Hyper Recognition Techniques for English Digits Using Statistical Analysis of Nodes and Fuzzy Logic for Pattern Recognition

Majdi Salameh[1] and Amer Abu Salem[2]
[1]University of Jeddah, Information Department, KSA, Jeddah, (Main Campus)
[2]Zarka University, Computer Department, Zarka, Jordan
[1]masalameh@uj.edu.sa, [2]abusalem@zu.edu.jo

***Abstract*– English optical digits recognition has lately focused on [1], as it is valuable in a many fields. This proposed paper offers an enhanced exist Arabic digits recognition using node and analysis techniques [2]. The paper offerings methods for improving English typewritten digits recognition rate [3]. First one is node method that estimates number of ends and conjunction nodes of the digit given shape. The second method pattern recognition using fuzzy logic by analyzing each stroke from the shape, and classifies it into the numbers categories. The stages are going to be done by the given methods, to recognize English digit, each come out with its own result and later compounds these result to acquire the concluding solution and statistical analysis. There will be several steps in the recognition system, initial with the image processing [4], then feature extraction after that classification step. The step of image processing [5] contains image reading, and thinning the shape [6], [7]. Feature extraction determine end and conjunction nodes number using nodes method, and character that specify curves, lines, and circles which form shapes, and a number for the position of end nodes depends to conjunction nodes in ambiguous cases such as 6, 9. Recognition includes compound between two vectors. The proposed technique was implemented and tested on set of 70 digits and the experimental results contribute high recognition rate for some fonts and low for other fonts. The reason of the low recognition rate is irregularity of some fonts, or weakening for one of the methods. Over all recognition rate for digits were 51%, and after enhancing thinning and conjunction/ending points techniques the recognition rate raised up to 74%. The dataset contains multi font types of English digits from 0 to 9. According to end conjunction node recognition rate is improved by using shape classification before end conjunction node classification.**

***Keywords*– English Digit Recognition, English Number Recognition, Node Method, Statistical Analysis, Fuzzy Logic, Pattern Recognition and Type Written Digits Recognition**

I. INTRODUCTION

English language is widely used, in all over the world. English language can mainly be under two categories: English characters (a-z, A-Z) and numbers (0 to 9). English are also two types: hand writing and type written. The improving of English digits recognition [3] can enhance the interaction between the computer and human [8]. The easier and faster the recognitions perform, the more applications and benefits could be done, many methods were done on digits recognition depending on type of feature extract. There were many researches recognizing handwritten using different methods and techniques such as hidden Markova model [9], fuzzy logic [10], NNW [11] and hand written numeral recognition using fuzzy logic [12] which compos digits into lines, curves and circle

Also another research decomposed based on detection of set of feature points such as ending, conjunction points [2]. This paper focus on English type written digits using hybrid of both classification method and detecting of set of feature points. Other studies used digits edges [13].

II. THE PROPOSED TECHNIQUE

This proposal discusses the methodology of typewritten digits recognition technique, which will uses proposed methods in feature extraction step. The recognition digits steps are going to be described; discussed in details.

Various font types are available now days, which added difficulty to recognize digits typed, which make the feature extraction relatively hard where more font options are to be taken in consideration.

However, for examination the recognition capability in this paper, 7 types of fonts studied for all digits from 0 to 9, such as, each image from type .bmp, and contains 10 digits, each digit presents different font of the 7 fonts types.

Fig. 1: English digits

The proposed techniques is shown in the following stages of the paper, the following flow diagram shows all the basic stages in proposed techniques.

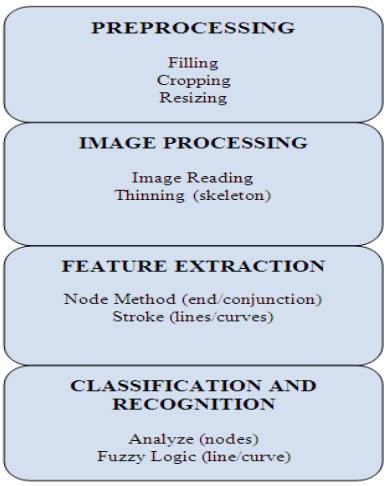

Fig. 2: Flow diagram for proposed techniques for recognizing English digits

III. PREPROCESSING AND IMAGE PROCESSING

Firstly in image preprocessing rotation, cropping, brightness, and clearance of the digit image will be done during the scanning of the image which will be done by the scanner program for all digits images. A set of operations on the image that will enhancement digits shape, a set of operations that will be applied on the input image to produce the probable images.

Then in image processing the image will be read as RGB (0-255,0-255,0-255) color, then it will be converted to gray scale (0-255) then it will be binarization (0,1). In binarization step some pixels will be removed and other pixels are filled with 1's.

Filling the pixels, using square structure element to perform the binary dilation on binary image, the edge is being traces for each object that are on the image. The preprocessing defines a compressed representation of the pattern, then obtaining the specified image according to a set of features. Such as image type "bmp" that contain set of digits as rows and columns [14]. The digits are in multi-fonts –7 fonts types – each digit in the image is a representation of a font type, this method is going to be trailed for numbers from 0 to 9.

Fig. 3: shows the colored, gray scale, and binary representation of digit 8

IV. FEATURE EXTRACTION

Feature extraction is the essential phase, thus distinguishing each digit from other digits. Therefore, it's very important to extract features so that the recognition of different digit becomes easier. Many methods considered in the feature extraction stage, a set of results is produced by each method. Then compound these two vectors of results to get the correct digit. There are duplicated states between, 6 and 9. The duplicated states will treat with other technique using the position of end point and conjunction point. The advantages of these methods stem from flexibility to recognize many fonts with well recognition rate, and no complex operation or calculations. Easy and fast way because no need to determine the entire shape; only the end and conjunction points with strokes types which represents the shape of the digit.

Nodes method, the number end points of each digit will be calculated in this method. Node is a pixel that has one or at most two neighbors concerning 8 pixels around it, that recognizing a digit is possible if we know the number of end/conjunction for that digit. The number of end/conjunction is used in this method to get out the inputted image. a thinning technique is going to be used giving the main skeleton, this method will be used in the pre-processing stage of pattern recognition system to compress data and to enhance feature extraction in the subsequent stage, and since it takes out a skeleton of the shape, all results going to be only 1 pixel thick, the thinning algorithm will be used in this study that can be found in MATLAB as the following with few enhancement on the results by removing the edge pixel for better skeleton [15]:

Thinned digit image =bwmorph (binary digit image,'thin',100)

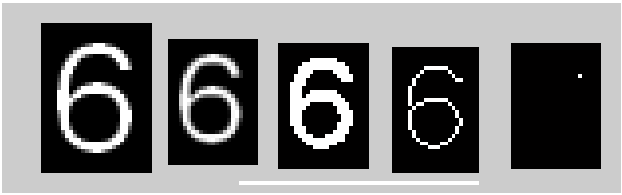

Fig. 4: Shows the thinning of digit 6

Divide the image into two distinct subfields in a checkerboard pattern. In the first sub iteration, delete pixel p from the first subfield, if and only if, the conditions G1, G2, and G3 are all satisfied. In the second sub iteration, delete pixel p from the second subfield, if and only if, the conditions G1, G2, and G3' are all satisfied [16].

Condition G1:

$$X_H(p) = 1$$

where

$$X_H(p) = \sum_{i=1}^{4} b_i$$

$$b_i = \begin{cases} 1, \text{ if } x_{2i-1} = 0 \text{ and } (x_{2i} = 1 \text{ or } x_{2i+1} = 1) \\ 0, \text{ otherwise} \end{cases}$$

$x1, x2, \ldots, x8$ are the values of the eight neighbors of p, starting with the east neighbor and numbered in counter-clockwise order.

Condition G2:

$$2 \le \min\{n_1(p), n_2(p)\} \le 3$$

where

$$n_1(p) = \sum_{k=1}^{4} x_{2k-1} \vee x_{2k}$$

$$n_2(p) = \sum_{k=1}^{4} x_{2k} \vee x_{2k+1}$$

Condition G3:

$$(x_2 \vee x_3 \vee \bar{x}_8) \wedge x_1 = 0$$

Condition G3':

$$(x_6 \vee x_7 \vee \bar{x}_4) \wedge x_5 = 0$$

The number of end/conjunction going to be calculated after the thinning step, And since each shape has a specific number of end/conjunction, and in some cases the number might get repeated with several shapes; as shown in Fig. 5, each shape generates different number of end/conjunction.

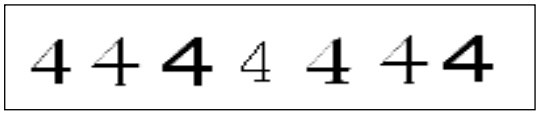

Fig. 5: Four rises in several shapes

To calculate the number of end/conjunction points Matlab function is used, for End points the following Matlab function used:

Newimage=bwmorph(skeleton_image,'endpoints',inf),

While for Conjunction points using the following function:

Newimage= conjunction_point(skeleton_image)

However there were so many un needed end points and conjunction point because of the skeleton itself, the skeleton (thinning) mat lab function created incorrect edge for all umbers because of the thickness of the font, therefore all edges must be removed before using the end, conjunction Matlab code.

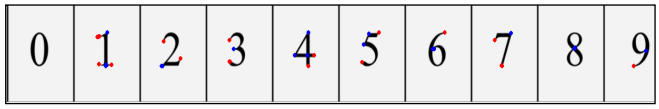

Fig. 6: Shows end in red/conjunction in blue for all digits

Count the number of all nodes in the shape such as endpoints, and conjunction points. Then compare the total number of nodes with the Table 1 for producing the output vector, consisting of expected digits. The Table 1 is prepared by experiential tests on dataset containing 7 font types.

Table 1: Results from node method for end node

Font type	*End points*									
	0	1	2	3	4	5	6	7	8	9
Amgdt	0	2	2	3	2	2	2	2	0	2
Arial	0	3	2	3	2	2	1	2	0	1
BELL MT	0	3	2	3	4	3	1	3	0	1
Courier New	0	3	2	3	3	2	1	2	0	1
Lucida Sans Typewriter	0	3	2	3	2	2	1	2	0	1
Sans Serif	0	2	2	3	2	2	1	2	0	1
Time New Roman	0	2	2	3	2	2	1	2	0	1
Average	**0**	**3**	**2**	**3**	**2**	**2**	**1**	**2**	**0**	**1**

Table 2: Results from node method for conjunction node of all fonts

Font type	*Conjunction points*									
	0	1	2	3	4	5	6	7	8	9
Amgdt	0	1	1	1	3	2	0	1	2	0
Arial	0	1	1	1	3	3	1	1	1	1
BELL MT	0	2	2	1	5	3	1	2	1	1
Courier New	0	2	3	1	5	2	1	3	2	1
Lucida Sans Typewriter	0	2	1	2	3	2	1	1	1	2
Sans Serif	0	2	2	1	3	2	1	2	1	1
Time New Roman	0	2	2	1	3	2	1	2	1	1
Average	**0**	**2**	**1**	**1**	**3**	**2**	**1**	**1**	**1**	**1**

Table 3: Results from node method for end and conjunction points numbers (x,y,p)

Number	*Elements of vector (x,y, p)*
0	(0, 0, 0)
1	(3, 2, 5)
2	(2, 1, 3)
3	(3, 1, 4)
4	(2, 3, 5)
5	(2, 2, 4)
6	(1, 1, 2)
7	(2, 1, 3)
8	(0, 1, 1)
9	(1, 1, 2)

By contemplation in the Table 1 to Table 6, we can observe 6 cases of nodes, which are concerned with five cases, and the last case (over 5 nodes) is excluded from this study, where those numbers which have over 5 nodes are irregular shapes, and will be needed for orientation with this study.

Table 4: Shows the distribution of digits according to number of nods

Number of nodes	*Elements*
0	0
1	8
2	6, 9
3	7, 2
4	3, 5
5	1, 4

Lines and curves detection: Each shape would represent a different digit that distinguishes it from others; an array will be created that contains the indexes for all the pixels in the given picture for each stroke of the shape.

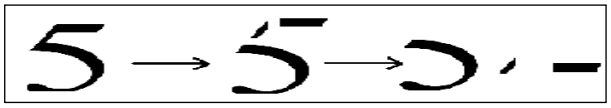

Fig. 7: Shows the Strokes of numbers 5

Strokes determine: strokes is created by a set of all possibilities. In a vector will be stored each stoke extracted out of the character. Scanning the stroke will be from the left upper corner to the right down corner assign end points and the conjunction points.

Lines determine*:* every stroke will be determined using tangent angle and mathematical operation and turn it into one kind of the lines (line, denoted as L).

$$MVL(x) = \begin{cases} 1 - |1/m|fe & \text{if } |m| > 1 \\ 0 & \text{if } |m| <= 1 \end{cases}$$

$$MhL(x) = \begin{cases} 1 - |m|fe & \text{if } |m| < 1 \\ 0 & \text{if } |m| >= 1 \end{cases} \quad (1)$$

$$Mob(x) = 1 - |(\theta - 45)/45|fe$$

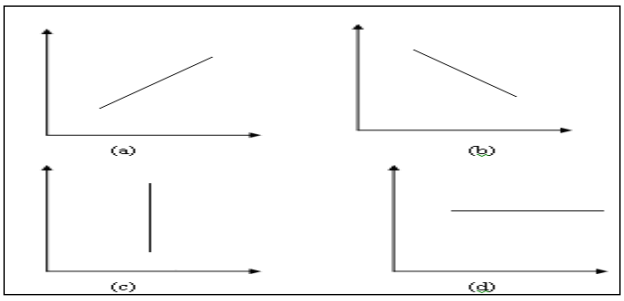

Fig. 8: a,b,c,d Shows Basic lines

Curves determine: The slop will be calculated for all carves in order to determine in which direction the curve will be and to classify any in-between curves to the nearest curve. Then the following formula is used to determine the lines between the end and the beginning of the curve: ***Middle Point of the curve Index order of element = (number of elements) /2 which denoted as (X3 ,Y3).*** After that if any element belongs to the line formula considering each curve, then it is a curve. The following formula is going to be used to determine the curves that represents the strokes and will be denoted as (C) [2]:

M(a) =Δ y /Δ x = y2-y1 / x2 –x1 , for any point on the line (x,y)
Let (x, y) belong A then
Y – y1 = m (x –x1) , Y – m x =y1 – mx1
Using (x3, y3) in the previous formula:
If m(a) = 0 and y3 < y1 then UC
If m(a) = 0 and y3 > y1 then DC
If x1 = x2 or {m(a) > 1 or M(A) < -1 } and X3 > x1 Then RC
If x1 = x2 or {m(a) > 1 or M(A) < -1 } and X3 < x1 then LC

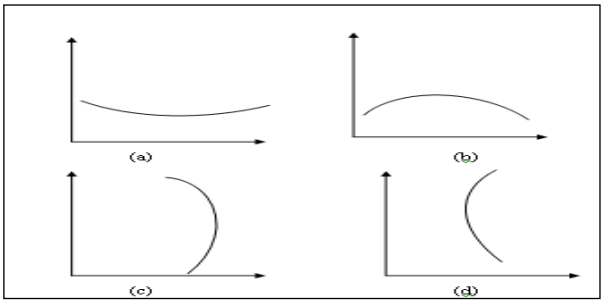

Fig. 9: a,b,c,d Shows Basic Curves

all digits will be presented by the following vector with five elements, first element is going to represent the number of end point in the digit, second element is for the number of conjunction point, third we are going to sum the total of end points and conjunction points of the digit, the fourth element is for where the end points to the conjunction is located (1-above, 2-below, 0-the same line), if more than one conjunction or no conjunction are found, the value will be N as null. And the fifth element will be for the stroke.

V1 = (number of endpoint, number of conjunction point, number of total points, number for position, stroke)

Table 5: Expected output vectors of shape

Characters of Shape	*Output Vectors*
1- L (line)	1, 4 ,7
2- C (curves)	3, 6, 9
3- LC (both)	2, 5
4- CR (circle)	0, 8

V. CLASSIFICATION AND RECOGNITION

In this stage all features extracted from the shape of input digit will be used to recognize and expect the correct number. The vector is the output of the pervious methods; each contains a set of elements. Thus, in the classification stage, those elements that we are going to match according to

Table 6 that shows all the possible result of all possible states, the dataset used in testing and experiment consists of 7 font type. Then to determine and obtain the wanted output:

Calculate the number of nodes for the shape in the given image that will describe the expected elements in vectors of the nodes method, then study the shape of the digit use characters to describe the shape of the digit in image, the characters may be {L,C,LC,CR } or in some cases "fails". Table 5 shows the elements of vector obtained from this method. Then compound and union the elements that we found, form a vector. From the previous steps, confusion states that we found needed to treated, that confusion occurred between the digits 6 and 9. For that an extra operation are taken to clarify the confusion cases, the following Table 6 present the adopted measures:

V2 = (number of endpoint (E), number of conjunction point (C), number of total points (T), number for position (P), stroke id number (S))

Table 6: The results for union the two vectors

Number	*E*	*C*	*T*	*P*	*S*
0	0	0	0	N	4
1	3	2	5	N	1
2	2	1	3	0	3
3	3	1	4	N	2
4	2	3	5	N	1
5	2	2	4	N	3
6	1	1	2	1	2
7	2	1	3	0	1
8	0	1	1	N	4
9	1	1	2	2	2

Studying the position of the end point to the intersection point in the digits will help defer between the digits that caused confusion, which are 6 and 9. if the end point above the intersection point then specific variable (4^{th} parameter of the vector) will have the value of 1, if the endpoints below the conjunction points the value will be 2, if one of end point at the same level of conjunction then set to 0, otherwise set to N null. In case (6, 9): if the return value is 1 then the number is 6, if the return value is 2 then the number is 9.

Fig. 10: Duplicated case (6,9)

Algorithm

Read image number
For each number find conjunction and end nods
Store end node number as E, and conjunction numbers as C and total as T
Extract the shape for each number
Determine two letters for each shape(C,L,CL,CR)
Let each group specify number and store in S
Classify each number to its group number S
Create P as integer equal to 1 if numbers has conjunction node under the end node,2 if above, 0 same level, and null for more than one conjunction
Store set of parameters for each number
Number< E, C, T, P, S>
Compare new number vector with stored vector (find similarity)

VI. EXPERIMENTAL RESULTS

In this section the results retrieved through the application of the proposed technique are going to be presented as well as the analysis of the results. The reason for low recognition rate or the fonts are not being recognized, the recognition rate depends on the size of the given image and the type of the fonts that are used, and the efficiency of the method in the feature extraction, that shows the flexibility of these methods and how it can recognize the fonts with different fonts.

Image Data: The used dataset to evaluate the performance of the proposed method is made up of set of images using same sizes and format. This study assumes that the given image does not have any noise from the scanning and to obtain the same results, we must observe the condition on the dataset.

Many experiments were needed to determine and describe the range of abilities for the proposed method in digit recognition. Some of the results came out well, other was failed. Table 7 shows the recognition level 70 digits.

Table 7: Results for recognize 70 digits from 7 fonts

digits	Recognition Rate
0	100%
1	57%
2	57%
3	85%
4	71%
5	71%
6	85%
7	57%
8	71%
9	85%
Average	74%

The observation from Table 7 is that the number of recognition states larger than the number of non-recognition states. The errors of recognition results are due to error either in shape or nodes method or both.

Due to thinning of the shape; node error occur, shape which increases number of nodes or changed in number of nodes due to irregular shapes. The deformed shapes originally are caused by error reading for skeleton of the shape. And this was fixed by deleting the edge pixels which enhanced the recognition rate.

In Table 7 the recognition rate are calculated; by dividing the number of recognition state by the number of all states (74%). Each digit (from 0 to 9) has an image that contains 7 fonts. The font size is considered 12 for all the fonts. From the given results that are shown in Table 7, we found out that some of the fonts have high recognition rate. The fonts recognized from 0 to 9; the recognition reach to 100%.

Some font recognition rate was low and failed for some fonts. Overall, all these results are expected, because the original work did not have perfect recognition result, plus the difference between the types of the fonts are massive. The change in size or style type for some fonts doesn't affect the recognition rate, because the shape and the taken skeleton of these fonts didn't change through editing the style of it. The rate for some reached to 100% or less than 50%. In some other font recognition rate; the result was low or failed, these fail and low cases, were either an error in studying the shape or in the number of nodes, are result of irregular shapes and disconnected edge of shapes that happen due to the difference of thinning results.

After studying the results it is been found that the results obtained from proposed technique on range of font types proved the flexibility for recognizing multi font. In principle for the taken method that have the ability to recognize a set of types, because we take the most repeated value for digits , also the end/conjunction method detecting number of endpoints and conjunction points will not be the same shape different if font type changed. The skeleton shape is almost steady for any size of font but slight difference for different types.

Few enhancement achieved by using shape category classification before end conjunction node classification step for recognizing the digit because the comparison will reduced to few element (at most 3 elements) rather than all digits, each category holds two to three elements such as category four (circular) which contain just 0 and 8 to recognize and by using the end conjunction node method it will be easier to recognize one of them.

VII. CONCLUSIONS

The proposed technique for the digit recognition appearances the capability to recognize the digits and multi font digits, the node method and feature extraction of the shape method were deployed on large number of fonts.

Through studying and analyzing the experiments; results gained, we concluded that the proposed technique is suitable, and able to give a good result, and its qualified to recognize 100 fonts in 90% rate of recognition. The shape study was developed through a set of simple operations; the classification stage relies on logical comparison operation.

The presented technique is flexible for change in size and type of fonts. The technique is simple and easy, therefore, can use it in PDA or in limited memory processor devices, also has the ability to recognize irregular shape for some fonts.

VIII. PROBLEMS

Some of the problems accord because of the skeleton algorithm from Matlab which gives incorrect shape for the digits specially at the end of the shape, which will give wrong conjunction and end points when using end/conjunction methods but we solved it by deleting the edge of the shape which gives excellent results. Another problem due the irregular shape of font type which gives almost different shapes from the rest of font types and this is the reason of the unrecognized rate.

IX. FUTURE WORKS

Work to improve the efficiency of this technique, and increased rate of recognition and identify a significant number of fonts and sizes. Apply method to recognize, handwritten, isolated, English and Arabic characters.

REFERENCES

[1]. Aj Palkovic, "Improving Optical Character Recognition", Villanove University, CSRS, Vol. 4, No. 2, pp. 109-114, December 2001.

[2]. Majdi Salameh " Arabic Digits Recognition Using Statistical Analysis for End/Conjunction Points and Fuzzy Logic for Pattern Recognition Techniques" World of Computer Science and Information Technology Journal (WCSIT),ISSN: 2221-0741,Vol. 4, No. 4, 50-56, 2014.

[3]. N.S.Arjun, G.Navaneetha, G.V.Preethi and T.K.Babu, "An approach to multi-font numeral recognition", TENCON 2007 - 2007 IEEE Region 10 Conference , Taipei, 2007.

[4]. Frank Y.Shih, "Image Processing and Pattern Recognition Fundamentals and techniques", John Whily and Sons, Directory of IEEE Book and Information services, pp.3-12, 2010.

[5]. Frank Y.Shih, "Image processing and Mathematical morphology fundamentals and techniques", Wiley-IEEE, CRC Press, United State of America, pp. 4-17, 2009.

[6]. L.Lam, Seong-Whan Lee, and Y. Suen Ching, "Thinning Methodologies-A Comprehensive Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 9, pp. 869-885, September 1992.

[7]. B.V.Dhandra, V.S.Malemath, H.Mallikarjun, and H. Mallikarjun, "Multi-font Numeral Recognition without Thinning based on Directional Density of Pixels", Gulbarga Univ., Digital Information Management, 1st IEEE International Conference, 2006.

[8]. Jun rkimoto , katashi nagao "the wold thrugh the computer : computer augmented interaction with real word environments", proceeding UIST 95 the 8th annual ACM symposium on user interface and software technology, p29-36.

[9]. Mou-yen chen, kundu.A ,jian zhou," off-line handwritten word recognition using a hidden markov model type stochastic network", pattern analysis and machine intelligenc, IEEE transaction , col:16 , issue: 5, page: 481-496, 1994.

[10]. N. Belkhamza, A. Akkouche, "Fuzzy Logic Character Recognition", Madinah College of Technology, Saudi Arabia, 2004.

[11]. Abhijit S. Pandya, Robert B. Macy, "Pattern recognition with neural networks in C++", CRC Press, SBN: 0849394627, Inc. Boca Raton, FL, USA ,1995.
[12]. Hany ferdinando," handwrittendigit recognition with fuzzy logic" jurnal teknik elektro, Vol. 3, No. 2, pp. 84-87, Sep 2003.
[13]. M. K. Al.bakkar, M. rababah ' A Hybrid Technique for Multifonts Numeral Recognition' al albyt university, jordan, Master Thesis, 2011.
[14]. W. Abu-Ulbeh, "New Algorithm to Noise Reduction using Morphological Operations", Msc. Thisis, Al Al-Bayt University, Mafraq, Jordan, 2009.
[15]. Oge marques " practical image and video processing using matlab ", publisher john wiley & sons, p. 356, 2011.
[16]. R. C .Gonzalez, and Woods, "Digital image processing", second edition, New Jersey: Prentice Hall, pp. 34-69, 2002.