



Feature Extraction in Six Blocks to Detect and Recognize English Numbers

Heba Kh. Abbas¹, Haidar J. Mohamad^{2*}

¹Department of Physics, College of Science for women, University of Baghdad, Baghdad, Iraq

²Department of Physics, College of Science, Mustansiriyah University, Baghdad, Iraq

Received: 31/3/2021

Accepted: 11/6/2021

Abstract

The Fuzzy Logic method was implemented to detect and recognize English numbers in this paper. The extracted features within this method make the detection easy and accurate. These features depend on the crossing point of two vertical lines with one horizontal line to be used from the Fuzzy logic method, as shown by the Matlab code in this study. The font types are Times New Roman, Arial, Calabria, Arabic, and Andalus with different font sizes of 10, 16, 22, 28, 36, 42, 50 and 72. These numbers are isolated automatically with the designed algorithm, for which the code is also presented. The number's image is tested with the Fuzzy algorithm depending on six-block properties only. Groups of regions (High, Medium, and Low) for each number showed unique behavior to recognize any number. Normalized Absolute Error (NAE) equation was used to evaluate the error percentage for the suggested algorithm. The lowest error was 0.001% compared with the real number. The data were checked by the support vector machine (SVM) algorithm to confirm the quality and the efficiency of the suggested method, where the matching was found to be 100% between the data of the suggested method and SVM. The six properties offer a new method to build a rule-based feature extraction technique in different applications and detect any text recognition with a low computational cost.

Keywords: Fuzzy algorithm, isolated number, recognition, Normalized Absolute Error, feature extraction.

استخراج خصائص في ستة مربعات لكشف وتمييز الأرقام الإنكليزية

هبة خضير عباس¹, حيدر جواد محمد^{2*}

¹قسم الفيزياء، كلية العلوم للبنات، جامعة بغداد، بغداد، العراق

²قسم الفيزياء، كلية العلوم، الجامعة المستنصرية، بغداد، العراق

الخلاصة

في هذا البحث تم تطبيق تقنية المنطق المضبب للكشف والتمييز عن الأرقام الإنكليزية من (0-9). عملية استخلاص الخصائص في هذه الطريقة تجعل عملية الكشف سهلة ودقيقة. حيث اعتمدت على تقسيم

صورة الرقم إلى ستة مناطق من خلال تقسيم الصورة إلى خطين عموديين مع خط أفقي واحد وتطبيقه بطريقة المنطق المضرب (البرنامج معروف في هذه الدراسة). تم اخذ الأرقام بخمسة خطوط مختلفة هي (Times , 28 , 22 , 16 , 10) بأحجام خطوط مختلفة (Andalus , Arabic , Calabria , Arial , New Roman , 36 , 42 , 50 , 72). يتم فصل هذه الأرقام تلقائيًا باستخدام الخوارزمية المصممة (البرنامج معروف في هذه الدراسة). الصورة الرقمية التي تم اختيارها باستخدام خوارزمية التضبيب تعتمد على خصائص ستة مناطق فقط. أظهرت مجموعات المناطق (عالية ومتوسطة ومنخفضة) لكل رقم سلوكًا فريدًا للتعرف على أي رقم. تستخدم معادلة الخطأ المطلق المعياري لتقييم نسبة الخطأ للخوارزمية المقترحة. وكان أقل خطأ 0.001% مقارنة مع الرقم الحقيقي. تم فحص البيانات بواسطة خوارزمية جهاز ناقل الدعم (SVM) لتأكيد جودة وكفاءة الطريقة المقترحة، حيث كانت المطابقة 100% بين بيانات الطريقة المقترحة و SVM. بينت النتائج أن الخصائص المستخلصة الست (p1,p2,p3,p4,p5,p6) طريقة جديدة لبناء تقنية استخلاص الخصائص القائمة على القواعد في مختلف التطبيقات واكتشاف أي التعرف على النص بتكلفة حسابية منخفضة.

1. Introduction

Recognizing a handwritten letter is an easy task for humans, but it may be problematic for a computer. The issue implies how a machine can be programmed to do that kind of task. Therefore, we need to express the solution with a traditional method, and then a smarter method needs to be established for this task. The creation of the Fuzzy Logic algorithm, the Genetic system, and the Neural Network method, etc., assisted people to build computer algorithms to choose the action, although at a basic level, that imitates the behavior of the organism.

The method of pattern recognition is known to recognized an input pattern information, therefore the output data represent the classes of the image [1]. Character recognition is one of the essential areas of pattern recognition. While it offers a clarification for automatically-processed large data, it also enhances human-machine communication. The key to character recognition is to turn a visual document into a textual one [2]. Several methods have been reported for pattern recognition. Fuzzy logic is used in some of these schemes. Most character recognition systems require pattern-based preprocessing operations [3-5]. The idea of the pattern features in the recognition method is removed and/or modified data within an image. Preprocessing is considered as an important step in classifying and identifying unfamiliar patterns [6].

The term feature extraction consists of two meanings; identification and collection of features [7]. The detection process aims to get features in the greatest extent and retain useful information about the image. The goal of the features collection is to decide the features key components, according to the classification task, to obtain an accurate classification [8]. The above concept indicates that the image information is expressed by the output of the feature detector. The features extraction is important step to extract the obvious features in the input data [9].

The transformation of handwriting into symbolic representation is referred to as the recognition process [9]. The purpose of handwriting recognition is the conversion of information of handwritten phrases, words, and characters to be identified. The identification of characters is a part of the issue of handwriting recognition. Arica and Yarman-Vural presented character recognition (CR), a method that was developed in three stages [10]. The handwriting character recognition (HCR) is considered as an important field of pattern recognition; it is referred to as the optical character recognition (OCR). The early phase of OCR was between 1900 and 1980. The beginning of OCR was said to have begun with the intention of designing blind reading machines.

Nowadays, the tremendous growth in automated systems, which make computers more friendly and easy to deal with, is needed to make a computer understands natural languages,

i.e. it can take orders from people through natural languages. The vision training of computers occupied a great attention to develop intelligent tools for specific needs, such as the monitoring tools for security purposes, depending on the Fuzzy logic method [11]. The development of Fuzzy logic in control applications, as in the inverted pendulum engineering approach, was presented without the need to do high computations [12]. Handwriting English Numbers Recognition (HENR) has been used in computer vision with the real-time applications [9]. There are three stages in HENR, namely pre-processing, classification, and feature recognition. The first stage is the pre-processing, which enhances the input image to move to the next step of classification, which generates recognized data set [13]. Feature extraction from a high-dimensional to a low-dimensional feature space used in the medical application [14]. Feature extraction defined as the method that retrieves the characteristics of handwritten characters to create Fuzzy rules. Feature extraction is a challenge when various persons write the same letter in different ways. For manuscript recognition system developers, the creation of a methodology that can lead to the identification of handwritten characters is still a hard task. The Fuzzy logic presents the opportunity for identifying handwritten characters with low computational costs by constructing a rule-based function extraction technique.

There was also some use of Fuzzy logic for character recognition, usually as an improvement to initial feature extraction algorithms. The input noise is separated to have simple adaptable structures from the characterization step [15]. However, since heterogeneity in a Fuzzy structure is managed by applying linguistically specified sets, some researches have shown that it is possible specific rule to execute the recognition task based on the membership of a given input to the different defined Fuzzy sets [16]. Fuzzy logic is inherently superior to processing precise results, which is natural for this use [17]. Data treatment is required so that it can be conveniently represented in the Fuzzy laws. Feature extraction is the key to numeral recognition when each number is distinctive in its own. It is vital to extract characteristics in such a way that it is easier to identify different numerals based on each number's properties [18].

In this paper, feature extraction is studied extensively by means of designing an algorithm in Matlab software using Fuzzy logic method. The introduced algorithm divides the input number's image into six blocks by two lines, one is vertical and one is horizontal. Data of these blocks are essential to recognize any number. The number of blocks is implemented automatically with the designed algorithm. There are six features extracted from these blocks which determine the recognition process. The Normalized Absolute Error equation is used to calculate the error percentage for all tested numbers. Moreover, the support vector machine is used to check the performance of the six properties employed to recognize any number.

2. Methodology

2.1 Number Recognition System

The function that is used to translate intelligible handwritten data in a computer is called the Handwritten Character Recognition (HCR), while the data is then processed by several automated systems. HCR can be usually divided into three steps: preprocessing, extraction of features, and classification. The preprocessing step creates an image with characters that can be used immediately by the extraction phase of the function, in which data redundancy is removed. The step of classification is performed to help in recognize characters or words. HCR is a complicated issue since changing in fonts and weights leads to variations of the same character. The variations in font styles and sizes make the task of identification challenging. Thus, the process of character recognition might not bring a better result.

There are many difficulties in making a machine do simple activity like recognize numerals in question with handwritten. The linear model of conventional computing is confronted by differences in writing style, accuracy, alignment, and stray marks in the vicinity of a number,

etc., which highlights the need for a computing system that can handle data more in the way the human brain does [1].

In this study, a handwritten numeral (HN) image is converted by using the threshold method to a binary image. The next step is the HN image isolation, which results in a character's image with a fixed size. Fuzzy logic assigns each character as a recognition stage. The results in this research are presented as printed numerals from 0 to 9. The number of patterns for these 10 numbers is 300, as shown in table 1. The ten different numbers were converted into 60×100 binary images. The number set that are used in this study are presented in Figure 1.

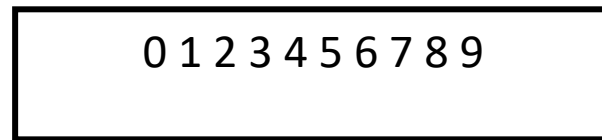


Figure 1-Numbers image details used in the study.

It is impossible to extract objects in the image by directly using Fuzzy logic because the data elements are large. Therefore, the feature extraction step is applied [6]. The most significant information is extracted by the feature extraction step to achieve a small set of new data [7].

The novel idea presented here relies on using one horizontal and two vertical lines in the tested image, as shown in Figure 2. The new feature is represented by the white pixel in the box of the crossing position between these lines. Figure 2 shows that each number has two junction points in the vertical lines. Therefore, the total number of boxes needed to extract the features is six. The tested images of all numbers have the same steps of the algorithm used to obtain number's features. There are two methods used in the Fuzzy inference system (FIS) for acquiring images.

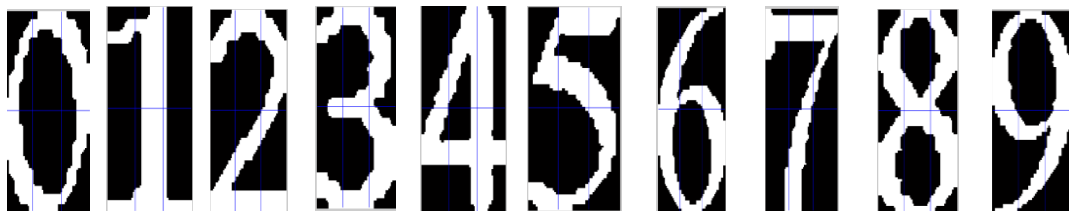


Figure 2-Feature extraction using 3 lines and dividing the image into six boxes.

2.2 Fuzzy Logic

The designer's knowledge must be transferred to the system, and this can be achieved by using Fuzzy logic. Fuzzy logic is employed to fill the gap between learned recognition and scientific human thinking [8]. Kim created a Fuzzy reasoning system to distinguish mainly confusing numerals [9]. The features of Fuzzy sets considered corner sharpness, hook shape, and relative spacing of the parallel lines by putting ten membership functions to separate the numbers.

In 1965, Lotfi Zadeh announced the Fuzzy logic method that was based on Fuzzy sets [10]. In a Fuzzy environment, the fuzzification mechanism converts all inputs into a single variable. This type of variable is described in a way that is simple to comprehend. Weak, decent, and excellent were once used to describe the degree of satisfaction with a snack bars service. In the input membership function, these are referred to as labels (mf). People cannot exactly describe those conditions, but they are present in real life. The general block diagram of a Fuzzy structure is shown in Figure 3. There are also labels for its parameters on the output side. The inference mechanism connects both the input and output data. The consequence of

Fuzzy logic is still in the Fuzzy variable, and the de-fuzzification procedure was used to restore it to its original state value.

There is a Fuzzy toolbox in MATLAB. This toolbox includes a FIS editor, which allows users to quickly build Fuzzy systems [11]. This editor is usually used to create input and output mf, write rules, and even evaluate the behavior of a particular input value. The outcome was saved as a *fis* file. The *read fis.m* and *evalfis.m* functions were used to integrate this design within the MATLAB code. To obtain the result, the user must first read the Fuzzy configuration (*read fis*) and then analyze the input signal (*eval fis*).

The selection feature does not have enough information about the original image, but it does need to have enough information to separate the different image classification groups [2, 3]. Figure 3 shows a block diagram of the function extractor for the classification scheme. Several repetitive operations in Calligraphy, i.e. the method of interpreting handwritten numerals, can be automated to improve the effectiveness of recognition systems. An optical scanner transforms every handwritten number into a digital file, which is then classified as one of the digits zero through nine by computer programs. Numerical recognition systems can speed up activities like reading tax returns, sorting inventory, and screening mail by reducing the need for human interaction. To do so, few steps need to be taken. The visual representation of handwritten numbers should first be captured by a recognition machine. Any preprocessing would be needed before attempting to classify the numerals.

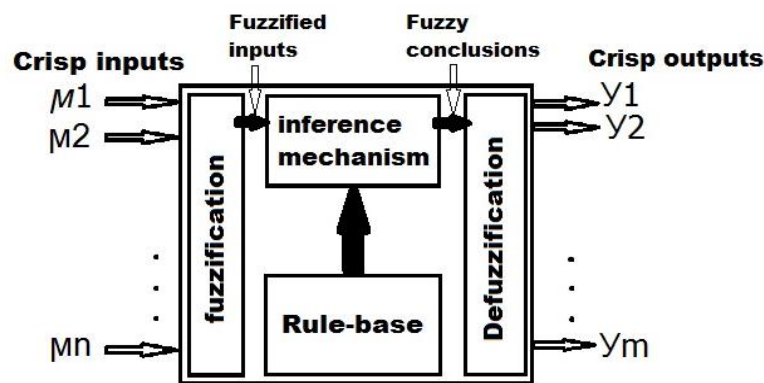


Figure 3-Fuzzy block diagram [17].

Although this process is simple to understand qualitatively, it is difficult to reduce to a few mathematical procedures. The issue arises due to the intrinsic variations in human handwriting. A useful recognition system must be resilient in the face of changes in height, shape, orientation, thickness, and so on. Closed-form mathematical models tend to be inadequate for this task due to the multiple possible representations of the same image. The problem presents certain difficulties that make pixel-by-pixel pattern matching impossible.

The edge of a number line, for example, can appear in two or three data slices depending on where the slices overlap (all the pixels along one column). Furthermore, small variations in printing cause number height and width to vary, and the imaged number may be blurred due to paper miss-feeding.

2.3 Support Vector Machine

For problems of two-group classification, a support vector machine (SVM) is a supervised machine learning model that uses classification algorithms. After providing an SVM model set of named training data for each category, they will categorize the text. Compared to other classifiers, such as logistic regression and decision trees, SVM has very high precision. It is renowned for handling nonlinear input spaces with its kernel trick. It is used in a number of

applications, such as facial identification, detection of interference, email classification, news stories and web sites, gene classification, and recognition of handwriting [19].

SVM is an exciting algorithm and the suggestions are reasonably straightforward. Using a hyperplane with the highest amount of margin, the classifier divides data points. That is why an SVM classifier is often regarded as a classifier that is discriminatory. SVM discovers an ideal hyperplane that helps to distinguish new data points. Help Vector Machines were usually called a classification approach, but can be seen with all forms of problems of classification and regression. It can accommodate different continuous and categorical variables easily. SVM builds a hyperplane to distinguish various groups in multidimensional space. SVM generates the ideal hyperplane that is used to mitigate an error in an iterative manner. The central principle of SVM is to find the optimal marginal hyperplane (MMH) for grouping the data.

The main goal is to isolate the given dataset as effectively as possible. The difference between the two closest points is known as the margin. The target is to find a hyperplane with the greatest possible margin between support vectors in the given dataset. SVM Classifiers forecast with fair precision and speed. They also use less memory since they only use a subset of training points during the decision process. SVM fits best with a clear separation margin and elevated dimensional space [20].

2.4 Image Preprocessing Algorithm

Before applying any recognition technique, known as pre-processing [13], most images need some manipulation. To boost the strength of the features that need to be extracted, preprocessing was performed. Preprocessing stages are often carried out to minimize noise in the data image. The main goal is to enhance the image data in a way that improves the other processing succeeding.

The input image contains all digits, 0 through 9, in any font size and type. After converting the input image to grayscale, the grayscale image is converted to binary using a threshold value. All pixels less than a threshold value have a value of 0 (black) in the output binary image, while all other pixels have a value of 1 (white). Among the most significant techniques for segmenting an image of a homogeneous object and a context with a different intensity level is to use a threshold.

We shall need an image that has been segmented and, if possible, morphologically filtered to remove object features. As a consequence, objects can be clearly marked and can be labelled and stored separately. After all of the image's objects have been numbered, the data for each object can be viewed as a binary image by assuming that the labelled object has a value of 1 and everything else has a value of 0. The labeling procedure begins with the definition of desired connectivity, followed by a scan of the image and the labeling of similar objects with the same symbol.

We shall standardize the image by reducing the size of the digit image to a size that will be easier to identify later once we have labelled the objects and have an image filled with object numbers. The features of interest were extracted from this picture. This approach alters the image data's proportions. To fit the new image dimensions, the image data is stretched or compressed as required. The image dimensions are determined by the new width of 60 pixels and the new height of 100 pixels.

Algorithm

The resized binary image can be coded in Matlab software as:

```
BW = im2bw(I,0.85);  
[h w] =size(BW);  
[L Ne] =bwlabel(BW);  
stats = regionprops(BW);  
nc=length(stats);
```

```

ii=0;
for i=1:nc
rectangle('Position', stats(i).BoundingBox,'EdgeColor','w','LineWidth',2)
end
for i=1:Ne
[r,c] = find(L==i);
y1=min(r);y2=max(r);x1=min(c);x2=max(c);dx=x2-x1;dy=y2-y1;
figure; imshow(BW);
n1=BW(y1:y2,x1:x2);
n1=imresize(n1,[100 60]);
figure; imshow(~n1);
name=strcat(int2str(ii), fname); imwrite (~n1, name);
ii=ii+1;
end

```

2.5 Fuzzy Program in MATLAB

The training of the system in Matlab software consists of a few steps. The first step after running the Matlab software is to design a FIS file that has six features according to this study. These features divided the image shown in Figure 7 into three regions named as low, mid, and high. These three regions are different for each number and presented in table 1. The second step is the rule editor. This step is coded by the user depending on the results in table 1; for instance, the membership for number zero in Arabic font is as (If (P1 is HIGH) and (P2 is HIGH) and (P3 is LOW) and (P4 is HIGH) and (P5 is HIGH) and (P6 is HIGH) then (output is ZERO)). The output data from these two steps is triangles plots saved as (.fis) file and the output shape is shown in Figure 4.

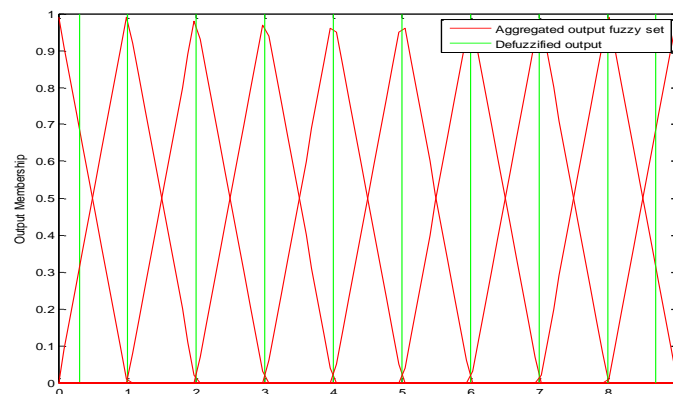


Figure 4-The output triangles of each number using the suggested method.

The recognition step is implemented by designing an algorithm in Matlab software code, as follows:

- 1- Read the input image that consists of digit numbers.
- 2- Convert it to binary with a threshold value of 0.85.
- 3- Use the *Regionprops* function to select the number area individually, then apply the morphological process (Image cleaning).
- 4- Find the area of the six parts in the number area, namely P1-P6, by dividing the input image into six regions.
- 5- Use the training file FIS, that was described earlier, to apply step 4 as input features for all font types.
- 6- Use the *evafis* function in step 5 to get the final results.

Normalized Absolute Error (NAE) criteria were used to check the error percentage for the Fuzzy output. The NAE values are promising, except the two values for zero and nine

numbers only. The output values of these criteria are listed in table 2. The NAE equation is written as:

$$NAE = \frac{|real\ value - Fuzzy\ value|}{maximum\ value} \times 100\% \tag{1}$$

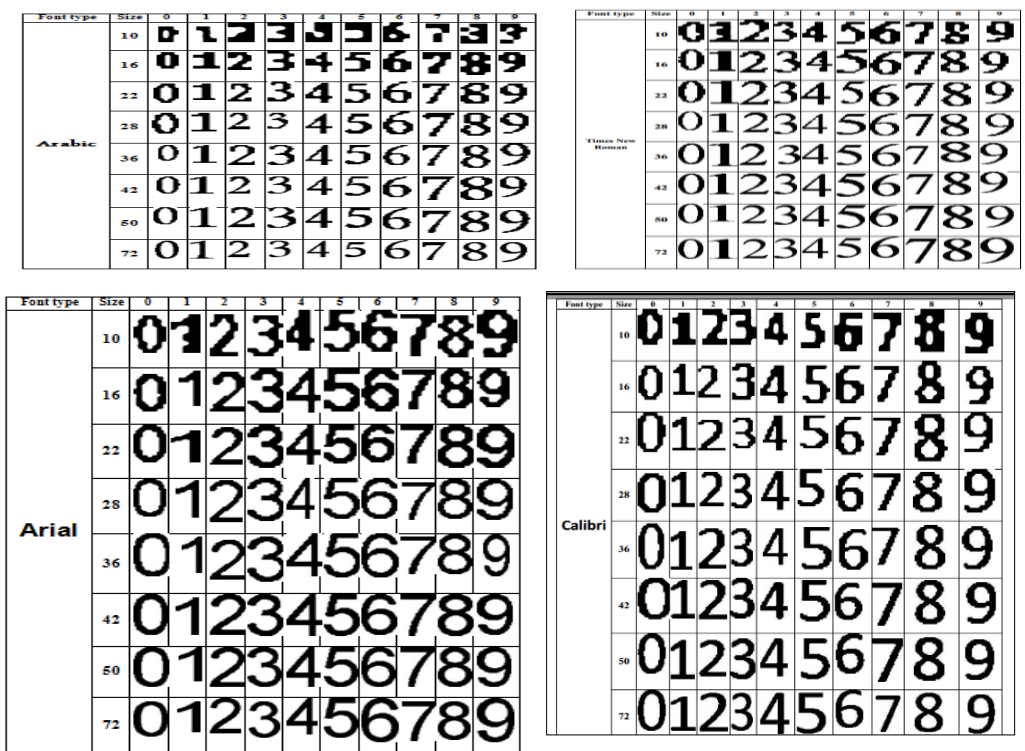
3. Results and Discussion

3.1 Experiment to extract geometric characteristics

The properties resulted from the figures of numbers from 0-9 were obtained depending on the division of the image into two vertical lines and one horizontal line. Every part within the six elements is named as P, therefore there are six boxes named from P1 to P6. These properties represent the number of edges within the square. The algorithm code in Matlab software is written as:

```
BW = im2bw(I,0.85);
stat1 = regionprops(BW);
A=stat1.Area;
[r c]=size(BW);
P1=sum(sum(BW(1:round(r/2) , 1:round(c/3) )));
P2=sum(sum(BW(round(r/2):end , 1:round(c/3) )));
P3=sum(sum(BW(1:round(r/2) , round(c/3):round(2*c/3) )));
P4=sum(sum(BW(round(r/2):end , round(c/3):round(2*c/3) )));
P5=sum(sum(BW(1:round(r/2) , round(2*c/3):end )));
P6=sum(sum(BW(round(r/2):end , round(2*c/3):end )));
figure(5);imshow(BW);
X=[1,c];Y=[round(r/2),round(r/2)];line(X,Y);
X=[round(c/3), round(c/3)]; Y=[1,100]; line(X,Y);
X=[round(2*c/3), round(2*c/3)]; Y=[1,100]; line(X,Y);
P=[P1 P2 P3 P4 P5 P6];
```

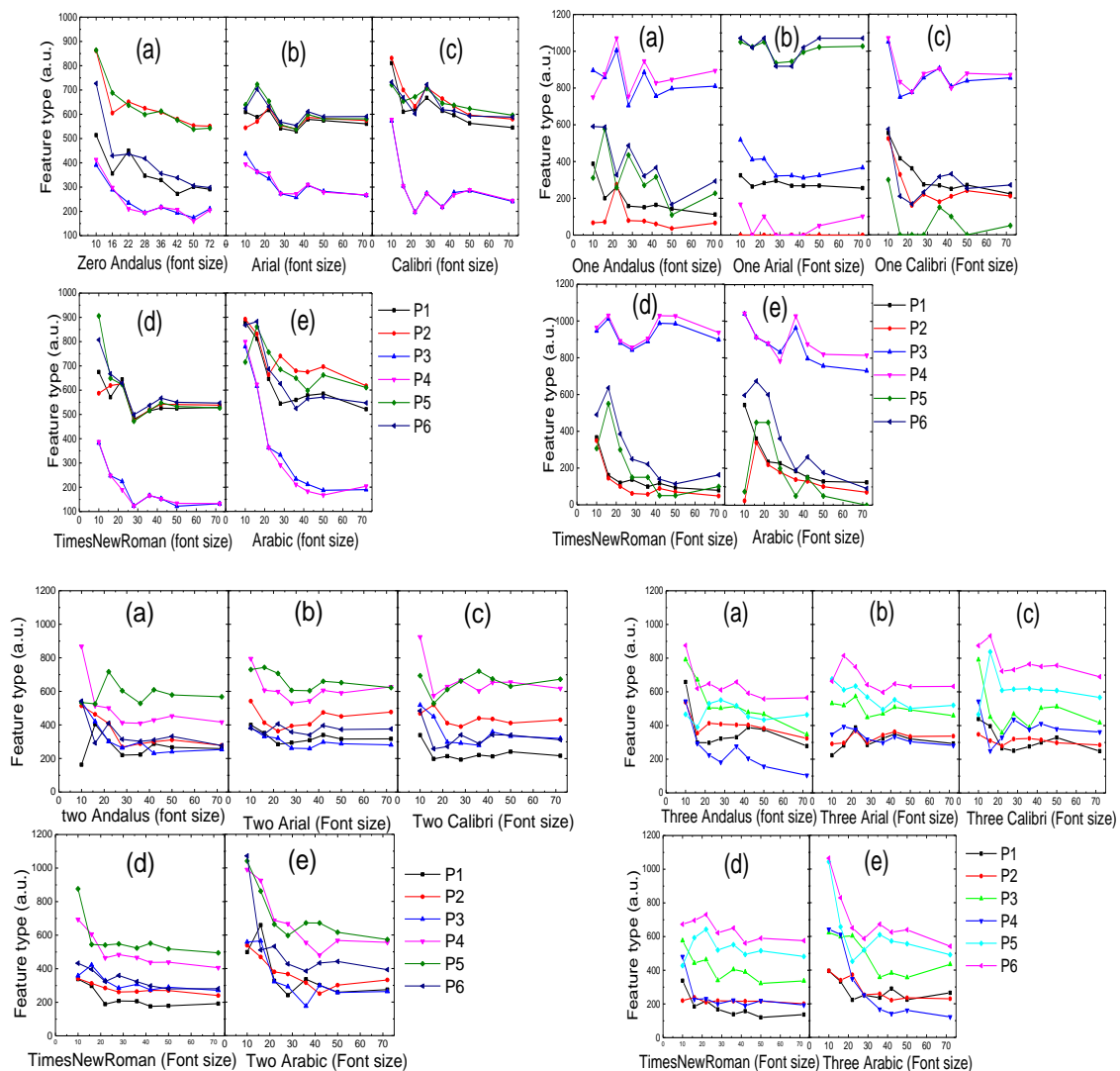
The images of numbers with all font sizes and font types are presented in figure 5. These images are resized and isolated with the presented algorithm to be used within this study.



Font type	Size	0	1	2	3	4	5	6	7	8	9
Andalus	10	0	1	2	3	4	5	6	7	8	9
	16	0	1	2	3	4	5	6	7	8	9
	22	0	1	2	3	4	5	6	7	8	9
	28	0	1	2	3	4	5	6	7	8	9
	36	0	1	2	3	4	5	6	7	8	9
	42	0	1	2	3	4	5	6	7	8	9
	50	0	1	2	3	4	5	6	7	8	9
	72	0	1	2	3	4	5	6	7	8	9

Figure 5-The input image numbers with different font size and font type to be recognize in Fuzzy algorithm.

Figure 6 shows the properties P1-P6 for all font sizes and font types resulted from the previous algorithm code. This is the first step to train the Fuzzy algorithm to recognize the numbers.



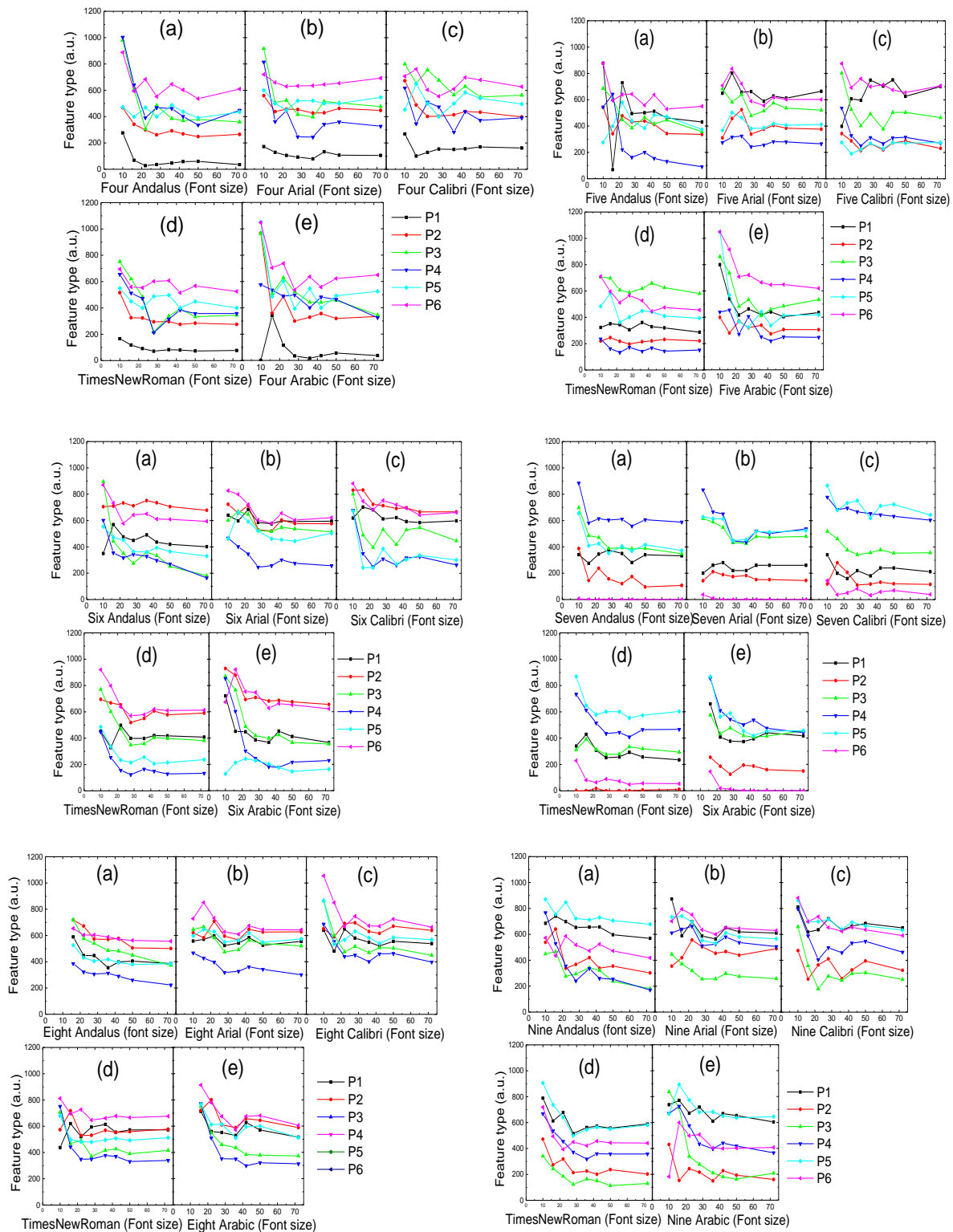


Figure 6-The six properties of each font size and font type of input image numbers 0-9.

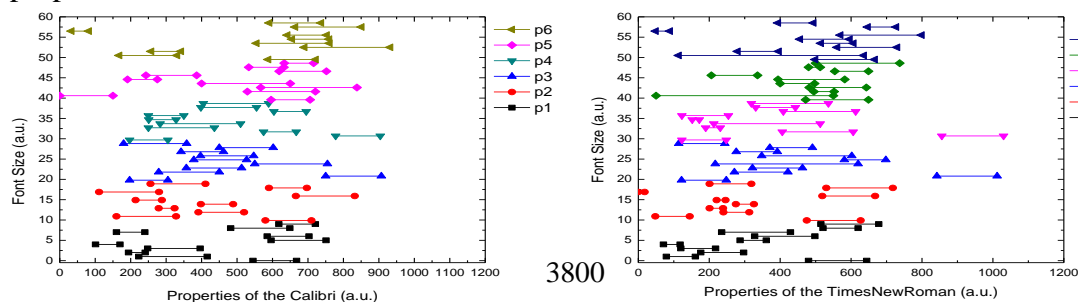
The minimum and maximum values were set for each property and each number of digits (0-9). These values are plotted in figure 7 and indexed in table 1 to obtain groups of regions (High, Medium, and Low). These regions are used in the FIS file for the training system. The

FIS file is then used to recognize any number with different size and font type. We find that there is a unique behavior for each number by dividing the properties into regions.

Table 1-Region properties for each number and for tested font types.

Font type	P	0	1	2	3	4	5	6	7	8	9
Arabic	P1	H	L	-	L	L	M	-	M	H	H
	P2	H	L	M	M	M	M	H	L	H	M
	P3	L	H	-	M	-	M	M	-	M	L
	P4	H	M	L	-	L	L	M	M	M	M
	P5	H	-	H	H	-	-	L	H	H	H
	P6	H	-	M	H	H	-	H	L	H	-
Andalus	P1	H	L	-	L	L	M	-	M	H	H
	P2	H	L	M	M	L	M	H	L	H	M
	P3	L	H	-	M	-	M	M	-	M	L
	P4	L	H	M	L	-	L	L	M	M	M
	P5	H	-	H	H	-	-	L	H	H	H
	P6	H	-	M	H	H	-	H	L	H	-
Times	P1	H	L	-	L	L	M	M	M	H	H
	P2	H	L	M	-	M	L	H	L	H	-
	P3	L	H	M	-	-	H	-	M	M	L
	P4	L	H	M	L	M	L	L	M	M	M
	P5	H	-	H	H	H	H	M	H	H	H
	P6	H	-	M	H	H	H	H	L	H	M
Calibri	P1	H	M	L	M	L	H	H	L	H	H
	P2	H	L	M	L	M	L	H	L	H	-
	P3	L	H	M	M	H	M	M	M	H	L
	P4	L	H	H	M	-	L	L	H	M	M
	P5	H	L	H	H	M	L	L	H	H	H
	P6	H	L	L	H	H	H	H	L	H	H
Arial	P1	H	L	M	M	L	H	H	L	H	H
	P2	H	L	M	M	M	M	H	L	H	-
	P3	M	M	M	H	H	H	H	H	H	M
	P4	M	L	H	M	M	M	M	H	M	H
	P5	H	H	H	-	M	M	-	-	H	H
	P6	H	H	M	H	H	H	H	L	H	H

The properties of the six regions for each number are presented as P1-P6. These properties are useful to recognize any number with the Fuzzy method. The minimum and maximum values of each property are plotted in figure 7. Figure 7 shows each font types with all number properties. Each property has ranges that are different between numbers and font type. Properties ranges are vital in the coded stage and, especially, in the training step. Properties ranges are obvious in Table (1) which shows the high, medium, or low range depending on its value in figure 6. The number sequence, in figure 7, starts from 0 at the bottom and ends with 9 at the top, as shown with the Arial font type. This style is also applied for all font types and all properties.



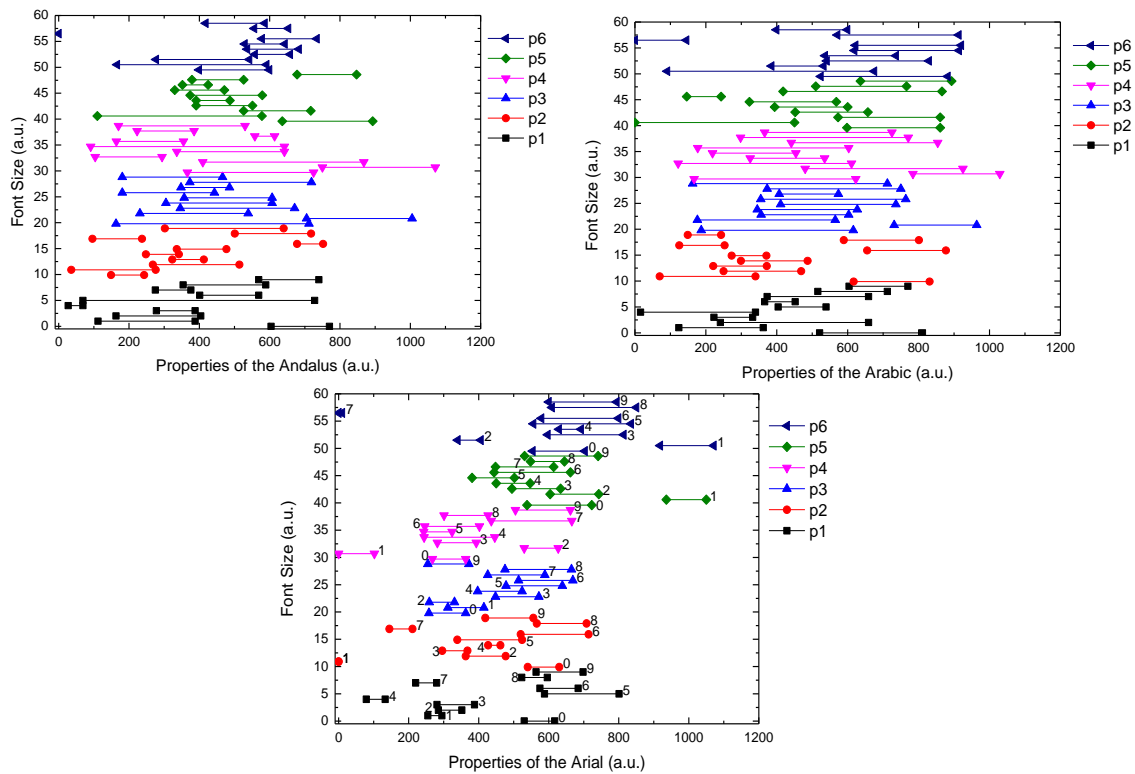


Figure 7-The minimum and maximum values for each property of digits (0-9). These values divided the property region into (High, Medium, and Low) for all font types. Arial font type is labelled only for clarification.

Table 2-The number value with the Fuzzy algorithm value and the error percentage

Number	Fuzzy output	NAE Error %
0	0.3044	3.38
1	1.0007	0.0077
2	2.0005	0.0055
3	3.0003	0.0033
4	4.0001	0.0011
5	4.9999	0.0011
6	5.9999	0.0033
7	6.9999	0.0055
8	7.9993	0.0077
9	8.6961	3.376

The results of the suggested method are compared with the SVM algorithm output. Figure 8 shows the matching in the output by testing the numbers with different types and font sizes. It is encouraging to notice the matching in the outputs for both methods. The SVM algorithm is not presented within this paper because it is well known and presented online [21].

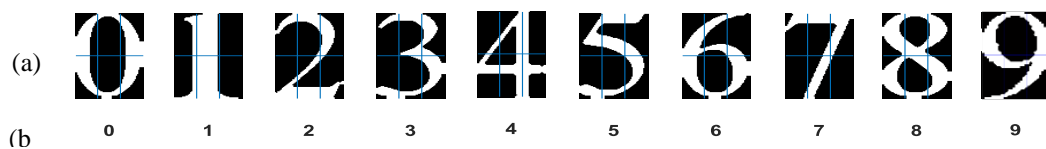


Figure 8-The output of the SVM for the Times New Roman font type with size 72 (a) the suggested method, and (b) the SVM output.

4. Conclusions

The Fuzzy method is used to recognize numbers with different font sizes and font types. The suggested algorithm with the presented Matlab code is the fundamental principle to identify any number. The input number's image has the tested numbers; each numbers is isolated as a single image with a new dimension of 60 x100 pixel. The binary image is divided into six boxes automatically to use new features depending on each box. These data are used to train the Fuzzy method with the presented codes. The number of features is only six according to the six sub-boxes, and it can be increased to nine or twelve, which may increase accuracy. However, these six feature (P1-P6) are enough to recognize any number with any font type and size. The normalized absolute error is used as a criterion, which showed good matching between the real number and the resulted recognition from the Fuzzy logic method. The SVM algorithm shows 100% matching with the proposed six properties.

References

- [1] A. Al-Hmouz, G. Latif, J. Alghazo, and R. Al-Hmouz, "Enhanced Numeral Recognition for Handwritten Multi-language Numerals Using Fuzzy Set-Based Decision Mechanism," *International Journal of Machine Learning and Computing*, vol. 10, pp. 99-107, 01/26 2020, doi: 10.18178/ijmlc.2020.10.1.905.
- [2] G. Katiyar and S. Mehruz, "A hybrid recognition system for off-line handwritten characters," *Springerplus*, vol. 5, 2016.
- [3] L. Sadouk, T. Gadi, and E. Essoufi, "Handwritten tiffinagh character recognition using deep learning architectures," in *IML '17*, 2017.
- [4] M. Kadhm and A. Karim, "Offline Isolated Arabic Handwriting Character Recognition System Based on SVM," *International Arab Journal of Information Technology*, vol. 16, pp. 467-472, 05/01 2019.
- [5] M. Zare, M. Jampour, A. S. Arezoomand, and M. Sabouri, "Handwritten Recognition based on Hand Gesture Recognition using Deterministic Finite Automata and Fuzzy Logic," in *2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, 6-7 March 2019 2019, pp. 93-99, doi: 10.1109/PRIA.2019.8786038.
- [6] K. V. Reddy, D. R. Rao, and K. Rajesh, "Hand Written Character Detection by Using Fuzzy Logic Techniques," 2013.
- [7] E. Elbaşı, "Fuzzy Logic-Based Scenario Recognition from Video Sequences," *Journal of Applied Research and Technology*, vol. 11, no. 5, pp. 702-707, 2013/10/01/ 2013, doi: [https://doi.org/10.1016/S1665-6423\(13\)71578-5](https://doi.org/10.1016/S1665-6423(13)71578-5).
- [8] A. Jain, D. Tayal, and N. Sehgal, "Control of Non-Linear Inverted Pendulum using Fuzzy Logic Controller," *International Journal of Computer Applications*, vol. 69, 05/01 2013, doi: 10.5120/12141-8278.
- [9] N. A. Kotak and A. K. Roy, "An accelerometer based handwriting recognition of English alphabets using basic strokes," in *TENCON 2017 - 2017 IEEE Region 10 Conference*, 5-8 Nov. 2017 2017, pp. 111-116, doi: 10.1109/TENCON.2017.8227846.
- [10] G. Katiyar and S. Mehruz, "A hybrid recognition system for off-line handwritten characters," (in eng), *Springerplus*, vol. 5, pp. 357-357, 2016, doi: 10.1186/s40064-016-1775-7.
- [11] J. Zhang, Z. Deng, K. Choi, and S. Wang, "Data-Driven Elastic Fuzzy Logic System Modeling: Constructing a Concise System With Human-Like Inference Mechanism," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 2160-2173, 2018, doi: 10.1109/TFUZZ.2017.2767025.
- [12] N. R. Soora and P. S. Deshpande, "Review of Feature Extraction Techniques for Character Recognition," *IETE Journal of Research*, vol. 64, no. 2, pp. 280-295, 2018/03/04 2018, doi: 10.1080/03772063.2017.1351323.
- [13] M. Jasim, A. Al-Saleh, and A. Aljanaby, "A Fuzzy Based Feature Extraction Approach for Handwritten Characters," *International Journal of Computer Science Issues (IJCSI)*, vol. 10, pp. 208-215, 07/04 2013.

- [14] Heba Kh. Abbas , Anwar H. Mahdy , Ali A. Al Zuky , Sally F. Ahmad, "Analytical Study of Medical Image Combination Techniques", *Iraqi Journal of Science*, 2018, Vol. 59, No.3B, pp: 1538-1547 DOI:10.24996/ij.s.2018.59.3B.20.
- [15] K. Manoj, M. Lata, and S. Gyan, "A Survey in Fuzzy Logic: An Introduction," *International Journal for Scientific Research & Development*, vol. 3, no. 6, 2014.
- [16] N. Kaja, A. Shaout, and D. Ma, "Fuzzy Based Threat Assessment Model (FTAM)," in *2019 International Arab Conference on Information Technology (ACIT)*, 3-5 Dec. 2019 2019, pp. 144-149, doi: 10.1109/ACIT47987.2019.8991129.
- [17] Heba Kh. Abbas, Anwar H. Al-Saleh, Ali A. Al-Zuky, "Modelling vision angles of optical camera zoom using image processing algorithm", *IOP Conference Series: Materials Science and Engineering*, 2019, 571 012117 DOI: 10.1088/1757-899X/571/1/012117.
- [18] A. A. Mohammed, A. K. Abdul-Hassan, and B. S. Mahdi, "Authentication System Based on Hand Writing Recognition," in *2019 2nd Scientific Conference of Computer Sciences (SCCS)*, 27-28 March 2019 2019, pp. 138-142, doi: 10.1109/SCCS.2019.8852594.
- [19] K.-X. Han, W. Chien, C.-C. Chiu, and Y.-T. Cheng, "Application of Support Vector Machine (SVM) in the Sentiment Analysis of Twitter Data Set," *Applied Sciences*, vol. 10, no. 3, p. 1125, 2020. [Online]. Available: <https://www.mdpi.com/2076-3417/10/3/1125>.
- [20] Heba Khudhair Abbas , Sally Fawzi Ahmed , Anwar Haasan Mahdy; "Quantitative Analysis based on Supervised Classification of Medical Image Fusion Techniques", *Iraqi Journal of Science*, 2017, Vol. 58, No. 3B, pp: 1546-1554 DOI: 10.24996/ ijs.2017.58.3B.19.
- [21] Heba Khudhair Abbas, Farah Faris, Sale Sami, Al Zahraa Fadel;" Adopting Image Integration Techniques to Simulate Satellite Images", *Iraqi Journal of Science*, 2020, vol. 61, no. 12, pp: 3446-3456 DOI: 10.24996/ij.s.2020.61.12.32..