DOI: 10.24996/ijs.2023.64.4.35

ISSN: 0067-2904

Automatic Numeral Recognition System Using Local Statistical and Geometrical Features

Shaymaa H. Kafi[1], Nabeel M. Mirza[2]*, Ali J. Karam[3], Anwar Al-Saleh[4], Ali Al-Zuky[1]

[1]*Department of Physics, College of Science, Mustansiriyah University, Baghdad, Iraq*
[2]*Department of Physics, College of Education, Mustansiriyah University, Baghdad, Iraq*
[3] *Department of Information Technology, College of Computer and Information Technology, University of Garmian, Iraq.*
[4] *Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq*

Received: 24/1/2022 Accepted: 13/8/2022 Published: 30/4/2023

Abstract

Optical Character Recognition (OCR) research includes computer vision, artificial intelligence, and pattern recognition. Character recognition has garnered a lot of attention in the last decade due to its broad variety of uses and applications, including multiple-choice test data, business documents (e.g., ID cards, bank notes, passports, etc.), and automatic number plate recognition. This paper introduces an automatic recognition system for printed numerals. The automatic reading system is based on extracting local statistical and geometrical features from the text image. Those features are represented by eight vectors extracted from each digit. Two of these features are local statistical (A, A_{th}), and six are local geometrical (P_1, P_2, P_3, P_4, P_5, and P_6). Thus, the database created consists of 1120 statistical and geometrical features. For the purpose of recognition, the features of the test image are compared with the features of all the images saved in the database depending on the value of the Minimum Distance (MD). All digits (0–9) were identified with 100% accuracy. The average computational time required to recognize a numeral at any font size is 0.06879 seconds.

Keywords: Feature Extraction; Numerals Recognition; Segmentation; Thinning.

نظام التعرف التلقائي على الأرقام باستعمال الميزات الإحصائية والهندسية الموقعية

شيماء حسن[1]، نبيل مبارك [2]*، علي جبار[3]، أنوار الصالح[4]، علي عبد داود الزكي[1]

[1]قسم الفيزياء، كلية العلوم، الجامعة المستنصرية، بغداد، العراق
[2]قسم الفيزياء، كلية التربية، الجامعة المستنصرية، بغداد، العراق
[3]كلية الحاسوب وتكنولوجيا المعلومات، جامعة گرميان، كلار، العراق
[4]قسم الحاسبات، كلية العلوم، الجامعة المستنصرية، بغداد، العراق

الخلاصة

تشمل أبحاث التعرف الضوئي على الحروف (OCR) كلٌ من الرؤية الحاسوبية، الذكاء الاصطناعي، والتعرف على الأنماط. حظي التعرف على الحروف بالكثير من الاهتمام في العقد الماضي نظراً لتنوعه الواسع

* Email: nabeel.mirza@uomustansiriyah.edu.iq

في الاستعمالات والتطبيقات، بما في ذلك بيانات اختبار الاختيار من متعدد، وثائق العمل (البطاقات التعريفية، الأوراق النقدية، جوازات السفر، وما إلى ذلك)، والتعرف التلقائي على لوحة الأرقام. تقدم هذه الورقة نظام التعرف التلقائي على الأرقام المطبوعة. يعتمد نظام القراءة الأوتوماتيكي على استخراج الخصائص الإحصائية والهندسية المحلية من الصورة النصية هذه الميزات متمثلة بثمانية نواقل مستخرجة من كل رقم. اثنان من هذه الميزات إحصائية موقعية (A, A th) ، وستة ميزات هندسية موقعية (P_6، P_5، P_4، P_3، P_2، P_1). وبذلك، تتكون قاعدة البيانات التي تم إنشاؤها من 1120 ميزة إحصائية وهندسية. لغرض التعرف، تتم مقارنة ميزات صورة الاختبار مع ميزات جميع الصور المحفوظة في قاعدة البيانات اعتمادًا على قيمة الحد الأدنى للمسافة (MD). تم تحديد جميع الأرقام (0–9) بدقة 100٪. معدل الوقت الحسابي المطلوب للتعرف على أي رقم وبأي حجم خط هو 0.06879 ثانية.

1. Introduction

Recognition of printed numerals is an active research field that accompanies technological development because of its applications in many of the necessary issues related to administrative leadership and digital governance, such as currency and passport recognition, electricity bills reading, water, zip code, commercial bar codes, and bank statements [1].

Documents containing printed numerals or characters are scanned, and then the OCR engine reads the images and converts them into ASCII data [2]. In general, the techniques of character recognition can be categorized into two types: off-line and on-line characters. The first type is divided into two sub-divisions: (a) Optical Character Recognition (OCR), and (b) Manual Character Recognition (MCR) [3]. Various methods of identification were reviewed, for example handwritten numerals [4, 5], printed and handwritten mixed kannada numerals [6], writings and scripts of Arabic [7], Arabic numbers [8], Vietnamese character recognition [9], and license plate recognition [10]. In general, all studies aim to obtain the best possible accuracy in identification, but feature extraction methodologies and classification methods vary.

People interested in the field of artificial intelligence often seek to find formulas or models with accurate capabilities in classifying and distinguishing printed numerals. Since numbers are associated with us in the majority of our daily dealings, many researchers are interested in the identification of printed numbers. This work aims to design a graphical user interface to recognize and categorize a set of digits (0–9) based on the features of structure. However, a problem arises when a digit is unable to be recognized due to differences in font size, bold, or font style. Taking into account the aforementioned problem, we developed a scheme to treat the problem related to the numerals automatic reading system. The ability of the developed technique to recognize all digits of varying sizes is predictable. Furthermore, this study will develop a framework capable of presenting an appropriate method for printed numerals' identification with the use of a simple and somewhat novel methodology that primarily relies on carefully extracting features and then building a database.

To achieve the objective of the research, we divided the image of the numeral into six local statistical parts and extracted six features (P_1, P_2, P_3, P_4, P_5, and P_6). Two geometrical features (A and A_{th}) that represent the area of the digit before and after the thinning process, respectively, are also used. So, they are proportional to the different font sizes of Arabic numerals. The database generated consists of 1120 features concerned with printed Arabic numerals for the Times New Roman font style. The creation of the database depends on the structure of the digits being thinned in the process of extracting features. The classification process depends on the minimum distance classifier.

The research for digit recognition began in the late eighties. Since then, different recognition methods have been proposed in order to achieve the aforementioned purpose. Several approaches were employed to increase recognition accuracy, such as the decision tree classifier [4], minimum distance classifier [3], and CNN architecture [11].

There are previous studies in which various techniques have been introduced and developed for the purpose of recognition and identifying printed and handwritten characters. Most of these studies depend on identifying the features of each symbol that are installed in a database to be used later to identify and classify unknown symbols. Some of these studies are: Zhou *et al*. [12] proposed a system for hand-printed numeral recognition based on two algorithms: one of them is hole detection and the other is contour concavity detection. Then they applied these algorithms to a practical automatic form reading system by Transym Optical Character Recognition (TOCR V1.0). TOCR reads simple characters automatically, handwritten numbers, or printed data. They collected test data while implementing the system in China as well as the CENPARMI database, and their proposal succeeded in handwriting recognition. Dhandra *et al.* [13] proposed a technique using Euler numbers that recognizes 17 different font styles with various sizes (8-72) points for recognition of English numerals. Their technique involved superimposing a small rectangle-bounding box over an isolated digital image and cropping it. After that, they found the pixels' exterior densities in each direction (bottom, top, left, and right). They then estimated the densities' ratios to the full area of the cropped numeral image and recorded the findings in a feature vector.

They tested 3200 numeral images, including 160 for training, and the classification accuracy of their proposed method was about 99.78%. Montazer *et al.* [14]; they proposed a holistic technique, including a neuro-fuzzy inference engine, to recognize the Farsi numeral. They combined the Mamdani fuzzy rule inference engine with a multilayer perceptron neural network to learn the features of different fonts of numerals (33 different Farsi fonts). The suggested strategy was able to identify Farsi numerical characters more comprehensively. Their proposal recognition rates were greater than 97%. Das *et al.* [15]; exploited a genetic algorithm (GA) to classify the CMATERdb handwritten Bangla numeral data set (6000 images of handwritten isolated Bangla numerals without restrictions). They presented a methodology for creating local regions of dynamically varying heights and widths. The GA is then used on the created localities to select the optimal localities in order to extract a set of optimal features. They used SVM for classification and achieved an accuracy rate of 97%. Hassanpour and Samadiani [16] have applied SOM neural networks in order to recognize 30 different font styles of different sizes ranging from (10-28) points for recognition of English numerals. They preprocessed the data by rotating and normalizing it, then split the image horizontally into two halves, and then extracted the features by whisking the data in the direction of row and column in each portion of the image. Even with varied fonts, these features are the same for a digit. They used a similarity scale to define the category of data in a neural network and the SOM neural network to recognize digits and classify them. Approximately 2000 numeral images were tested in the database. The recognition accuracy was around 99.55%. Jasim *et al.* [17] presented a system to recognize the Arabic handwritten numerals (0-9). Their system was called Handwritten Numeral Recognition Using Fuzzy Logic (HRUFL), which included practically (from preprocessing to recognition): converting an image to binary, denoising, thinning, isolate numerals, standardization, and feature extraction. Their system achieved a recognition rate of 94%. In other work, Damayanti *et al.* [18] adopted the idea of license plate recognition based on extracting the area feature for characters (A to Z) and numerals (0 to 9). They used the K-NN method to recognize the plate based on the value of features extracted. However, the computational time needed during the recognition processing is about 0.6 seconds in the area (10 x 10) pixels and 1.3 seconds in the area (20 x 20) pixels. The real size of the image is (300 x 720) pixels. As a result, the whole image recognition process would take longer. The accuracy of license plate recognition was about 68.57%, while the accuracy of character recognition on the license plate was approximately 92.72%. Many research domains have employed thinning as a preprocessing step, including numeral recognition [15], character recognition [19], image coding and biometrics [20], and fingerprint identification [19]. Where the result of thinning is utilized in determining the spin direction of the fingerprint, as in [20, 21].

The present work covers the following sections: Section 2 summarizes the methodology of creating a database and numeral recognition systems, which includes image acquisition, preprocessing, extraction of features, and numeral recognition. Section 3 introduces graphical user interface design. Section 4 covers the recognition results and analysis. Finally, the conclusion in section 5.

2. Methodology

Character recognition is a procedure that enables a reading machine or computer to understand printed or written characters in different languages. Whether they are numbers, letters, or symbols, they are converted into a formula that can be handled and modified by the computer. Generally, numeral recognition techniques are divided into two basic methods:
(i) methods that rely on extraction features.
(ii) OCR Method.

The features technique is based on the extraction of geometric or statistical features or both, while (OCR) technique depends on the calculation of either Euclidean distance or mean square error (MSE) between the original image of the symbol and images of symbols in the database to find the best match.

Figure 1 depicts a proposed automatic recognition system, each stage of which will be explained in detail in the subsequent paragraphs. The newly introduced recognition system consists of three main phases. The first phase is preprocessing, which includes (converting to binary, segmentation, thinning, and normalization). The second step is feature computation, followed by numeral recognition. This methodology will be able to identify the numerals in the document (distinguishing the category of digits according to arrangement).

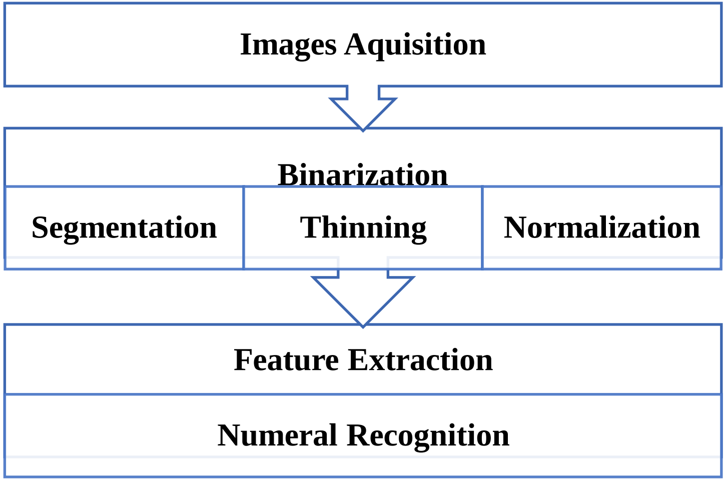


Figure 1: Scheme of numeral recognition system.

2.1. Images Acquisition

The Snipping Tool provided within the Windows Accessories environment and Microsoft Word Documents have been used in the image acquisition stage to obtain grayscale images for Arabic numerals (0-9) in JPEG format. The numerals are typed in a certain font size within the word document, the numbers are enclosed by a rectangle using the Snipping Tool, and then the image of the numerals is cropped and saved. Figure 2 displays some of the images that were collected. These images were acquired in the Times New Roman font. Image dimensions vary by font size. For example, 198 x 41, 228 x 46, 283 x 57, 390 x 95, 425 x 70, 431 x 111, and 626 x 136 of font size (16, 22, 28, 36, 42, 50, and 72) respectively. Thus, 70 images of varied sizes were used to create the database. The test data consists of 26 test images (100 numerals). The font sizes in the test data vary, as well as the number of digits contained in each image, which ranges from a single numerical order to nine numerical orders (1–9) as detailed in Table 1.

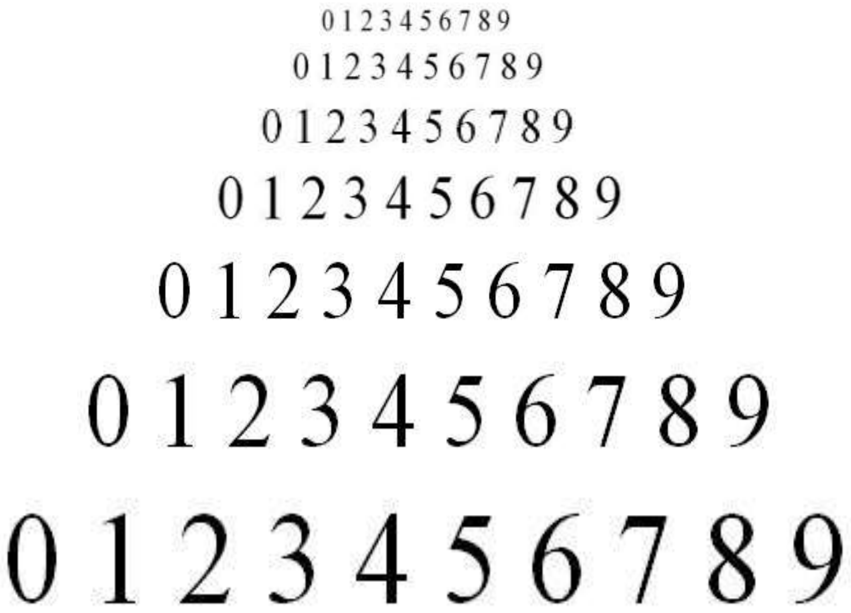


Figure 2: Samples of data set: Arabic numeral (0-9) with Times New Roman font style in different sizes.

Table 1 shows the number of test images versus the number of digits contained in each image

No. of images	No. of digits contained
7	1
4	2
3	3
2	4
3	5
2	6
1	7
2	8
2	9
Total 26	100

2.2. Pre-Processing

Because all numeral images are grayscale and highly clear, they are just converted into binary images for the purpose of segmenting and separating them from the backdrop. Grayscale images are converted to binary images using a threshold. This functionality is available in the MATLAB program. For segmentation, the MATLAB tool package called *RegionProps* is used. This command segments binary images into multiple targets, with the option of isolating each target by enclosing it with a rectangle. Naturally, the segmented numerals will have different dimensions depending on their geometric shape and size. As a result, we resized all rectangles (the segmented numerals) to be (100 x 60) pixels. Figure 3-a and 3-b show a sample of numerals binarized and segmented, respectively.

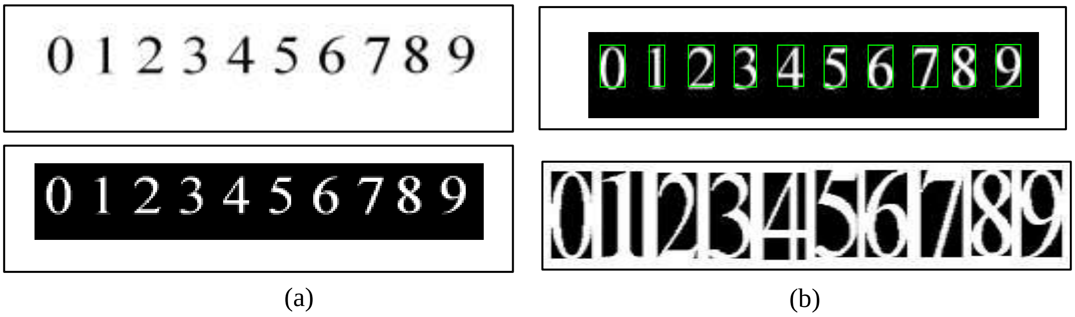


Figure 3: Result of (a) Binarization, and (b) Segmentation.

Thinning is a typical approach for identifying the geometric features of objects, such as numbers, by reducing their thickness. It is one of the morphological operations that apply to binary images, as the image is thinned through the command *bwmorph (BW, operation, Inf)* available within the MATLAB library tool, where Inf refers to repeating the process until the image stops changing. Figure 4 shows the results of the normalization experiment and thinning.

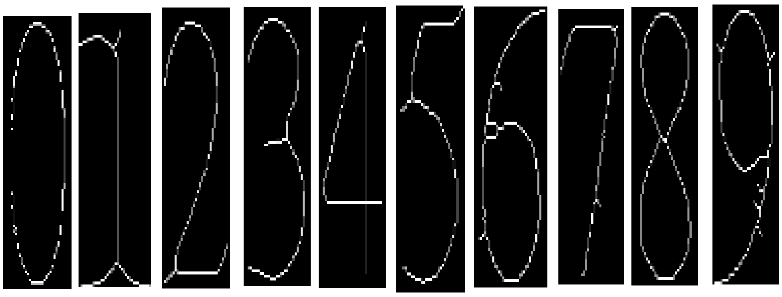


Figure 4: Result of the normalization and thinning.

2.3. Calculation of Features

Feature vectors are used to collect features that include as much useful information about an image as feasible. For the purpose of effective classification, we chose a set of non-traditional features from the data to distinguish samples that have similar shapes, such as 5, 6, and 9, as well as 0. For each segmented numeral, a vector of eight features has been presented in this study, V_f = [A, A_{th}, P_1, P_2, P_3, P_4, P_5, and P_6].

Where A is the numeral's area, which denotes the number of pixels; A_{th} is the numeral's thinning area (number of edge pixels); and (P_1-P_6) are the number of pixels in each of the six parts of the numeral as shown in Figure 5.

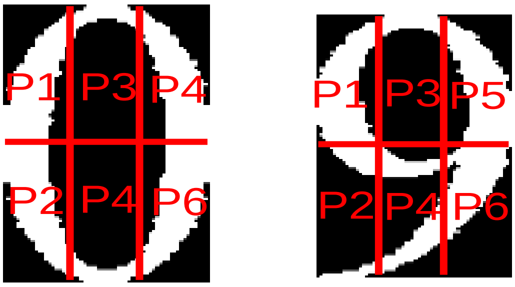


Figure 5: Determination of features for image's digit (0 and 9).

2.4. Characters Recognition

The identification of numerals in the text image is the final stage of the proposed system. In numeral recognition, the primary procedure is to convert the numeral image to a text file, which can be edited and used by any other program or application according to the need.

The Vector coefficients (V_C) for each character image are generated after getting a text image of numerals, which represents the test image. In this paper, the Minimum Distance (MD) technique is used, which calculates the difference between the generated vector and each vector in the database by Euclidean distance, whereas the minimum value of (MD) denotes the most careful identification of the numeral that has been tested. As a result, the numeral may be recognized.

The database (DB) consists of 1120 features and includes ten digits (0 - 9) of font styles (Times New Roman 70 images and Calibri 70 images), and each digit has eight (8) features as previously mentioned (V_C = A, A_{th}, P_1, P_2, P_3, P_4, P_5, and P_6). The following relationship has been used to normalize the feature values in the database to be within the range (0–1):

$$DB\ (i,j) = \left|\frac{V_C(i,j)_{max} - V_C(i,j)_{min}}{V_C(i,j)_{max} + V_C(i,j)_{min}}\right| \qquad (1)$$

V_C (i, j) $_{max}$ is the maximum value of (j^{th}) feature, V_C (i, j) $_{min}$ is the minimum value of (j^{th}) feature.

The recognition process is done by using eq. 2 [22]:

$$MD_{in} = \sum_{j=1}^{C} \left|\frac{V_C(i,j) - DB\ (i,j)}{V_C(i,j) + DB\ (i,j)}\right| \qquad (2)$$

Where: (MD_{in}) indicates the difference between the image of the entered number and those kept in the database, (C) denotes the total number of features, V (i, j) is the (j^{th}) feature of the (i^{th}) entered image, and DB (i, j) denotes the image feature (j) of the numerals that were kept in the database.

3. Graphical User Design

A Graphical User Interface (GUI) is an expandable graphical gateway that uses icons, signs, or other visual pointers to connect and interact with electronic gadgets. It is composed of a set of commands. In this paper, a GUI for digit recognition was designed in the

MATLAB program. Here are the steps in designing a graphical user interface (GUI) performance for our proposed system:

1. When running the program, the main screen will appear (see Figure 6-a), as all buttons are inactive.
2. When the select image upload button is pressed, a file selection dialog is displayed to select the text image file to be recognized (Figure 6-b).
3. Pressing the (process) button converts the text image into binary as shown in Figure 6-c.
4. The segmentation button segments the text image and displays the result (Figure 6-d).
5. OUT button implements the numeral recognition and displays the result (Figure 6-e).
6. The reset button deletes the previous steps and initializes the recognition system, i.e., returns to the main screen as shown in Figure 6-f.

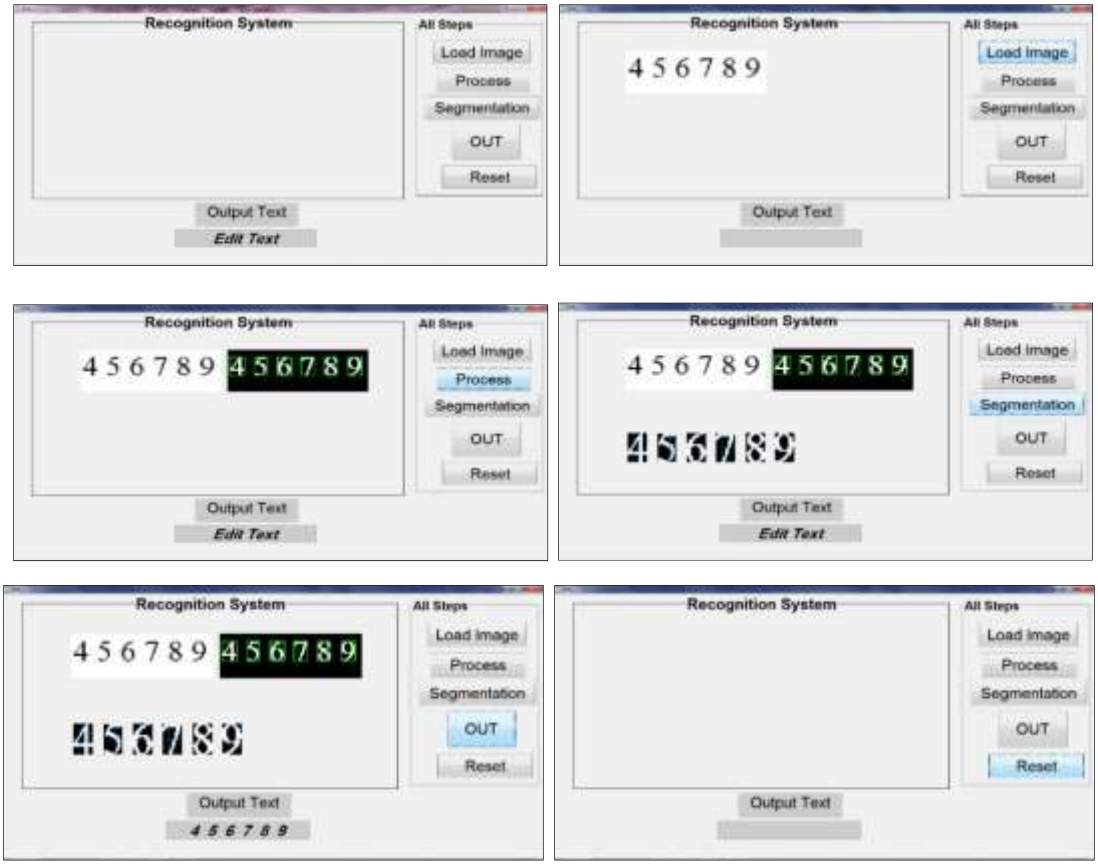


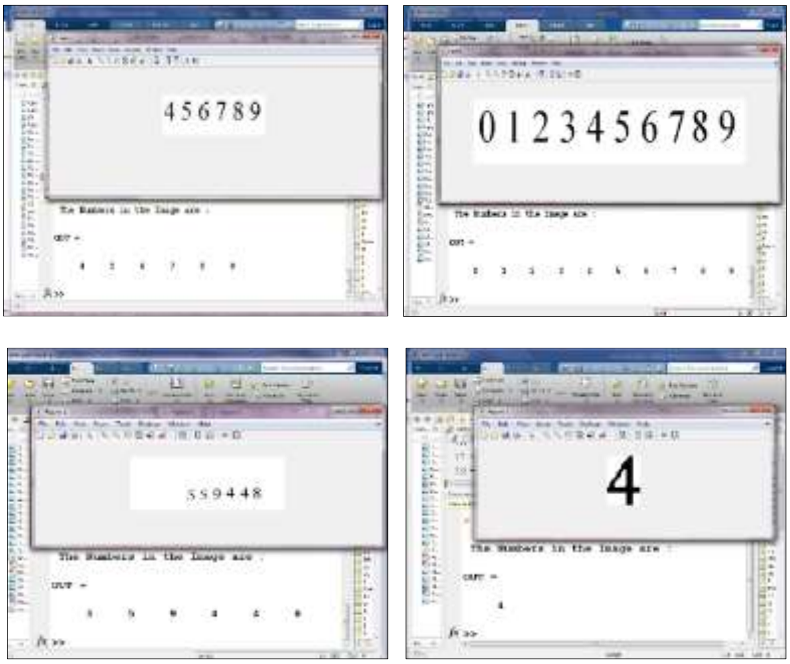
Figure 6: Applying of GUI window.

4. Experimental Results

For the numeral recognition model, a database of 1120 features was constructed based on (8) parameters for feature extraction. In this paper, many images of characters were tested; these images differ from each other in terms of font size and the number of digits each image contains. To complete the numeral recognition system, several experimental steps were adopted during the research. Figure 8 depicts some of the preparation measures and tests performed on the implemented recognition system. Equation 1 was applied to test pictures that had multiple number orders in one image, such as (ones, tens, hundreds, .. , billions), and in different font sizes. According to the test and training outcomes of all text images, our system can identify and differentiate printed numerals regardless of font size or style, where all the numerals in the text images are correctly identified. Figure 7 illustrates some samples of the program running for various entered images, as well as how the recognition system's output is saved in a vector named OUT. The suggested approach is capable of determining all digits with more accuracy and in less time than the method used by [18]. The average processing time required to recognize numerals is shown in Table 2.

Table 2 The average computational time required to recognize numerals at various font sizes

Font Size (point)	Average time for numeral recognition (sec)
16	0.057439
22	0.060617
28	0.061654
36	0.067111
42	0.074026
50	0.078015
72	0.082666



```
The Numbers in the Image are :

OUT =

     4     5     6     7     8     9
```

```
The Numbers in the Image are :

OUT =

     0     1     2     3     4     5     6     7     8     9
```

```
The Numbers in the Image are :

OUT =

     5     5     9     4     4     8
```

```
The Numbers in the Image are :

OUT =

     4
```

Figure 7: Implementing GUI window.

The categories of true positive (TP) and true negative (TN) are both accurate classifications. When a sample that is actually negative is forecasted as positive, this is known as a "False Positive" (FP). A false negative (FN) occurs when a positive sample is incorrectly anticipated to be negative.

The metric of accuracy used to evaluate the suggested approach is precision, which relates to the results of data checking in terms of correctly signalling a test result (the existence of a true condition) and incorrectly indicating a test result (the presence of a false condition). The following formula can be used to represent it:

$$\text{precision} = \frac{true\ positive\ (tp)}{true\ positive\ (tp) + false\ positive\ (fp)} \qquad (3)$$

Table 3: Results of Test Numerals Recognition

Digit Test	Mind	Font-Size	Check	Digit in DB	Digit Test	Mind	Font-Size	Check	Digit in DB
0	0.05	TN-50	True	0	1	0	TN-22	True	1
	0.11	TN-72	True	0		0.63	TN-36	True	1
	0.17	TN-42	True	0		0.64	TN-28	True	1
	0.25	TN-36	True	0		0.87	TN-72	True	1
	0.45	TN-28	True	0		0.88	TN-42	True	1
	0.51	CB-36	True	0		1.03	TN-50	True	1
	0.63	CB-28	True	0		1.37	TN-16	True	1
	0.75	CB-72	True	0		1.42	TN-16	True	1
	0.89	CB-50	True	0		1.67	TN-16	True	1
	0.94	TN-36	True	0		1.78	TN-22	True	1
2	0.01	TN-72	True	2	3	0	TN-16	True	3
	0.19	TN-50	True	2		0.36	TN-22	True	3
	0.25	TN-42	True	2		0.37	TN-36	True	3
	0.34	TN-36	True	2		0.51	CB-28	True	3
	0.4	TN-22	True	2		0.61	CB-72	True	3
	0.42	TN-28	True	2		0.62	TN-28	True	3
	0.72	TN-72	True	2		0.67	TN-42	True	3
	0.74	TN-50	True	2		0.7	TN-50	True	3
	0.74	TN-42	True	2		0.72	CB-50	True	3
	0.77	TN-72	True	2		0.73	CB-72	True	3
4	0.08	TN-22	True	4	5	0.04	TN-22	True	5
	0.45	TN-16	True	4		0.21	TN-50	True	5
	0.49	CB-22	True	4		0.22	TN-72	True	5
	0.6	CB-50	True	4		0.28	TN-36	True	5
	0.6	TN-42	True	4		0.32	TN-28	True	5
	0.61	CB-42	True	4		0.37	TN-42	True	5
	0.62	CB-16	True	4		0.83	TN-16	True	5
	0.71	CB-36	True	4		0.84	TN-42	True	5
	0.72	TN-50	True	4		0.94	CB-36	True	5
	0.72	TN-72	True	4		0.97	TN-72	True	5
6	0.12	TN-22	True	6	7	0.13	TN-22	True	7
	0.33	TN-42	True	6		0.31	TN-50	True	7
	0.46	TN-50	True	6		0.33	TN-36	True	7
	0.58	TN-72	True	6		0.34	TN-72	True	7
	0.6	TN-36	True	6		0.35	TN-28	True	7
	0.78	TN-28	True	6		0.37	TN-42	True	7
	0.98	TN-16	True	6		0.64	TN-16	True	7
	1.04	TN-42	True	6		0.77	CB-72	True	7
	1.05	TN-36	True	6		0.78	CB-42	True	7
	1.06	CB-36	True	6		0.78	CB-50	True	7
8	0.28	TN-16	True	8	9	0	TN-22	True	9
	0.46	CB-42	True	8		0.64	TN-16	True	9
	0.56	CB-22	True	8		0.84	TN-42	True	9
	0.59	CB-72	True	8		0.88	TN-50	True	9
	0.6	TN-36	True	8		0.88	TN-36	True	9
	0.65	CB-22	True	8		0.97	TN-72	True	9
	0.67	CB-50	True	8		1.05	TN-28	True	9
	0.7	CB-36	True	8		1.06	CB-50	True	9
	0.72	TN-72	True	8		1.06	CB-36	True	9
	0.73	CB-28	True	8		1.11	CB-28	True	9

The test data consists of 26 test images. The total number of numerals in these images is 100. For each test digit (0–9), the MD of the entered test characters is computed and then compared with those saved in the database to show the correctness and robustness of the proposed approach. Table 3 lists the 10 lowest MD values in increasing order. The font types (TN) and (CB) stand for Times New Roman and Calibri, respectively. In the examination field, true or false statements were employed to signify a match or a non-match.

Table 3 contains the experimental results. Correct recognition is generally indicated by the lowest MD value, i.e., the greatest match between entered image features and those saved in the database. However, in this study, the results of 10 x 10 numerals were exhibited to confirm the effectiveness of the suggested approach. All of the instances show that everything is in order. Table 4 compares the proposed method's results to those of other widely used recognition approaches.

Table 4: Comparison of the proposed method to other commonly used methods

Classification method	Type of data tested	Accuracy (%)	Computational time per digit (Sec)
Hermite Curves [24]	Printed English numerals	--	5.21
Decision Tree [23]	Handwritten English numerals	84.732	---
KNN [18]	Printed English numerals	92.72	0.616125 1.307945
Fuzzy Logic [17]	Arabic handwritten numerals	94	---
Neuro-fuzzy, and Neural Network [14]	Printed Farsi numeral	97	---
Genetic Algorithm, SVM [15]	handwritten Bangla numeral	97	---
SOM Neural Network [16]	Printed English numerals	99.55	---
Euler Number [25]	Printed English numerals	99.76	0.0135
Directional Density of Pixels [13]	Printed numerals	99.78	0.0160
Proposed Method	Printed Arabic numerals	100	0.06879

5. Conclusion

This paper presents a new technique for recognizing Arabic numerals printed on a document image. Firstly, the images are processed using various stages such as segmentation, thinning, normalization, and feature-vector calculation to extract features and then generate a database. A variety of images of numerals (0-9) have been acquired by the Snipping Tool Icon, which is available under Windows. These images are miscellaneous in terms of font styles (Calibri and Times New Roman), font size (16, 22, 28, 36, 42, 50, and 72), and the number of digits contained in each image.

A new approach has been taken to extract effective features from the Arabic numeral recognition system. The method for calculating features includes two stages: (a) thinning and normalization, and (b) vector-features. Both are described in section (2.3). In addition to the design of a Graphical User Interface (GUI) to display the identification result, which is detailed in section (3).

The minimum distance classifier has been used to differentiate numerals that represent the difference between the feature value of a digit within the image (entered) and those saved in the database. Whereas the least value of (MD) indicates the highest match for the test character, the numeral is recognized. The technique demonstrated a very hopeful result with 100% overall accuracy. The average time it takes to recognize each digit in any font size is 0.06879 seconds.

Despite the fact that the entered test character features (Time News Roman and Calibri) differ from those saved in the database (Time News Roman) only, the results show that 28 test numbers are identical to (Calibri), all of which were true.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to Mustansiriyah University (www.uomustansiriyah.edu.iq) in Baghdad, Iraq, for its assistance with this work.

References

[1] P. Jain, K. Taneja, and H. Taneja, "Which OCR toolset is good and why? A comparative study", *Kuwait Journal of Science*, vol. 48, no.2, 2021.

[2] V. Wu, R. Manmatha, and E. M. Riseman, "TextFinder: An Automatic System to Detect and Recognize Text In Images," *IEEE Transaction on Pattern Analysis and Machine Intelligence,* vol. 21, no.11, pp. 1224- 1229, 1999.

[3] N. M., Mirza, "Printed Arabic Characters Recognition Based on Minimum Distance Classifier Technique," *Iraqi Journal of Science*, vol. 59, no.2A, pp. 762-770, 2018.

[4] T. A. Assegie, and P. S. Nair, "Handwritten Digits Recognition with Decision Tree Classification: A Machine Learning Approach," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 9, no. 5, pp. 4446-4451, 2019.

[5] C. L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten Digit Recognition: Benchmarking of State-of-the-art Techniques," *Pattern Recognition* 36, pp. 2271-2285, 2003.

[6] G. G. Rajput, R. Horakeri, and S. Chandrakant, "Printed and Handwritten Mixed Kannada Numerals Recognition Using SVM," *International Journal on Computer Science and Engineering (IJCSE),* vol. 2, no. 5, pp. 1622-1626, 2010.

[7] H. Ramzi, and G. Catherine, "Arabic Text Recognition," *International Arab Journal of Information Technology*, vol. 1, no. 2, pp. 156-163, 2004.

[8] L., M. Ibrahim and H. H. Ali, "Use Artificial Neural Network Neococcontron in Distinguishing Handwritten Arabic Numerals," *The 1st Conference of Technical Information 22-23 Dec.*, vol. 1, no. 2, pp. 1-14, 2008.

[9] T. H. Phan, D. C. Tran, and M. F. Hassan, "Vietnamese Character Recognition Based on CNN Model with Reduced Character Classes," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 2, pp. 962-969, 2021.

[10] S. Du, M. Ibrahim, M. Shehata and W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311-325, 2013.

[11] P. Ahamed, S. Kundu, T. Khan, V. Bhateja, R. Sarkar, and A. F. Mollah, "Handwritten Arabic Numerals Recognition Using Convolutional Neural Network," *Journal of Ambient Intelligence and Humanized Computing,* vol. 11, no. 11, pp. 5445–5457, 2020.

[12] J. Zhou, Q. Gan, and C. Y. Suen, "A High Performance Hand-printed Numeral Recognition System with Verification Module," *Proceedings of 4th International Conference on Document Analysis and Recognition*, vol. 2, pp. 293-297, 1997.

[13] B. V. Dhandra, V. S. Malemath, H. Mallikarjun, and R. Hegadi, "Multi-font Numeral Recognition without Thinning based on Directional Density of Pixels," *IEEE,* vol. 1, pp. 157-160, 2006.

[14] G. A. Montazer, H. Q. Saremi, and V. Khatibi, "A Neuro-Fuzzy Inference Engine for Farsi Numeral Characters Recognition," *Expert Systems with Applications*, vol. 37, 6327–6337, 2010.

[15] N. Das, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri, and D. K. Basu, "A Genetic Algorithm Based Region Sampling for Selection of Local Features in Handwritten Digit Recognition Application," *Applied Soft Computing*, vol. 12, 1592–1606, 2012.

[16] H. Hassanpour, and N. Samadiani, "Recognition of Multi-Font English Numerals using SOM Neural Network," *International Journal of Computer Applications*, vol. 83, no. 10, pp. 37-43, 2014.

[17] M. K. Jasim, A. M. Al-Saleh, and A. Aljanaby, "A Fuzzy Logic Based Handwritten Numeral Recognition System," *International Journal of Computer Applications*, vol. 98, no. 4, pp. 37-41, 2013.

[18] F. Damayanti, S. Herawati, F. M. Ayu, and A. Rachmad, "Indonesian License Plate Recognition Based on Area Feature Extraction," *TELKOMNIKA*, vol. 17, no. 2, pp. 620-627, 2019.

[19] P. Yawalkar, and M. U. Kharat, "Effective Thinning Algorithm for Recognition of Hand Written Devnagri Compound Characters Using Neural Network", *International Journal of Applied Engineering Research*, vol. 13, No. 12, pp. 10539-10550, 2018.

[20] H. Chatbri, and K. Kameyama, "Using Scale Space Filtering to Make Thinning Algorithms Robust Against Noise in Sketch Images," *Pattern Recognition Letters*, vol. 42, pp. 1-10, 2014.

[21] I. G. Babatunde, A. B. Kayode, A. O. Charles, and O. Olatubosun, "Fingerprint Image Enhancement: Segmentation to Thinning," *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 1, pp. 15-24, 2012.

[22] H. Mohamad, S. A. Hashim, and A. H. Al-Saleh, "Recognize Printed Arabic Letter Using New Geometrical Features," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 3, pp. 1518-1524, 2019.

[23] T. Assegie, and P. Nair, "Handwritten Digits Recognition with Decision Tree Classification: A Machine Learning Approach," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 4446-4451, 2019.

[24] G. Lingaraju, C. Sujata, D. Prabhakar, and A. Shantarajappa, "Cognition and Recognition of Numerals using Hermite Curves," *In Proc. of 2nd National Conf. on Document Analysis and Recognition (NCDAR)*, pp. 139-144. 2003.

[25] N. Arjun, G. Navaneetha, G. Preethi, and T. Babu, "An Approach to Multi-font Numeral Recognition," *In TENCON 2007-2007 IEEE Region 10 Conference* (pp. 1-4).