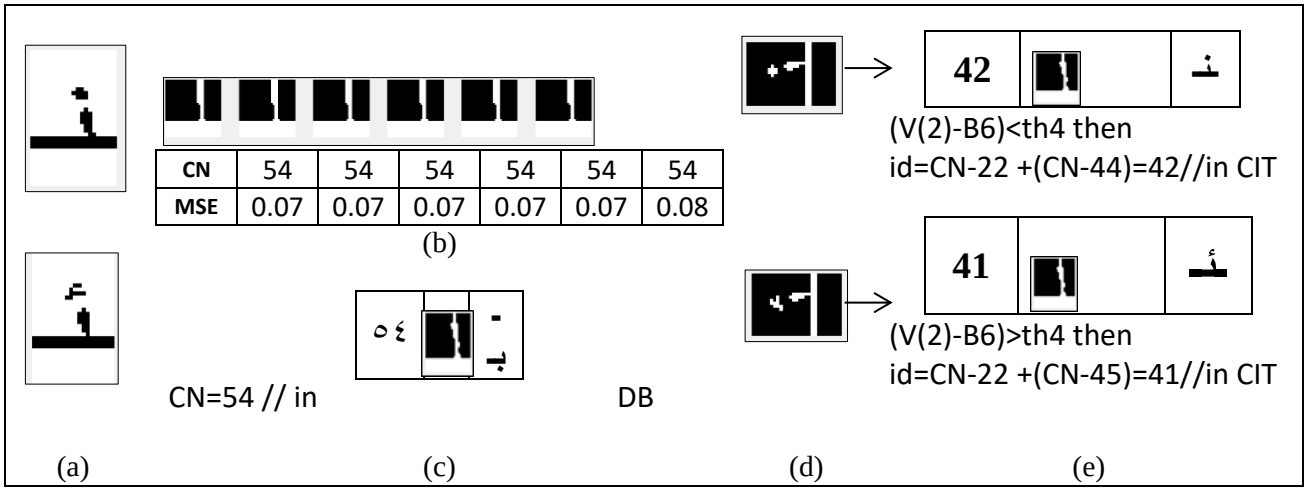


Figure 11 (a) is the input character images to recognition system, (b) The 6 best matching OCR results with DB and its CN number & MSE's value for each one., (c) the output in DB, (d) rotated input image by 90o, (e) the output of the recognition system.

Case 5: this is the last case, in which to recognize characters (ـلا ـلا ـلا). The condition used to complete recognition process is:

```
If CN>55 and CN<60
  if NT=1 // the input character image is of one target (ـلا ـلا ـلا ).
    id=CN // in DB.
  else   // the input character image is of two targets (ـلإ ـلأ ـلآ إ أ آ).
    if CN=56   / the input character image is one of these characters (ـلإ ـلأ ـلآ).
      if MaxA=A2        //  after rotation; (ـلأ ـلآ).
          Id=CN-22 +(CN-39) // in CIT.
      else              //  MaxA=A1, the input character image is  (ـلإ).
          Id=CN-22 +(CN-37)   //in CIT.
      end
    elseif CN=57  // the input character image is one of these characters (ـلإ ـلآ ـلأ).
      if MaxA=A2        //   (ـلآ ـلأ).
          Id=CN-22 +(CN-38)  // in CIT.
```

```
        else                    //    (لإ).
                Id=CN-22 +(CN-36)  // in CIT.
        end
    elseif CN=58   //  the input character image is one of these characters (لـإلـآلـأ).
        if MaxA=A2          //  (لـآلـأ).
                Id=CN-22 +(CN-37) // in CIT.
        else                //  (لـإ).
                Id=CN-22 +(CN-35)  // in CIT.
        end
    else          //  the input character image is one of these characters (إآأ).
        if MaxA=A2          //  (آأ).
                Id=CN-22 +(CN-36)   // in CIT.
        else                //  (إ).
                Id=CN-22 +(CN-34)     // in CIT.
        end
    end
  end
end
```

See Figure 12).

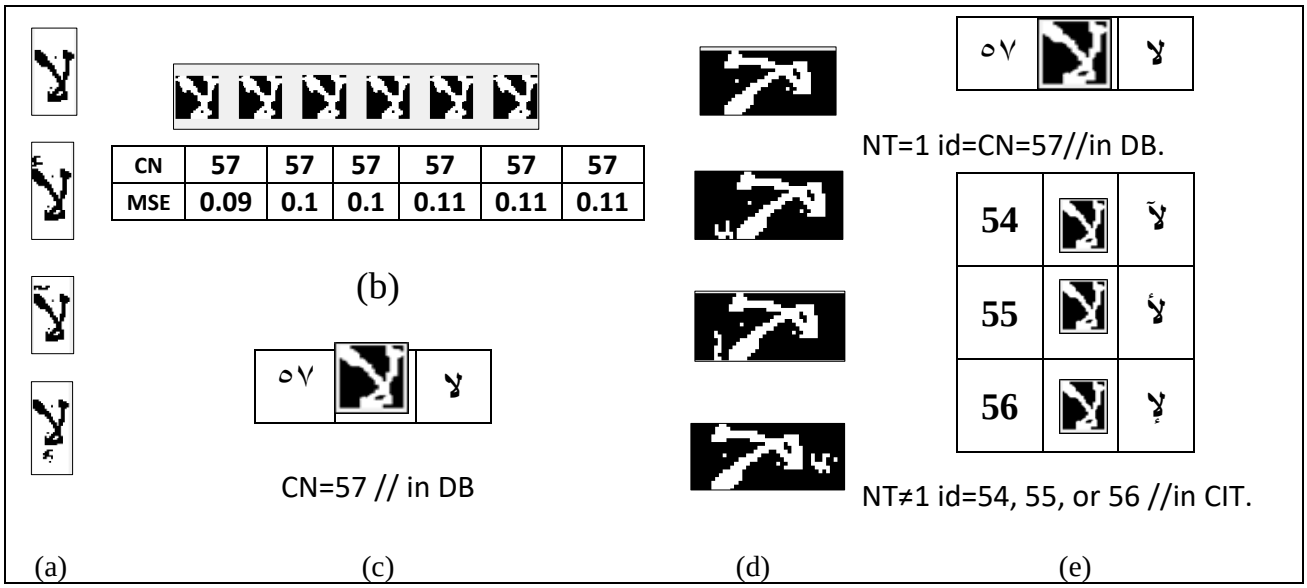


Figure 12 (a) is the input character images to recognition system, (b) The 6 best matching OCR results with DB and its CN number & MSE's value for each one., (c) the output in DB, (d) rotated input image by 90°, (e) the output of the recognition system.

5. CONCLUSIONS

This paper has proposed recognition technique to recognize isolated Arabic characters using OCR technique to recognize 121 deferent Arabic characters.
The proposed method was inclusive of all Arabic characters in their different positions and shapes, additional non basic characters hamza and short Taa, and incorporates some ligatures (لا). Where database DB of 59 classes and character index table CIT of 62 classes have created for recognition process. The tested Arabic character samples give correct recognition with a highest MSR value is (0.٠٩) for Arabic character (لا), while the lowest MSE value is (0.025)for Arabic character (ﻉ), where the number of accepted matches for each character was 6 cases. All cases gave the match with the correct locations in DB or CIT.

6. REFRENCES

1. Karanbir Kaur, N.K.G., *A Survey on Process of Isolated Character Recognition.* International Journal of Innovative Research in Computer and Communication Engineering, 2014. **2**(5): p. 4173-4178.
2. Abdel wahab zramdini, R.i., *Optical Font recognition from projection profiles.* Electroic publishing, 1993. **6**.
3. Zakaria, A.M.Z.a.M.S., *Challenges in Recognizing Arabic Characters.* International Islamic University alaysia(IIUM),Kuala Lumpur .Malaysia, 2004.
4. Suen, A.S.a.C., *OFF-LINE SYSTEM FOR THE RECOGNITION OF HANDWRITTEN ARABIC CHARACTER.* Computer Science & Information Technology (CS & IT), 2014: p. 227–244.
5. M. Cheriet, *Visual recognition of Arabic handwriting: challenges and new directions. In Arabic and Chinese Handwriting Recognition, Lecture Notes in Computer Science*,. Springer,, 2008. **4768**: p. 1-21.
6. Ahmad, M.A.-A.a.a.J., *Optical Character Recognition System for Arabic Text Using Cursive Multi-Directional Approach.* Journal of Computer Science 2007: p. 549-555.
7. N. VENKATA RAO, D.A.S.C.S.S., A.S.N.CHAKRAVARTHY, KALYANCHAKRAVARTHI P *OPTICAL CHARACTER RECOGNITION TECHNIQUE ALGORITHMS.* Journal of Theoretical and Applied Information Technology, 2016. **83**(2): p. 275-282.
8. J. Pradeepa, E.S., S. Himavathib, *Neural Network Based Recognition System Integrating Feature Extraction and Classification for English Handwritten.* International journal of Engineering, 2012. **25**(2): p. 99-106.
9. SupriyaDeshmukh, L.R., *Analysis of Directional Features - Stroke and Contour for Handwritten Character Recognition.* IEEE International Advance Computing Conference, 2009: p. 1114-1118.
10. El-Taweel, T.S.E.-S.a.S.G. and *Real-time arabic handwritten character recognition.* PR</cja:jid> Pattern Recognition, 1990

23(12): p. 1323-1332.

11. H. Al-Yousefi, a.S.S.U., *Recognition of Arabic characters.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992. **14**(8): p. 853 - 857.
12. S. Mozaffari, K.F., M. Ziaratban, *Structural decomposition and statistical description of Farsi/Arabic handwritten numeric characters.*

Proc. Int'l Conf. Document Analysis and Recognition, IEEE, 2006: p. 237-241.

13. Abdurazzag Ali Aburas , S.A.R., *JPEG for Arabic Handwritten Character Recognition: Add a Dimension of Application.* Advances in Robotics, Automation and Control , Vienna, Austria, 2008: p. 21-31.
14. M. I. Razzak, M.S., and S. A. Hussain, *Locally baseline detection for online Arabic script based languages character recognition.* Int. J. Phys. Sci. International Journal of Physical Sciences, 2010. **5**(7): p. 955-959.
15. Atallah M. Al-Shatnawi , K.O., Bader M. AlFawwaz , Ahmed M. Zeki, *Skeleton extraction: Comparison of five methods on the Arabic IFN/ENIT database.* Computer Science and Information Technology (CSIT), 2014 6th International Conference on , IEEE, 2014: p. 50-59.
16. Mansoor Alghamdi, W.T., *Experimental evaluation of Arabic OCR systems.* Emerald Insight, 2017. **1**(3): p. 229-241.
17. Farhan M. A. Nashwan , M.A.A.R., Hassanin M. Al-Barhamtoshy, *A Holistic Technique for an Arabic OCR System.* journak of Imaging www.mdpi.com/journal/jimaging, 2017. **4**(6).
18. Mirza, N.M., *Printed Arabic Characters Recognition Based on Minimum Distance Classifier Technique.* Iraqi Journal of Science, 2018. **59**(2A): p. 762-770.
19. Khorsheed, M.S., *Off-Line Arabic Character Recognition – A Review.* Pattern Analysis and Applications, Springer 2002. **5**(1): p. 31-45.
20. Jawad H. AlKhateeb, J.R., Stan S. Ipson, Jianmin Jiang, *Knowledge-Based Baseline Detection and Optimal Thresholding for Words Segmentation in Efficient Pre-Processing of Handwritten Arabic Text.* Fifth international conference on information technology: new generations. IEEE computer society, 2008: p. 1158-1159.