Recognition of Printed English Characters Using Morphological Operations

Rana A. Mohammed
College of Computer Sciences and Mathematics
University of Mosul, Mosul, Iraq

Received on: 29 / 12 / 2010 **Accepted on: 16 / 3 / 2011**

ABSTRACT

A morphological printed English character recognition approaches for binary images is shown in this research. Three morphological operations are used for printed English character recognition: Dilation, Erosion and Hit-or-Miss.

Keywords: Morphological operations, English characters.

تمييز الأحرف الانكليزية المطبوعة باستخدام الطرائق المورفولوجية

رنا محمد
كلية علوم الحاسوب والرياضيات، جامعة الموصل

تاريخ استلام البحث: 2010/12/29 **تاريخ قبول البحث: 2011/03/16**

الملخص

في هذا البحث تم استخدام الطرائق المورفولوجية (علم التشكل) لغرض تمييز الأحرف الانكليزية المطبوعة في الصور الثنائية. هنالك ثلاث طرائق مورفولوجية تم استخدامهما لغرض تمييز الأحرف:- التحشية والتآكل والتواجد – أو – الفقدان Hit – or - Miss.

الكلمات المفتاحية: الطرائق المورفولوجية، الاحرف الإنكليزية.

1. Introduction:-

The word morphology commonly denotes a branch of biology that deals with the form and structure of animals and plants. The same word is used in the context of mathematical morphology as a powerful tool for extracting image features and components that are useful in the representation and description of region shape such as boundaries, skeletons, and the convex hull. Morphological techniques are often used to pre-process or post-process images to facilitate analysis, such as morphological filtering, thinning and pruning [8].

There are three primary morphological functions: *Erosion, dilation, and hit-or-miss*. The others are special cases of these primary operations or are cascaded applications of them [6].

In morphology, a small shape (structuring element) is translated across the image during the course of processing. Certain mathematical logic operations are performed on the image using the structuring element to generate the processed image [1].Therefore morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, we can construct a morphological operation that is sensitive to specific shapes in the input image [5].

Character recognition has become one of the most successful applications of technology in the field of pattern recognition and artificial intelligence. The area of printed English and Arabic characters recognition has been the subject of many research in the past years[2][3][7].

In section 2 of this paper, some basic concepts from set theory are presented. Structure element description is given in section 3. Section 4 contains the primary morphological operations. Some practical applications are illustrated in section 5, highlighting the printed English characters recognitions.

2. Some Basic Concepts From Set Theory:-

Let ***A*** be a set in $\boldsymbol{Z^2}$, the elements of which are pixel coordinates ***(x,y)***. If ***w=(x,y)*** is an element of ***A***, then

$$w \in A \tag{1}$$

Similarly, if ***w*** is not an element of ***A***, then

$$w \notin A \tag{2}$$

a set B of pixel coordinates that satisfy a particular condition is written as

$$B = \{\, w \mid condition \,\} \tag{3}$$

For example, the set of all pixel coordinates that do not belong to set ***A*** denoted $\boldsymbol{A^c}$ is given by

$$A^c = \{w \mid w \notin A\} \tag{4}$$

This is called the complement of ***A***.

The union of two sets denoted by $\boldsymbol{A \cup B}$, is the set of all elements that belong to either ***A*** or ***B*** or both. Similarly, the intersection of two sets ***A*** and ***B*** is the set of all elements that belong to both sets, denoted by $\boldsymbol{A \cap B}$**.**

The difference of sets ***A*** and ***B*** denoted by ***A-B*** , is the set of all elements that belong to ***A*** but not to ***B***

$$A - B = \{w \mid w \in A, w \notin B\} \tag{5}$$

Figure (1) illustrates the basic set operations.

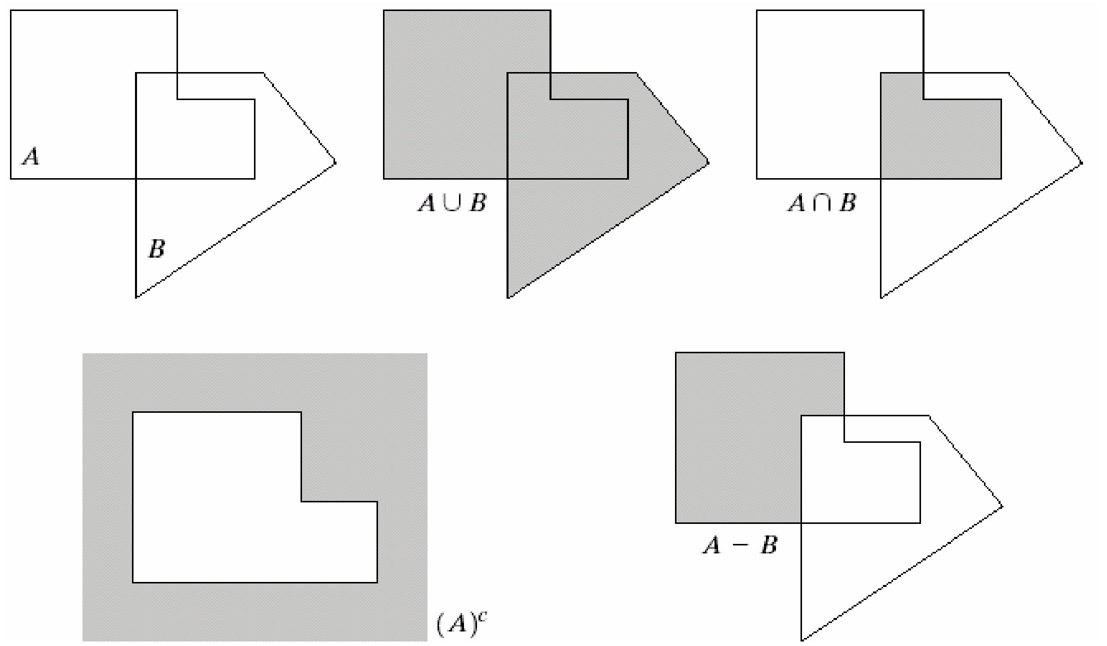


Figure (1). Basic set operations

The logical expression are used to perform the set operation on binary images, as shown in Table (1) [8].

Table(1). The logical expressions for set operations

Set operation	Logical expression	Name
$A \cap B$	A & B	AND
$A \cup B$	$A \mid B$	OR
A^c	~A	NOT
$A - B$	A & ~B	DIFFERENCE

3. Structuring Element

The basic idea in binary morphology is to probe an image with a simple pre-defined shape, drawing conclusions on how this shape fits or misses the shapes in the image. This simple "probe" is called structuring element and is itself a binary image [8].

A structure element is a matrix consisting of only 0's and 1's that can have arbitrary shape and size. The pixel with values of 1 define the neighborhood [5].Two dimensional structuring elements are typically much smaller than the image being processed. The center pixel of the structuring element, called the origin, identifies the pixel of interest (the pixel being processed)[5].

4. Primary Morphological Operations

The operations of *dilation* ,*erosion* and hit-or-miss are fundamentals to morphological image processing [8]. The rules for Dilation and Erosion is shown in the following Table(2) [5].

Table (2). Rules for Dilation and Erosion

Operation	Rule
Dilation	The value of the output pixel is the *maximum* value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to the value 1, the output pixel is set to1.
Erosion	The value of the output pixel is the *minimum* value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to 0, output pixel is set to 0.

4.1 Dilation

Dilation is an operation that "grows" or "thickens" objects in binary image. The specific manner and extent of this thickening is controlled by a shape referred to as a *structuring element* [8].

How it works

The dilation operation takes two pieces of data as input:-

1) The image which is to be dilated (almost binary).
2) Structuring element.

The dilation process is performed by laying the structuring element on the image and sliding it across the image in a manner similar to convolution. It is best described in a sequence of steps:

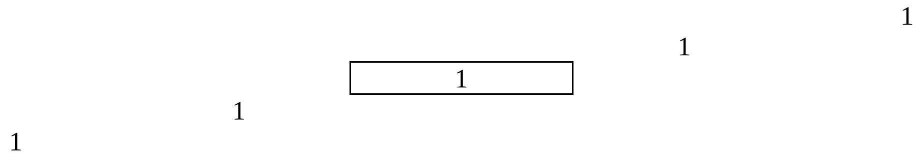
1) If the origin of the structuring element coincides with a '0' in the image, there is no change; move to the next pixel.
2) If the origin of the structuring element coincides with a '1' in the image, perform the OR logic operation on all pixels within the structuring element[9].

Figure (2) illustrates how dilation works. Figure (2.a) shows a simple binary image containing a rectangular object. Figure (2.b) is a structuring element, a five-pixel-long diagonal line in this case. Computationally, structuring elements typically are represented by matrix of 0s and 1s; sometimes it is convenient to show only the 1s,as illustrated in this figure. In addition, the origin of the structuring element must be clearly identified. Figure(2.b) shows the origin of the structuring element using a black outline. Figure(2.c) graphically depicts dilation as a process that translates the origin of the structuring element throughout the domain of the image and checks to see where it overlaps with 1-valued pixels. The output image in figure(2.d) is 1 at each location of the origin such that the structuring element overlaps at least one 1-valued pixel in the input image. Dilation of *A* by *B* denoted by: $\boldsymbol{A \oplus B}$

In words, the dilation of *A* by *B* is the set consisting of all structuring element origin location where the reflected and translated *B* overlaps at least some portion of *A* [8].

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure (2.a). Original image with rectangular object



Figure(2.b). Structuring element with five pixels arranged in a diagonal line. The origin is shown with a dark border

																▨
															▨	
														▨		
						▨	#	#	#	#	#	#	▨ #			
					▨	#	#	#	#	#	#	▨ #	#			
				▨	1	1	1	1	1	1	1	#	#			
			▨	#	1	1	1	1	1	1	1	#				
		▨	#	#	1	1	1	1	1	1	1					
			#	#	#	#	#	#	#	#						
			#	#	#	#	#	#	#							

# When the origin is translated to the # locations, the structuring element overlaps 1-valued pixels in the original image.

▨ The structuring element translated to these locations does not overlap any 1-valued pixels in the original image.

Figure(2.c). Structuring element translated to several locations on the image

				1	1	1	1	1	1	1
			1	1	1	1	1	1	1	1
		1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1		
1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1				

Figure(2.d). Output image

Figure(2). illustration of dilation

4.2 Erosion

Erosion "shrinks" or "thins" objects in a binary image. As in dilation, the manner and extent of shrinking is controlled by a structuring element [8].

How it works

The erosion operation takes two pieces as input:

1) The image which is to be eroded (almost binary).
2) Structuring element.

The erosion process is similar to dilation, but it turn pixel to '0', not '1'. As before slide the structuring element across the image and follow these steps:-

1) If the origin of the structuring element coincides with a '0' in the image, there is no change; move to the next pixel.
2) If the origin of the structuring elements coincides with a '1' in the image and any of the '1' pixels in the structuring elements extend beyond the object('1' pixels) in the image, then change the '1' pixel in the image to a '0' [9].

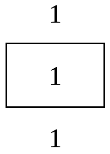
Figure(3) illustrates the erosion process. Figure(3.a) is the same as figure(2.a). Figure(2.b) is the structuring element, a short vertical line. Figure(3.c) graphically depicts erosion as a process of translating the structuring element throughout the domain of the image and checking to see where it fits entirely within the foreground of the image. The output image in figure(3.d) has a value of 1 at each location of the origin of the structuring element, such that the element overlaps only 1-valued pixels of the input image(i.e. it does not overlap any of the image background).

Erosion of *A* by *B* denoted by: $\boldsymbol{A \Theta B}$

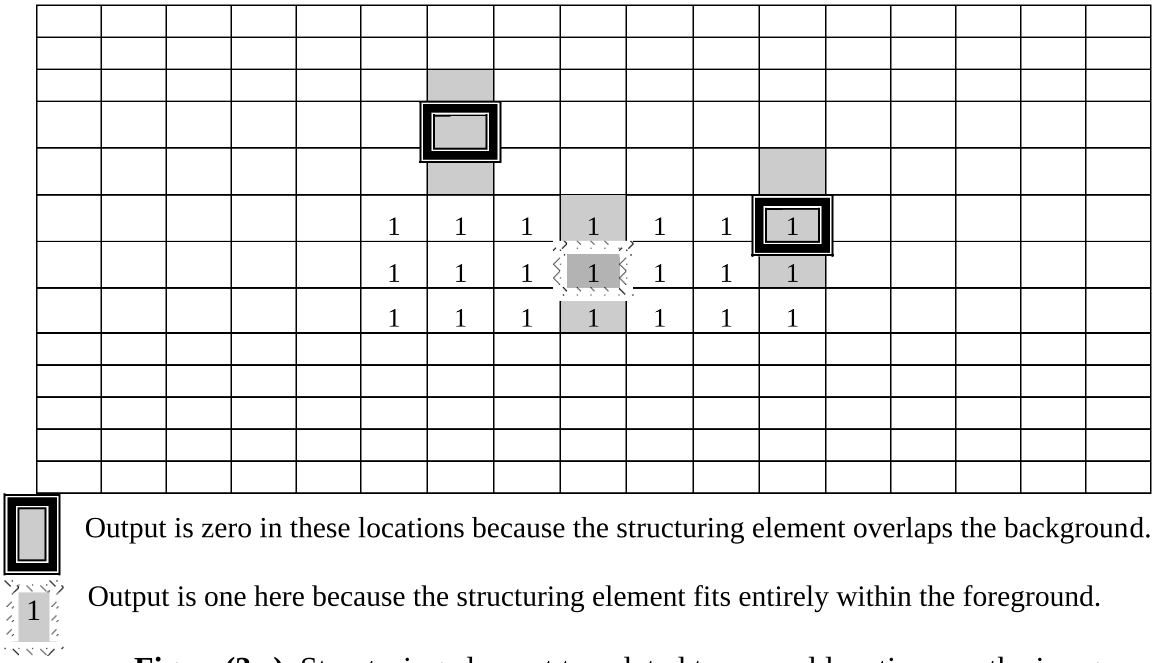
In words, erosion of *A* by *B* is the set of all structuring element origin locations where the translated *B* has no overlap with the background of *A* [8].

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure(3.a). Original image with rectangular object



Figure(3.b). Structuring element with three pixels arranged in a vertical line. The origin is shown with a dark border



Output is zero in these locations because the structuring element overlaps the background.

Output is one here because the structuring element fits entirely within the foreground.

Figure(3.c). Structuring element translated to several locations on the image

1 1 1 1 1 1 1

Figure(3.d). Output image

Figure(3). Illustration of erosion

4.3 Hit-or-Miss

It is useful to be able to identify specified configuration of pixels, such as isolated foreground pixels, or pixels that are end points of line segments The Hit-or-Miss Transformation is useful for applications such as these [8].

How it works

The hit-or-miss takes two pieces of data as input:
1) The image which is to be processed by hit-or-miss (almost Binary).
2) Two Structuring Element.

The Structuring element used in the Hit-or-Miss is a slight extension to the type that has been introduced for erosion and dilation, in that it can contain both foreground and background pixels, rather than just foreground pixels, both ones and zeros.

Note that the simple type of Structuring element used with erosion and dilation is often depicted containing both ones and zeros as well, but in that case the zeros really stand for "don't care's", and are just used to fill out the structuring element to a convenient shaped kernel.[8]

The Hit-or-miss transform of A by B is denoted by A⊗B, here, B is a structuring element pair, B= (B1, B2), rather than a single element, as before.
The Hit-or-miss transform is defined in terms of these two structuring element as:

$$\mathbf{A \otimes B = (A\ \theta\ B1) \cap (A^c\ \theta\ B2)} \tag{6}$$

Figure (4) shows how the Hit-or-miss transform can be used to identify the locations of the following cross-shaped pixel configuration. Figure (4.a) contains this configuration of pixels in two different locations. Figure (4.c) is erosion with structuring element B1 determines the location of foreground pixels that have north, east, south, and west foreground neighbors. Figure (4.d) is erosion of the complement with structuring element B2 determines the location of pixels whose northeast, southeast, southwest, and northwest neighbors belong to the background. Figure (4.g) is show the intersection (logical AND) of these two operations. Each foreground pixel of figure (4.g) is the location of a set of pixel having the desired configuration [8].

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure (4.a). Original Image A.

1

1 1 $\boxed{1}$

1

Figure (4.b). Structuring Element B1.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure (4.c). Erosion of A by B1.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	0	0	0	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	0	1	1
1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	1
1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1
1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure (4.d). Complement Of Original Image A^c.

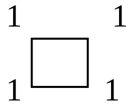


Figure (4.e). Structuring Element B2.

1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	0	1	0	0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1
1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	1
1	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0
1	1	1	1	0	1	0	1	1	1	1	1	0	1	0	1
1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1

Figure (4.f). Erosion of A^c by B2.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure (4.g). Output Image.

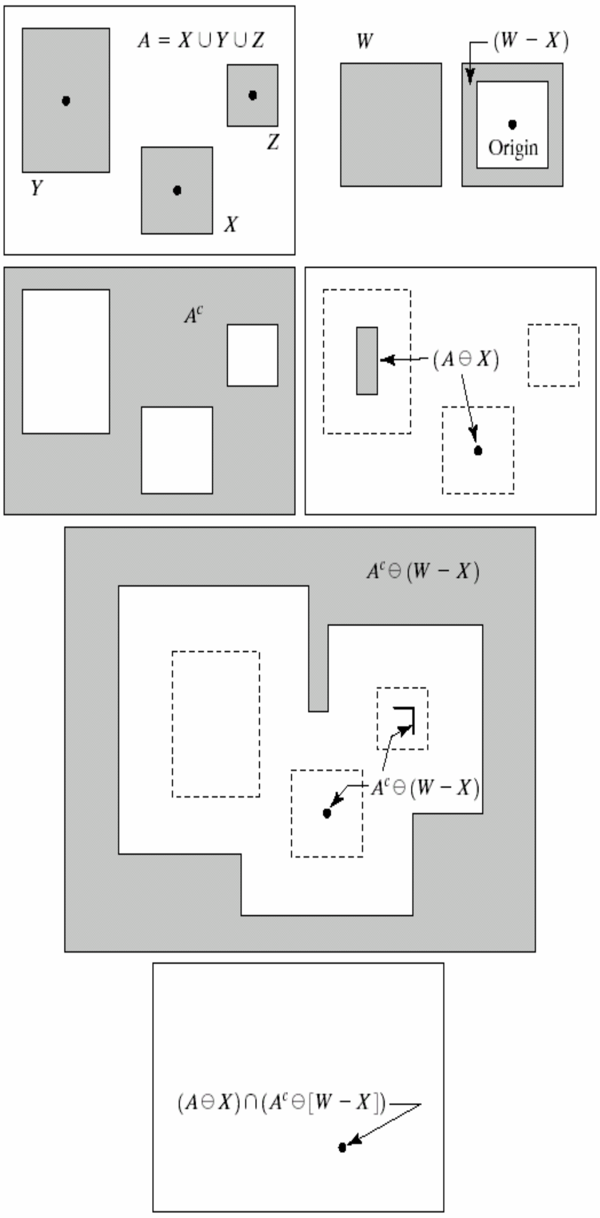
Figure (4). Illustration of Hit-or-Miss operation.

5. Practical Applications

The morphological *hit-or-miss* transform is a basic tool for shape detection, If B denoted the set composed of X and its background, then the match of B in A is

$$\mathbf{A\otimes B= (A\ \theta\ X) \cap [\ A^c\ \theta\ (W - X)]} \qquad (7)$$

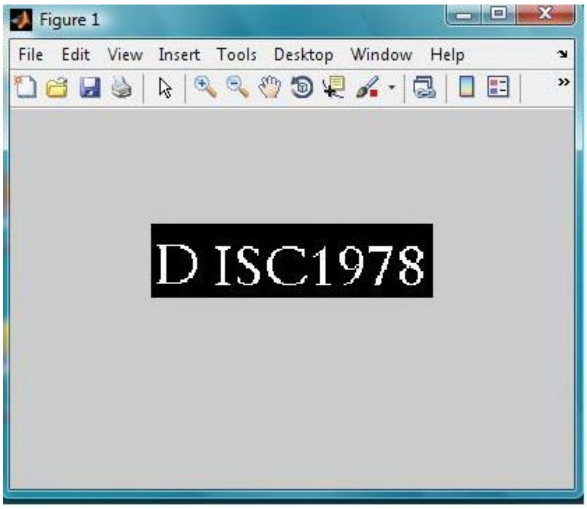
As shown in figure (5).



a b
c d
e
f

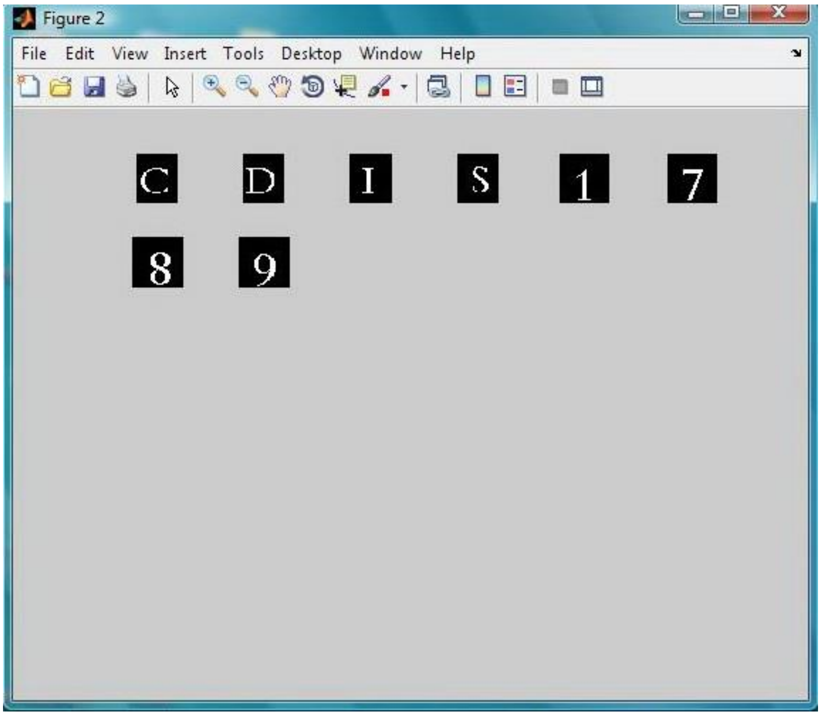
(a) Set **A.** (b) **A** window, **W**, and the local background of **X** with respect to **W**,**(W-X)**.
(c) Complement of **A**. (d) Erosion of **A** by **X**. (e) Erosion of $\mathbf{A^c}$ by **(W-X)**.
(f) Intersection of (d) and (e), showing location of the origin of **X**, as desired.

Figure (5).



Window (1) Test image

So we use the equation(7) in this paper in order to recognize printed English characters in binary images by comparing the test image (for example test image shown in window(1))with all printed English characters and numbers saved images, then the result of recognized printed English characters and numbers is given as shown in window(2) .



Window (2) Result of recognized numbers and characters

6. Conclusion

This paper shows that morphological approaches is good for recognition of printed English characters in binary images, because it takes little time in execution.

REFERENCES

[1] A. John Wiley and Sons Inc., 2007: " Digital Image Processing", Publication.

[2] Amin A., 2000: "Recognition of printed Arabic text based on global features and decision tree learning techniques", pattern recognition, Vol.33, pp.1309-1323.

[3] Bazzi l, Schwartz, R., Makhoul, J., 1999: "An Omnifont Open-Vocabulary OCR System for English and Arabic", IEEE, transaction on pattern analysis and machine intelligence, Vol. 21, No. 6, pp. 495-504.

[4] C. Pu and F. Shih, 1995: "Threshold Decomposition of Grey-Scale Soft Morphology into Binary Soft Morphology", CVGIP-Graphical Models and Image Processing, Vol.57, No. (6), pp. 522-526.

[5] Help of MATLAB.

[6] H. Heijmans, 1994: "Morphological Image Operators", Academic Press, New York.

[7] Mantas j., 1986: “An Overview of Character Recognition Methodologies”, Pattern Recognition, Vol.19, No.6, pp. 425-430.

[8] R. Gonzalez and R. Woods, 2004: "Digital Image Processing Using MATLAB", Addison-Wesley", New York.

[9] Scott E. Umbaugh, 1998: "Computer Vision and Image Processing", Prentice Hall PTR Upper Saddle River.