

Real-Time LCD Digit Recognition System

N. M. Saad*, N. S. M. Noor, A. R. Abdullah, O. Y. Fong, N. N. S. A. Rahman

Center for Robotic and Industrial Automation, Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia, Tel: +606
555 2000 / Fax: +606 555 2001

*Corresponding author, e-mail: norhashimah@utem.edu.my

Abstract

In recent years, the utilization of digital instruments in industries is quickly expanding. This is because digital instruments are typically more exact than the analog instruments, and easier to be read as they are hooked up to a liquid-crystal display (LCD). However, manual data entry from LCD display is tedious and less accurate. This paper proposes a real-time LCD digit recognition system for the industrial purposes. The system is interfaced with an IP webcam to capture the video frames from the LCD display. The digital data is pre-processed into grayscale and being cropped into a selected region of interest (ROI). Adaptive thresholding and morphological operation are applied for the digit segmentation process. Data extraction and characterization are done by utilizing neural network classifier. Finally, all the information are logged out to Microsoft Excel spreadsheet. The 90% accuracy is accomplished for 50 test images of various LCD display.

Keywords: LCD Digit Recognition; Adaptive Thresholding; Neural Network; Data Logger

Copyright © 2017 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

The 50-year march of Moore's Law has led to the creation of a relatively cheap and increasingly easy-to-use worldwide digital infrastructure of computers, mobile devices, broadband network connections, and advanced application platforms. This digital infrastructure has, in turn, accelerated the emergence of new technologies that enable transformations in how we live and work, how companies organize, the structure of entire industries and common research institutions [1, 2]. Since users can directly obtain the readings in digital form, these digital instruments are free from parallax and approximation errors.

In process control industries, digital instruments are used to measure various process parameters, such as temperature, voltage, pressure and current. Digital instruments that hooked up with a liquid-crystal display (LCD) are extremely common in the industry. However, manual data entry is time-consuming, labor-intensive and less accuracy [3]. Some process parameter needs to be monitored for the long-term duration, the operators have to work overtime to monitor and record the measured values.

It is useful to save the recorded data into a database, especially for long term monitoring process [4, 5]. Recent developments in computer vision and pattern recognition make it feasible to assist these individuals in developing a real-time computer vision based technology for LCD digit recognition and data logging systems [6]. The advantage of areal-time system task offered areal-time requirement such as deadline and period [7]. Furthermore, in this digital era of modern technologies, numerous element of this new computerized life requires various interactions with the machines [8]. Hence, the necessary actions can be taken to improve the process [9].

The objective of this project is to develop a low-cost LCD display data capturing system, which enables the users to capture the measured digit readings from the LCD display of their device to real-time display on the computer instead of depending on manual data recording approach. This system can be contributed as a data capturing system for most of the digital instruments used by industries and it is suitable for the long-term monitoring purpose in an industry.

The video of LCD display will be taken by using an IP webcam. After the system recognizes the reading on the LCD display, the information on the LCD display will be converted

in data form and cropped into the selected region of interest (ROI) before the adaptive threshold and morphological segmentation process. Neural network classification is utilized for digit recognition. The recognized values will be exported to a Microsoft Excel spreadsheet for graph plotting and further analysis.

2. Research Method

2.1. System Design Flow

Figure 1 shows the flow process of the analysis. The framework begins by capturing the image from the data screen, followed by displaying the area of data identification. After that, the data are segmented using adaptive thresholding followed by image noise filtering. The filtered data is detected and recognized before extracting out into Microsoft Excel.

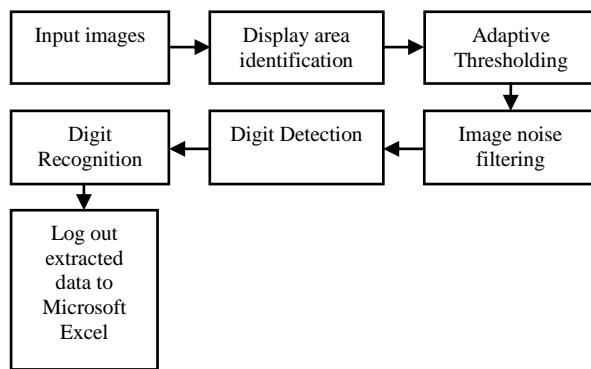


Figure 1. System design flow

2.2. Input Images

The image of the digital data is obtained from the LCD screen display. 50 samples of images are taken by capturing the recorded video frames using an IP webcam.

2.3. Display Area Identification

Figure 2 shows the display area identification. The area identification process is divided into two parts which are wanted and unwanted ROI. The coordinates of the cropping rectangle are then set and are used to crop the following input frames automatically [4].



Figure 2. ROI of the first frame is selected

Figure 3 shows an unwanted symbol that appears on the LCD display. The function of the unwanted region is to remove the uninformative symbol that may influence the result of data extraction. The location of the unwanted symbol is set by using 'roipoly', a user is required to specify the region of the unwanted symbol by selecting the vertices of the polygon. Finally, double click inside the polygon to remove the unwanted region by using the 'region fill' function.



Figure 3. The unwanted symbol is selected by user using 'roipoly'

Similarly, with the previous step, the unwanted region in the following input frames is removed automatically based on the specified coordinates.

2.4. Adaptive Thresholding

Adaptive thresholding is an important task of converting grayscale images into a binary image [4, 10]. For instance, in augmented reality, input images must be segmented to locate known markers in the scene that are used to dynamically establish the position of the camera. This technique is selected due to its capability in analyzing the image in a fast and simple way. Adaptive thresholding considers spatial variations in brightness [11]. The adaptive thresholding technique is applied by using an integral image approach because it is suitable for uneven brightness [12].

An integral image which also known as a summed-area table is used whenever the function is obtained from pixels to real numbers $f(x, y)$ and the sum of this function is calculated over a rectangular region of the image. To compute the integral image, each location of $I(x, y)$, the sum of all of (x, y) terms to the left and above the pixel is stored. This is accomplished in linear time using the following equation for each pixel.

$$I(x, y) = f(x, y) + I(x-1, y) + I(x, y-1) - I(x-1, y-1) \quad (1)$$

Once the integral image is obtained, the sum of the function for any rectangle with the upper left corner (x_1, y_1) , and lower right corner (x_2, y_2) is computed in constant time using the following equation.

$$f(x, y) = I(x_2, y_2) - I(x_2, y_1-1) - I(x_1-1, y_2) + I(x_1-1, y_1-1) \quad (2)$$

The steps to get a binary image from a grayscale image are shown as follows:

1. The average of a $s \times s$ window of pixels centered on each pixel is calculated. It accounts pixels at the surrounding of all sides, so it is a better average for comparison. The average computation is accomplished in linear time by using the integral image.
2. The integral image in the first pass through the input image is calculated.
3. In a second pass, the average using the integral image is calculated for each pixel in constant time and then perform the comparison.
4. If the value of the current pixel is t percent less than this average then it is set to 0 which is black. Otherwise, it is set to 1 which is white.

2.5. Image Noise Filtering

After obtaining the binary images, the images may include some noises that need to be removed [5]. A morphological noise filter is used to filter the noises. From this designed system, salt and pepper noises from the binary images are removed by applying the averaging filter approach which is 'imfilter' from the Matlab toolbox.

```
Mask = fspecial('disk', radius);
Output = imfilter(Input, Mask, 'replicate');
imshow(Output);
```

Mask = fspecial('disk', radius) returns a circular averaging filter within the square matrix of size $2 * \text{radius} + 1$. While the next coding: Output = imfilter (Input, Mask, 'replicate') is to assume the input array values outside the bounds of the array are to equate the nearest array border value.

2.6. Digit Detection

An identification of similar records is one of the important tasks in digit detection, especially for the entity matching applications [13]. After the image is enhanced, the digits in the image are detected by undergoing the following algorithm:

1. Firstly, determine the size of the image, $[m, n]$. Where m is the number of rows, and n is the number of columns.
2. For the coordinates of the first digit, sum up the pixel value for each column for $0 < x < n$.
3. To find the first x coordinate, x_1 , we should find the first column with a summation of the total pixel value is greater than 0. That is the first column with the presence of white pixel detected.
4. For the next x coordinate, x_2 , the step 2 and step 3 is repeated, but it is starting from $x_1 < x < n$.
5. Next, to find y coordinates: y_1 and y_2 , sum up the pixel value for each row: $0 < y < m$ in between x_1 and x_2 .
6. Find out all number of rows with a summation of pixel value greater than 0. Then find the minimum row number, which is the top y coordinates, y_1 . Next, find the maximum row number, which is the bottom coordinates, y_2 .
7. Store the ROI coordinates of the first detected digit as $ROI1 = [x_1, x_2, x_2-x_1, y_2-y_1]$, and then crop the first digit based on $ROI1$.
8. Repeat the steps above for the following digits.
9. Stop when there is no more digit detected.

2.7. Digit Recognition using Neural Network

There are many techniques used to recognize characters, such as template matching, axis distribution analysis, resolution reduction techniques and neural networks [14, 15]. A Neural network approach is used to recognize the seven segmented digits in the images. This is because of the robustness, non-linear input-output mapping, and higher accuracy. There are two modes in the neural network, which is training mode and prediction mode. In the training mode, a 2-layer feedforward supervised neural network as shown in Figure 4 is used.

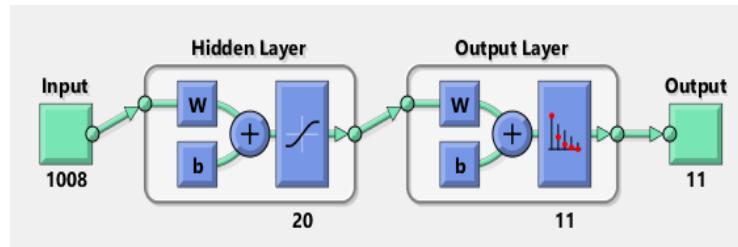


Figure 4. The schematic of implemented neural network model

There are ten training images for each digit, from 0 to 9 and including a negative sign. Therefore, there are total 11 characters to be recognized. Every training images are resized to 24×42 pixels and then arranged into a row vector of 1008 entries. Hence, there are total 110 such input vectors. For the target matrix, it includes 11 labels for 110 training images. Each label represents a character, and each character has ten training images.

For prediction mode, the extracted digits are converted into a row vector input to the neural network model, and with the assist of the trained parameters, it maps corresponding output value among the ten digits. The coding used in the prediction mode is as below:

```
Result = sim (net, input);
```

The 'net' is the trained neural network model, and the 'Result' shows the correlation value of each digit, compares all the correlation values and the highest correlation values among the ten digits is printed as the classified digit [4].

2.8. Log Out Extracted Data to Microsoft Excel

The recognized digits are saved in an output array, the array is then logged out to Microsoft Excel spreadsheet by using 'xlswrite' function for further analysis.

3. Results and Analysis

3.1. Display Area Identification

Figure 6 shows the ROI of the grayscale input frame cropped by the user and then automatically resize to 490×205 . Figure 7 shows the images after the unwanted symbol being removed by using 'roipoly'. The same process will be undergone by the following frame images.



Figure 6. The ROI of the grayscale image cropped by the user



Figure 7. The unwanted symbol removed by user using 'roipoly'

3.2. Adaptive Thresholding

In image pre-processing, adaptive thresholding using the integral image is employed. Figure 8 shows the result of adaptive thresholding.



Figure 8. The resulting image after implemented adaptive thresholding

3.3. Image Noise Filtering

In the noise filter process, the image is blurred to get rid of any unwanted details which are the salt and pepper noise by using a multidimensional image filtering approach which is 'imfilter'. The output image after filter noise is as shown in Figure 9.



Figure 9. The resulting image after implemented adaptive thresholding

3.4. Digit Detection

After the image enhancement, digit detection is undergone by using the designed algorithm mentioned in Part 2, section 2.6. The sample result is shown in Figure 10.



Figure 10. The resulting image after implemented adaptive thresholding

3.5. Digit Recognition using Neural Network

The trained neural network model is tested on a set of training image as shown in Figure 11. Ten different training images for each digit and then negative signs converted to a column vector before fed into the neural network. The neural network recognized the digits based on the input matrix.

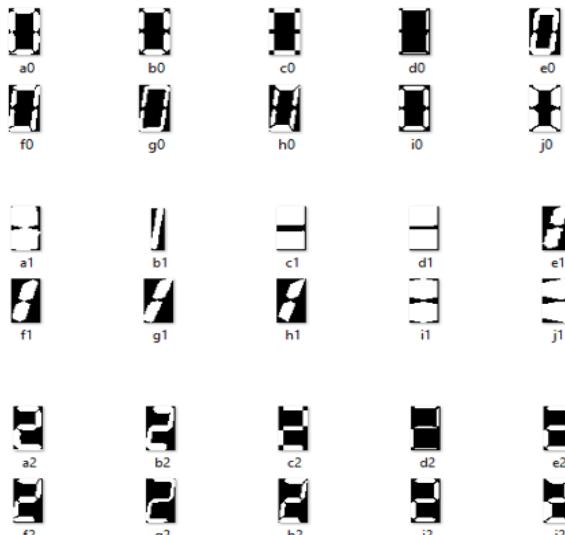


Figure 11. Training image used to train the neural network model

The trained neural network is shown in Figure 12. The iterations are set at 1000 to train, the figure shows that the performance goal is met with only 49 iterations out of 1000 iterations. The two layers which is a hidden layer and output layer feed forward supervised neural network is trained for the digit recognition in this project.

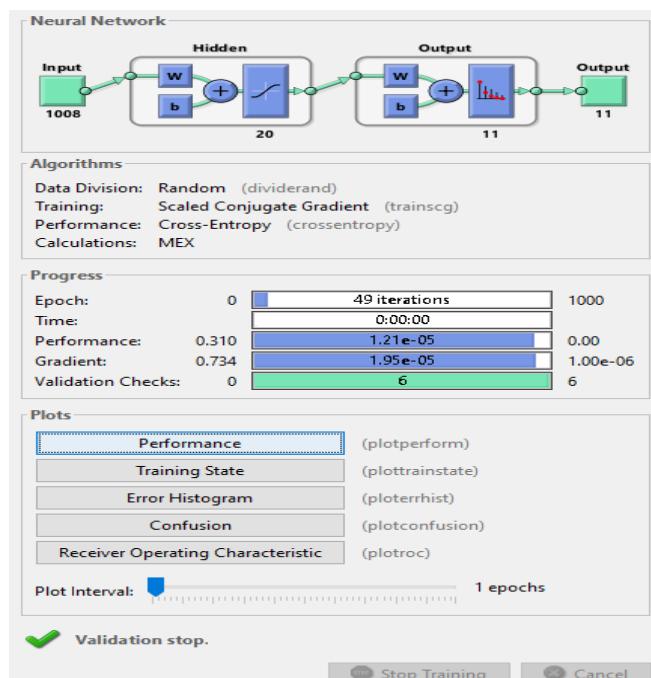


Figure 12. The trained neural network model

The graph of validation performance is plotted as shown in Figure 13. The graph shows that the validation is not much different compare with the test curves. Besides, the test curve does not increased significantly before the validation curve does. Hence, there is no overfitting condition occurs in the training model.

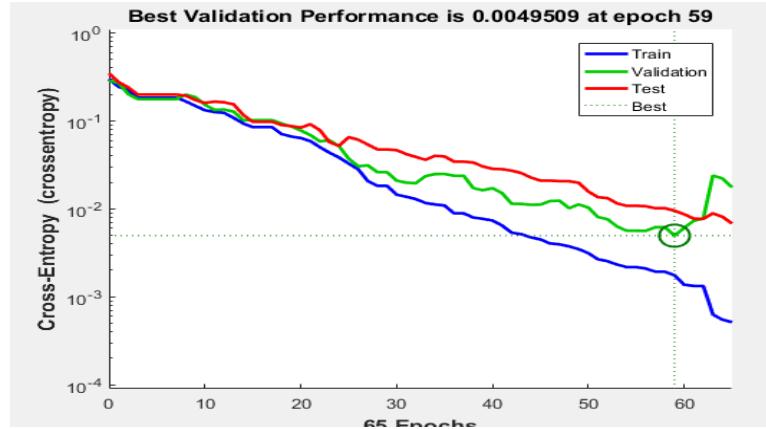


Figure 13. Plotted graph of validation performance

In prediction mode, after the command ‘Result = sim (net, input)’ is executed, the ‘Result’ generated an array to store the calculated correlation value for every label. Next, compare all the correlation values and the highest correlation values among the ten digits are printed as the classified digit. The Figure 14 shows a digit ‘9’ is tested and Figure 15 show a negative sign ‘-’ is being tested.

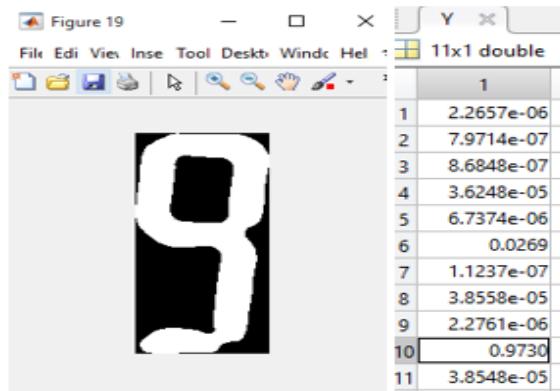


Figure 14. (Left) digit ‘9’ is being tested, (Right) results of the calculated correlation value of digit ‘9’

The label is arranged in sequence: ‘0’, ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, ‘6’, ‘7’, ‘8’, ‘9’, ‘10’, and ‘-’. The digit ‘9’ should be matched to the tenth label. From the Figure 14, the highest correlation value which is 0.9730 is located at the tenth row. While from Figure 15, the negative sign ‘-’ should be matched to the eleventh label, the highest correlation value is 0.9908, which is located in the last row. Hence, both of the digit successfully matches to the correct label. Next, by using ‘if else’, the condition can be set to print out the recognized digits. The recognized value in Figure 16 is tallied with the reading shown in the sample image.

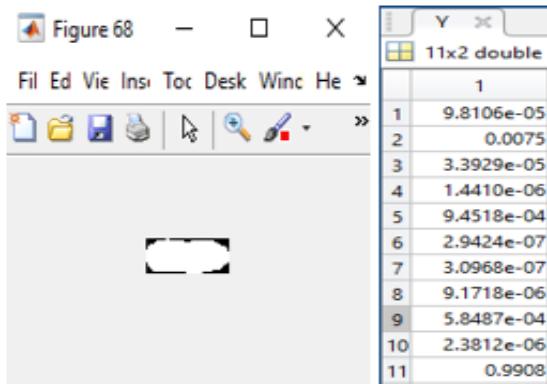


Figure 15. (Left) Negative sign is being tested, (Right) results of the calculated correlation value of negative sign

[abc]	DIGIT	'-093.3'
[abc]	digit1	'-'
[abc]	digit2	'0'
[abc]	digit3	'9'
[abc]	digit4	'3.'
[abc]	digit5	'3'

Figure 16. The result of recognized digits of the trained neural network model

3.6. Log Out Extracted Data to Microsoft Excel

The extracted result is successfully logged out to Microsoft Excel. Figure 16 shows the sample result from Figure 17 that is logged out to a spreadsheet in Microsoft Excel.

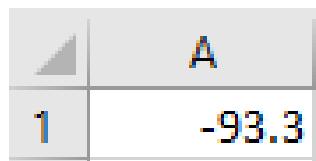


Figure 17. The logged-out data in Microsoft Excel spreadsheet

3.7. Recognition Rate

From Figure 18, the accuracy of the trained model is about 90%. 45 sample images out of the total 50 sample images are recognized correctly, while 5 sample images are recognized wrongly. Three out of the five samples are containing digit '8', where the digit is wrongly recognized as digit '0'. Other two samples are also containing digit '8', but is recognized as digit '6'. The error in image recognition can be improved by increasing the quality of the image captured.

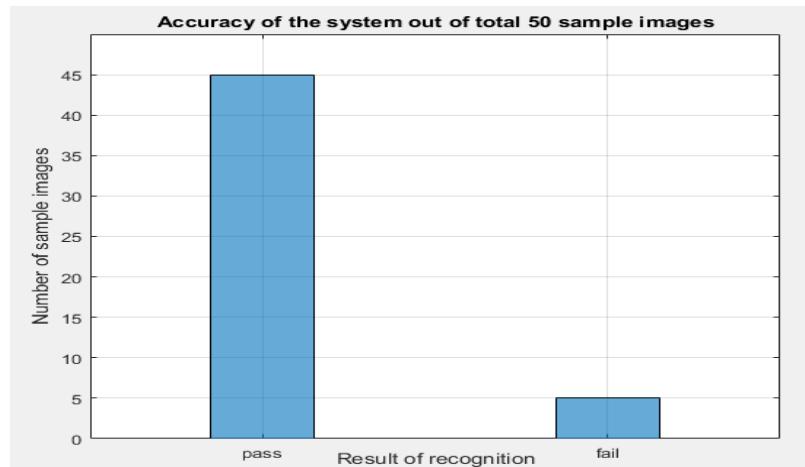


Figure 18. System accuracy out of 50 sample images

4. Conclusion

The system works well for various types of LCD display and under a wide variation of brightness and location. The ROI of the recorded video frames is selected by the user, and the unwanted symbol is also removed initially. An adaptive threshold and morphological segmentation process are applied to segment the digits. Neural network classification is utilized for digit recognition. Besides, this system is flexible because it is using wireless IP webcam to record the video frames. The camera can be put at anywhere if there is a Wi-Fi connection between IP webcam and the PC. The system also provides accurate results for almost all types of seven segment digits' images. In the future works, an algorithm will be designed to track the decimal place in the images. Besides, the system will be improved by increasing the inputs to multiple video frames inputs. Hence, more than one input can be processed by only using one system.

References

- [1] Fichman RG, Dos Santos BL, Zhiqiang (Eric) Zheng. *Digital innovation as a fundamental and powerful concept in the information Systems curriculum*. Mis Quarterly. 2014; 38(2): 329-43.
- [2] Jung SH, Kim JC, Sim CB. Prediction Data Processing Scheme using an Artificial Neural Network and Data Clustering for Big Data. *International Journal of Electrical and Computer Engineering*. 2016; 6(1): 330.
- [3] Saad NM, Rahman NA, Abdullah AR, Syafeeza AR, Noor NS. *Quadratic Distance and Level Classifier for Product Quality Inspection System*. In Proceedings of the International Multi Conference of Engineers and Computer Scientists. 2017; 1.
- [4] Ghosh S, Shit S. *A low cost data acquisition system from digital display instruments employing image processing technique*. In *International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE*. 2014: 1065-1068.
- [5] Li X. LCD/LED Digit recognition by iPhone. Phd Thesis. Texas Tech University. 2012.
- [6] Yi C, Tian Y, Ardit A. Portable camera-based assistive text and product label reading from hand-held objects for blind persons. *IEEE/ASME Transactions On Mechatronics*. 2014; 19(3): 808-17.
- [7] Madenda S, Wibowo EP. Object Feature Extraction of Songket Image Using Chain Code Algorithm. *International Journal on Advanced Science, Engineering and Information Technology*. 2017; 7(1): 235-41.
- [8] Seo J, Kim KH. A Prototype of Online Dynamic Scaling Scheduler for Real-Time Task based on Virtual Machine. *International Journal of Electrical and Computer Engineering*. 2016; 6(1): 205.
- [9] Ghugardare RP, Narote SP, Mukherji P, Kulkarni PM. Optical character recognition system for seven segment display images of measuring instruments. In *TENCON 2009-2009 IEEE Region 10 Conference, IEEE*. 2009: 1-6.
- [10] Bakar NA, Nawawi MR, Abdullah AR, Noordin A, Musa Z, Xian OC. *Malaysian vehicle license plate recognition using double edge detection*. In *2012 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. IEEE. 2012: 422-426.

- [11] Hashim NM, Ibrahim NA, Saad NM, Sakaguchi F, Zakaria Z. Barcode recognition system. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*. 2013; 2(4): 278-83.
- [12] Bradley D, Roth G. Adaptive thresholding using the integral image. *Journal of graphics, gpu, and game tools*. 2007; 12(2): 13-21.
- [13] Alenazi SR, Ahmad K. Record Duplication Detection in Database: A Review. *International Journal on Advanced Science, Engineering and Information Technology*. 2016 Dec 25;6(6).
- [14] Bonačić I, Herman T, Krznar T, Mangić E, Molnar G, Čupić M. *Optical Character Recognition of Seven-segment Display Digits Using Neural Networks*. In 32st International Convention on Information and Communication Technology, Electronics and Microelectronics, <http://morgoth.zemris.fer.hr/people/Marko.Cupic/files/2009-SP-MIPRO.pdf>, last access 2015; 3.
- [15] Arora M. Neural network based Numerical digits Recognition using NNT in Matlab. *International Journal of Computer Science and Engineering Survey*. 2013; 4(5): 19.