ISSN: 0067-2904
GIF: 0.851

Utilization of Edge Information in Handwritten Numerals Recognition

Nada Najeel Kamal*
Department of Computer Science, University of Technology, Baghdad, Iraq

Abstract

The recognition of handwritten numerals has many applications in automatic identification and cognition. This research contains three experimented scenarios to recognize the handwritten English (i.e. Arabic) numerals. In the first scenario the bilinear interpolation of the image is used, while in the second scenario and after the bilinear interpolation is being applied, the Sobel operators are applied on the resulted interpolated image. In the third scenario which represents the last one, the effect of normalization of image dimensions is tested. 550 images of handwritten numerals were tested. Three types of tests were conducted for each scenario namely: trained-set test, not-trained-set test and comprehensive-set test. Depending on the results obtained from the comprehensive-set test, the best scenario is the second scenario of bilinear interpolation followed by Sobel operators which leads to excellent success rate reaches to 97.63%.

Keywords: Handwritten, Recognition, Numerals, Sobel, Pattern Recognition.

أستخدام معلومات الحافة في تمييز الأرقام المكتوبة بخط اليد

ندى نجيل كمال*
قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق

الخلاصة

تَمييزُ أشكال الأرقام المكتوبة بخط اليد له العَديد من التَطبيقات في التعرّف والأدراك المُمَكّن. يَتَضّمن هذا البحث تجارب لثلاث سيناريوهات لتمييز الأرقام الأنكليزية (بمعنى:العربية) المكتوبة بخط اليد. في السيناريو الأول استُخدمَ الأستكمال ثنائي الخطية، بينما في السيناريو الثاني وبعد الأستكمال ثنائي الخطية طبّق عاملي صوبل المؤثرين على الصورة المستكملة الناتجة. في السيناريو الثالث، والذي يمثّل السيناريو الأخير، تم أختبار تأثير مساواة أبعاد الصورة. أختبرت ٥٥٠ صورة للأرقام المكتوبة بخط اليد. ثلاث أنواع من لأختبارات أجريت لكل سيناريو: أختبار المجموعة المدرّبة وأختبار المجموعة غير المدّربة والأختبار الشامل. بناءً على النتائج المستحصلة من الأختبار الشامل، أفضل سيناريو هو السيناريو الثاني بتطبيق الأستكمال ثنائي الخطية المتبوع بعاملي صوبل المؤثرين والذي أعطى نسب نجاح ممتازة وصلت إلى ٩٧،٦٣%.

Introduction

The recognition of handwritten numerals can be used in many applications like, security applications, electronic government and automatic postal communications.

This researches tackles the problem of auto-recognizing of handwritten English numerals by experimenting three different ways to find the one which gives the most accurate results with less time. Many literatures explored this problem using different scientific methods that achieved different results, for example: support vector machine (SVM) classification and nearest neighbor where used in [1] to identify the English and Kannada characters, including numerals, in natural scenes. Six different local features are extracted from each single character and fed to the classifiers.

*Email: nadajeelkaml@yahoo.com

This paper claimed the using of 15 training images, and the ability of enhancing the accuracy of recognition by 25% over OCR (optical character recognition) systems. While in [2], the neural network is the used methodology to recognize of handwritten numerals from 0 to 10. The scanned set of numerals' sheets are sliced to get isolated 16x16 pixels of numeral image. A horizontal, vertical and diagonal histograms are stored in a database and the fed to a neural network for discrimination. Recognition rate is not documented.

Fourier descriptor and normalized chain code are used in [3] to recognize Marathi handwritten numerals. The included steps before the methods of Fourier and chain code were applied are: noise removing, image binarizing, erosion and dilation, cropping to size of 40x40 pixels, boundary extracting, and representing boundary in a complex plane. 12690 tested samples where used with recognition rate of 98.15%.

A statistical classifier and 17 geometric features were used for the recognition of Devnagari handwritten numerals in [4]. These features like: the number of vertical, horizontal, right and left diagonal lines, filled area, and perimeter. This research used 1500 samples for training and,1500 samples for testing. The highest recognition rate reaches to 95.33% for the numeral "8".

The particle swarm optimization algorithm (PSOA) was used in [5] to recognize the isolated Arabic characters. After the characters being normalized to 28x28 pixels, local, global and diagonal features are extracted from character images. 168 samples used to training. The recognition rate reaches up to 87.856%.

The Used Database

The used database (DB) is available at http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/. This DB consists of 550 image comes in Portable Network Graphics (PNG) format. These 550 images fall into 10 classes for handwritten English (Arabic) numerals starting from class 0 to class 9. Each class contains 55 images. The width and the height of the class' images are 1200x900 pixels. The first 10 samples of each class of the DB are shown in Figure-1.

Preparing the images

Since the images of the DB used for the experiments come in PNG format, and because of the well-known properties of the Bitmap (BMP) format, such as it is non compressed format and it is a Microsoft easy format that is compatible with Windows operating system environment which represents the format of our experiments, then, the images of the DB are converted to Bitmap (BMP) format. Also, since the images of the DB comes in a large size of 1200x900 pixels, which means a high cost of time required for the processing. These images are converted into a size of 400x300 pixels for every image's width and height respectively, with the depth of 24 bit per pixel of the image.

The up mentioned conversions are not part of the tested recognition system, and that’s why an application program is used to perform them. The application program that is used to convert the image form PNG format to BMP format and to the size of 400x300 pixels is "Ashampoo Photo Commander 11.0.3". This application program provides a facility of converting a batch of images from one format to another along with ability of resizing these images. Since the images comes in a high resolution, and the converted to 400x300, so no noise removing algorithms are used here.

Figure 1- Samples of DB

Experiments Layout

The layout of the three tested experiments is shown in Figure-2.

Flood Fill Algorithm

To eliminate the space surroundings the region of interest of the numeral as much as possible, The BMP image of each numeral in the tested DB is segmented using flood fill algorithm. To perform the image segmentation, first, the bands of the input BMP image are extracted. Then, a gray image is extracted through the sum of the three bands and divide them by their number, which is three, as shown in equation (1).

$$\boldsymbol{Gray(x,y)\ \ image = Red(x,y)+ Green(x,y)+ Blue(x,y)/3} \qquad \textbf{(1)}$$

Where *Gray(x,y)* is the obtained gray image from the conversion of image pixels at location (x,y). *Red(x,y)*, *Green(x,y)*, and *Blue(x,y)* are the three image in that specific pixel location.

Then a binary image is obtained from the Gray image, using a threshold of 128. Where, every pixel in the gray image is scanned, if it is smaller than the determined threshold, then it is converted to black, while if the value of that pixel of the gray image is greater than the threshold, it will be converted to white.

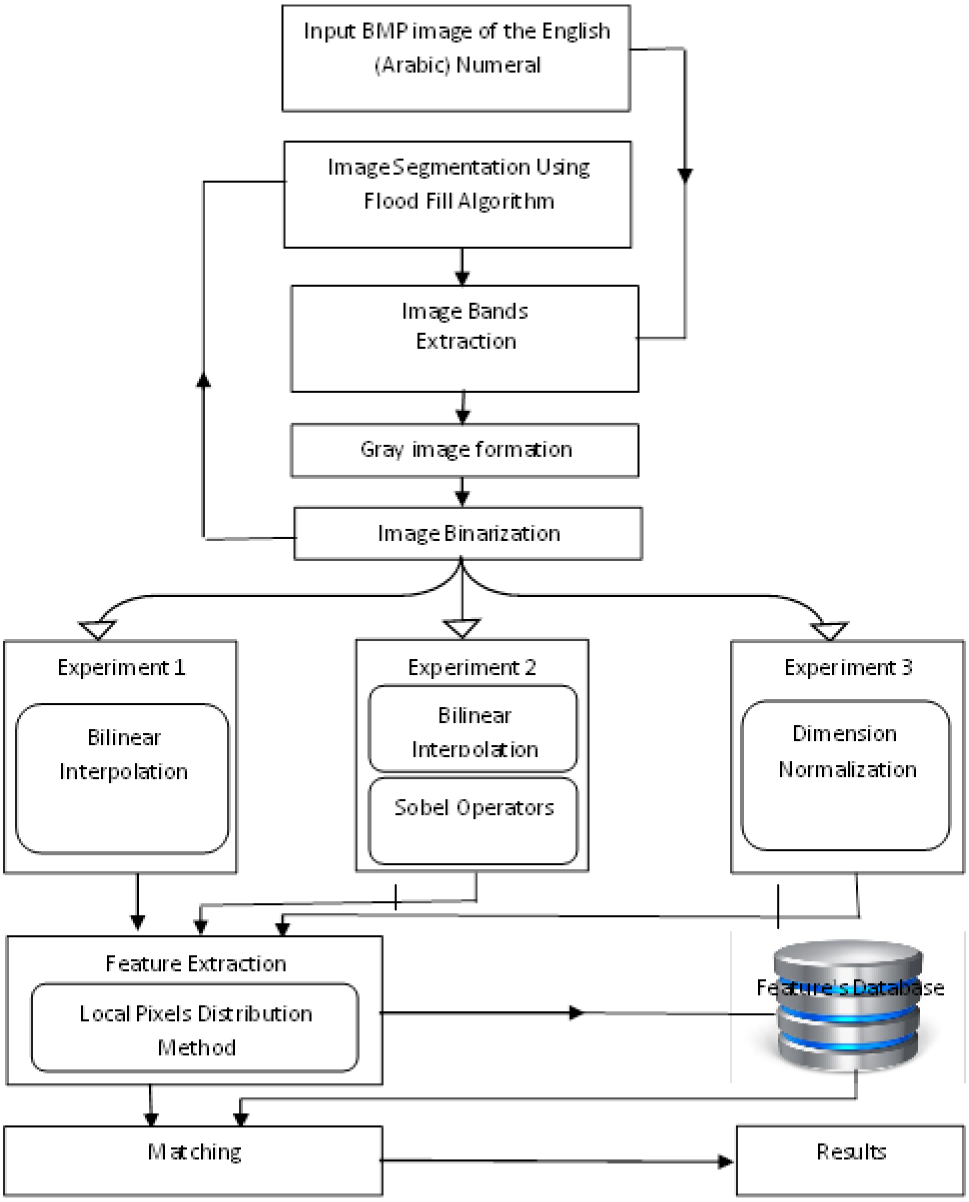
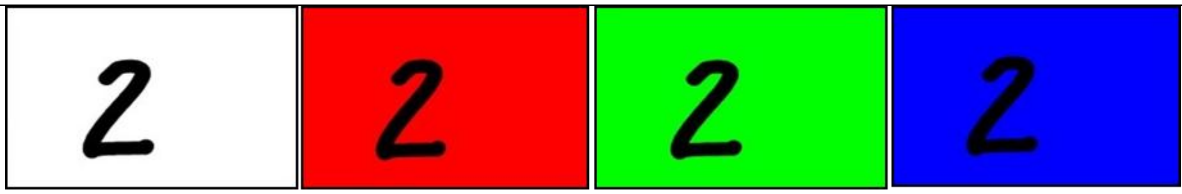


Figure 2- The experiments stages layout

Now, the binary image pixels are scanned to check if one satisfies the condition of being a numeral's pixel. So, if that condition is met then its location in the image will be recorded to a certain buffer. Then, the four neighbors of that pixel, the upper, the lower, the left, and the right one are tested too, if they met the condition of being part of numeral pixel, their location in the image will be recorded to that buffer too. This process is continued until all the locations of the numeral's pixels are recorded.

The locations that are stored in the buffer are used to create a numeral image stored in an array. That array represent a solid segment of the region of interest of the handwritten numeral, with less space surrounding the numeral. That array of segment is fed to procedure where it will be converted into a separate BMP image with related header and will be used in the next phases of the tested recognition.



(a) input image **(b) red band** **(c) green band** **(d) blue band**

(e) Resulted image from flood fill algorithm with less space surrounding the resulted image. This is the 47^{th} sample of the third class.

Figure 3- Input image and resulted segmented image.

Image Binarization

The BMP image that is resulted from the flood fill algorithm, as the one shown in Figure-3 (e) must be binarized. As a result of binarizaton the region of interest will be set to white and background will be set to black. So, to get the binary image the, three bands are extracted from the segmented image of the numeral. Then these bands will be used to get the gray image as in equation (1). The binary image will be obtained using a threshold value of 128, where every pixel value below that threshold will be converted to white, while the pixels with values below it will be turned into black. This is shown in Figure-4. Even all the tested numerals comes with dark color over a white background, which will results a gray image similar to the colored one, the conversion to gray scale image is necessary especially when different colors of pens used to write a numeral over a white background.

(a) Red band of the segmented image	(b) Green band of the segmented image	(c) Blue band of the segmented image	(d) Gray image	(e) Resulted binary Image

Figure 4- Successive converting to get the binary image

Preprocessing Scenarios

Now the resulted binary image, which is shown in Figure-4 image (e), is needed to be processed before it being passed to feature extraction phase to get a higher rate of successful recognition. This binary image will be fed to three different scenarios of preprocessing to see which one of these scenarios is the most suitable for a recognition system, by submitting these scenarios to the test. These three preprocessing scenarios are: Bilinear interpolation, Bilinear interpolation followed by Sobel operators, and dimension equalization.

First Scenario: Bilinear Interpolation

After the phase of image segmentation, the resulted images will be in different dimensions because it will depend on the area in pixels that the numeral takes in that image. To get a better performance of recognition, in this first scenario, all the images dimensions will be unified to the size of 130x130 pixel using bilinear interpolation. The used bilinear interpolation algorithm consists of the following steps:

- The determination of the new image dimension, which is 130x130 pixel.
- The determination of the longest dimension in the numeral image, Img(x,y).
- Set the longest dimension to a new variable.
- Finding the conversion coefficient (cc) which represents the difference between the original numeral dimension and new length to which will the image be converted to.

- Allocate a new array that will have the dimensions of the original dimension times the conversion coefficient, newImage (Lx, Ly).
- For every y from 0 to Ly Loop
    - Set yy ← The ratio between y and cc
    - Set Y1← integer value of yy
    - Set Y2← increment of Y1 by 1
    - Set Yf← the difference between yy and Y1
    - For every x from 0 to Lx loop
        - Perform the same four upper steps for x to get: xx, X1, X2 and Xf
        - Set P1 ← Img(X1, Y1) · (1 - Xf) + Img(X2, Y1) · Xf
        - Set P2 ← Img(X1, Y2)·(1 - Xf) + Img(X2, Y2) ·Xf
        - Set newImage (X, Y) ← P1 · (1 - Yf) + P2 ·Yf
    - End X loop
- End Y loop
- Allocate an image biImag with size of (130x130)
- Padd the image newImage with zeros.

The bilinear interpolation and padding is shown in Figure-5.

Figure 5- Bilinear interpolation with padding with size of 130x130.

Second Scenario: Bilinear Interpolation Followed by Sobel Operators

In this second scenario and after the image segmentation, the image is converted to fixed size of dimensions of 130x130 using the previous explained algorithm of bilinear interpolation. What followed the bilinear interpolation is the extracting of image edges using Sobel operators.

The Sobel operator involves the applying of its tow masks, the horizontal and the vertical, which are shown in Figure-6. The convolution of the tow masks with image that is resulted from bilinear will produce an image with edges as the one shown in Figure-7.

1	2	1
0	0	0
−1	−2	−1

(a)

1	0	−1
2	0	−2
1	0	−1

(b)

Figure 6- Sobel Masks [6]

Figure 7- Result of applying Sobel operators on interpolated image of the numeral.

Third Scenario: Dimension Normalization

Since the hand written numeral comes in different size, so the resulted image from the segmentation phase will come into different sizes. The sizes of these image, depending on the numeral area, ranges from 12x45 to 94x117.

The recognition algorithm that is used called local pixels distribution. This algorithm rely on equal areas in which the image is divided into. That’s why the image dimensions are set to be equal. For example the image with dimensions of 12x45 will be turned into an image with dimensions of 45x45 pixels. The following algorithm shows how width (W) and height (H) are equalized.

If W > H Then
  Set newD ←W
Else
  Set newD ←H
End If
Allocate EqImg(newD, newD)
Set Xs ← The difference between newD and W divided by 2
Set Ys← The difference between newD and H divided by 2
For every pixel in the numeral image Loop
  Set EqImg (add X's pixel to Xs, add Y's pixel to Yy) ← BinImg(X, Y)
End Loop

Numeral's Recognition with Local Pixels Distribution

Weather the image is obtained from bilinear interpolation, or bilinear interpolation followed by Sobel edge detector, or image dimension equalization; it will be fed to this recognition algorithm of local pixels distribution to identify and recognize the handwritten numeral.

This algorithm includes slicing the image into an equal sized areas in a square form. These areas are interlaced to certain proportions. Each area defines a feature value. This value is gained through the ratio of dividing the summation of pixels in each squared area by the total number of pixels in the entire numeral image. Figure-8 shows the numeral image that is resulted from bilinear interpolation followed by Sobel operators detector is sliced into 3x3 s interlaced squared areas.

Figure 8- Interlaced 3x3 squares

The template of the mean and the standard deviation for the features extracted from the numeral image is stored a database. The number of values in each template of a class is equal to the number of the squared areas where the image was divided into. For example if the images of certain class, say class "8", are divided into 5x5 interlaced squares areas, then number of features extracted from each single image will be equal to 25 features. Also, that means the number of values in the templates of the mean and the standard deviation will be equal to 25 value for each.

The used distance measure for comparing extracted features from a numeral image with the templates' values stored in the database is shown in equation (2).

$$\textit{Numeral Class No.} = \sum (feature_n - mean_n)^2 / \textit{standard deviation}_n \qquad \textbf{(2)}$$

Where *Numeral Class No.* represents the number that the numeral's image represents. *feature* refers to every feature extracted from the image. $_n$ represents the number of feature from 1 to n which equals to the number of the resulted squares from interlacing algorithm. $mean_n$ and *standard deviation*$_n$ are the mean and the standard deviation for that nth feature.

Results of the Applied Scenarios with the Recognition Algorithm

Since the 10 classes of the numerals consists of 55 images, each class is classified into two sets. The first set which used for training consists of 50 images in each class called the "Trained-Set", while The remaining 5 images in the class represents the images that are not involved in the training process called "Not-Trained-Set".

A *"Trained-Set" test* refers to testing the recognition algorithm using the trained samples. While the *"Not-Trained-Set" test* refers to testing the recognition algorithm using the "Not-Trained-Set" image samples. A *"Comprehensive-Set" test* involve testing the *all* of the class samples weather they were trained or not trained samples.

The three scenarios explained in this paper where tested using the three types of tests mentioned above. In each test the number of the true positive samples is calculated. True positive refers to the number of the samples which passed the local pixels distribution algorithm correctly. The percentage is calculated using equation (3). The time elapsed in the tests refers to the average of time required in seconds to recognize each single numeral image. To find the most suitable number of features to be extracted from the image that give the highest recognition rate, each image is divided from 2x2 to

100x100 squared areas. The proportion of interlacing between these areas ranges from 0 to 1, i.e.: 0,0.1, 0.2,0.3, … 1.

Percentage = No. of true positive samples/ Total Number of samples test **(3)**

The results of the "Trained-Set", "Not-Trained-Set" and the "Comprehensive-Set" tests for the explored three scenarios with local pixels distribution method are shown in Tables-1, -2 and 3 respectively.

Table 1- Tests results for ***bilinear interpolation*** scenario

Type of Test	Total samples	True Positive	Percentage	Time elapsed in seconds	No. of squared areas	Rate of interlacing
"Trained-Set" Test	500	467	93.4	0.045	49x49	0.3
"Not-Trained-Set" Test	50	46	92	0.021	11x11	1
"Comprehensive -Set" Test	550	511	92.90	0.077	64x64	0.4

Table 2- Tests results for ***bilinear interpolation followed by Sobel operators*** scenario.

Type of Test	Total samples	True Positive	Percentage	Time elapsed in seconds	No. of squared areas	Rate of interlacing
"Trained-Set" Test	500	499	99.8	0.158	92x92	0.2
"Not-Trained-Set" Test	50	43	86	0.017	14x14	1
"Comprehensive-Set" Test	550	537	97.63	0.143	87x87	0.6

Table 3- Tests results for ***dimension equilization*** scenario

Type of Test	Total samples	True Positive	Percentage	Time elapsed in seconds	No. of squared areas	Rate of interlacing
"Trained-Set" Test	500	465	93	0.011	25	0
"Not-Trained-Set" Test	50	46	92	0.002	8	0.3
"Comprehensive-Set" Test	550	508	92.36	0.057	57	0.1

Depending on the *"Comprehensive-Set" test* of the three scenarios, the highest recognition rates are obtained from the second scenario, i.e. the bilinear interpolation followed by Sobel operators, which gives the recognition rate of 97.63%. The detailed results for this second scenario for each class of the "Trained-Set" test, "Not-Trained Set" test and "Comprehensive-Set" test are shown below in Tables-4, 5 and 6 respectively. As shown in Table-2, and since the highest recognition rates for the "Comprehensive-Set" test of the second scenario are gained when the number of features equals to 87x87 with rate of interlacing of 0.6, then, the detailed results of "Trained-Set" and "Not-Trained Set" tests when the number of features equals to 87x87 with rate of interlacing of 0.6 are depicted in Figures-8, and 9 respectively.

Table 4- Details of "Trained-Set" tests results for bilinear interpolation followed by Sobel operators scenario when the number of interlaced areas is 92x92 with rate of interlacing of 0.2.

Class Number	Number of tested samples per class	passed	percentage
0	50	50	100
1	50	50	100
2	50	50	100
3	50	50	100
4	50	50	100
5	50	50	100
6	50	50	100
7	50	49	98
8	50	50	100
9	50	50	100
Total	500	499	99.8

Table 5- Details of "Not-Trained-Set" tests results for bilinear interpolation followed by Sobel operators scenario when the number of interlaced areas equals to 14x14 with rate of interlacing of 1.

Class Number	Number of tested samples per class	passed	percentage
0	5	5	100
1	5	5	100
2	5	4	80
3	5	5	100
4	5	4	80
5	5	4	80
6	5	4	80
7	5	5	100
8	5	4	80
9	5	3	60
Total	50	43	86

Table 6- Details of "Comprehensive -Set" tests results for bilinear interpolation followed by Sobel operators scenario where number of interlaced areas is 87x87 with rate of interlacing of 0.6.

Class Number	Number of tested samples per class	passed	percentage
0	55	55	100
1	55	55	100
2	55	53	96.36
3	55	54	98.18
4	55	54	98.18
5	55	54	98.18
6	55	54	98.18
7	55	50	90.90
8	55	55	100
9	55	53	96.36
Total	550	537	97.63

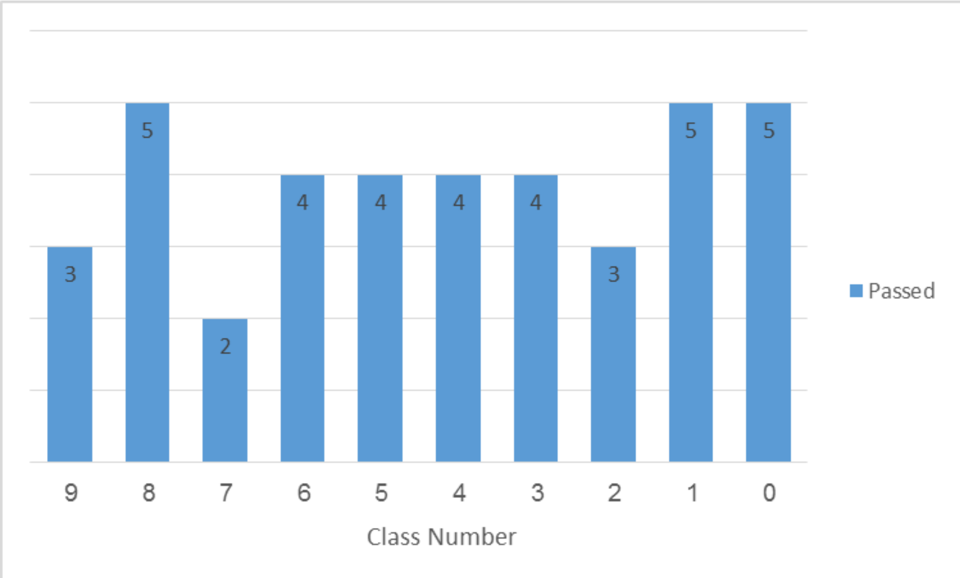


Figure 8- Results of "Not-Trained-Set" Test when number of interlaced areas equals to 87x87 and the rate of interlacing is 0.6

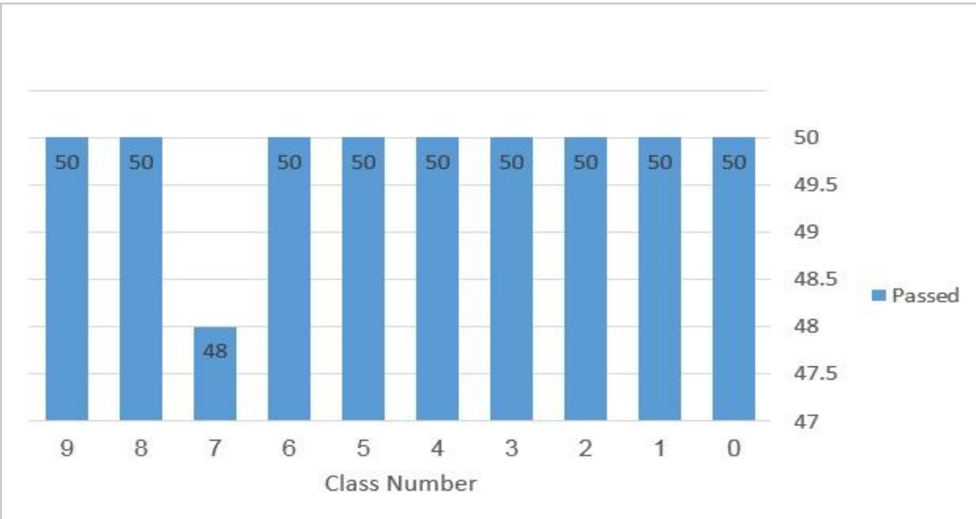


Figure 9- Results of "Trained-Set" Test when number of interlaced areas equals to 87x87 and the rate of interlacing is 0.6

Conclusions and Feature work

In this paper, three scenarios are implemented and tested to recognize handwritten English numerals. The first scenario is the bilinear interpolation of the image, the second scenario is bilinear interpolation followed by Sobel operators, and the third scenario is dimension normalization.

Three types of tests were conducted for each of the three used scenarios. The test which includes all the images involved in the training process called the "Trained-Set" test. The test that includes all the images that are not included in the training process called the "Not-Trained-Set" test. The "Comprehensive-Set" tes contains all the images of a certain class, whether they were involved in the training process, or not. The classification, which used to determine which scenario is the best scenario and gives the highest recognition rates, depends on the results that are obtained from the "Comprehensive-Set" test. So, depending on that test, it has been noticed that the best recognition rates are attained when the second scenario of bilinear interpolation followed by Sobel operators is used.
Since the second scenario gives the highest recognition rates, and in addition to the "Comprehensive-Set" test results, the "Trained-Set" test and the "Not-Trained-Set" test are show in details too.

From the detailed results for the second scenario, one can conclude that the numeral that gives the recognition rate of 100% is are "0" and "1". The recognition algorithm that used to recognize the handwritten numeral called "local pixels distribution". This Algorithm includes the slicing of the image of the handwritten numeral into squared areas that are interlaced for certain proportions. From the results of the "Comprehensive-Set" test for the second scenario, one can conclude that the best number of the squared areas equals to 87x87 and the best rate of interlacing is 0.6. Also, from the obtained results of different tests of numeral samples: "Trained-Set", "Not-Trained-Set" one can conclude that the recognition algorithm of local pixels distribution is an efficient algorithm with rate of recognition reaches to 97, 63%.

From the different types of tests one can conclude that the rate of interlacing must not exceed 0.6 for the three used scenarios. This local pixels distribution is a fast recognition algorithm, because when the second scenario was used with this algorithm, it gives a discrimination rate that not exceed 0.215 second. Even the highest recognition rates are attained with the second scenario of bilinear interpolation followed by Sobel operators, but the results show that dimension normalization is the fastest scenario, then the bilinear interpolation scenario comes in the second place. Also, from the obtained results one can conclude that bilinear interpolation scenario and dimension normalization scenario come after the scenario of bilinear interpolation followed by Sobel operators in the obtained recognitions rates.

This Local pixels distribution can be used in other future researches for the recognition of handwritten character of many languages, signature recognition, or certain shapes detection.

References

1. De Campos, T. E., Babu, B. R. and Varma, M. **2009**. Character Recognition in Natural Images, VISAPP conference, pp: 273-280.
2. Shah, F. T. and Yousaf, K. **2007**.Handwritten Digit Recognition Using Image Processing and Neural Networks, Proceedings of the World Congress on Engineering, 1, London, UK.
3. Rajput, G. G., and Mali, S. M. **2010**. Marathi Handwritten Numeral Recognition using Fourier Descriptors and Normalized Chain Code, *IJCA Special Issue on Recent Trends in Image Processing and Pattern Recognition*, pp:141-145.
4. Dongre, V. J., and Mankar, V. H. **2013**. Devnagari Handwritten Numeral Recognition using Geometric Features and Statistical combination Classifier, *International Journal on Computer Science and Engineering (IJCSE)*, 5(10), pp: 856-863.
5. Abed, M. A. **2014**. Improving Handwritten Isolated Arabic characters Recognition with Particle Swarm Optimization Algorithm, *Diyala Journal For Pure Sciences (DJPS)*, 10(2), pp:18-34.
6. Acharya, T. and Ray, A. K. **2005**. *Image processing principles and applications*, A John Wiley & sons, MC., USA, pp:135-136.