



AN INTELLIGENT MODEL FOR CLASSIFICATION OF PRODUCTION PARTS OR PART FAMILIES USING IMAGE PROCESSING AND NEURAL NETWORK IN WARE HOUSE OR STORAGE OF INTELLIGENT FACTORIES

Samad. Dadvandipour, *aitsamad@uni-miskolc.hu* University of Miskolc, 3515, Hungary

Farid. Alidoost, *info@alidoost.ir* Research& Development Centre of Moj Pardazan RAD Co.,
Tabriz, Iran

ABSTRACT

In this paper, we have simulated an intelligent model for identifying finished manufactured parts which are whether used as an individual unit or assembling product parts and delivered to the storage or ware houses applying image processing and Artificial Neural Networks detecting techniques. In the proposed model, we have used the appropriate method of edge extraction (such as: applying Laplace and Sobel mathematical models), where each part is separated and detected, then applying ANN, it is identified for further steps. The paper emphasizes the implementation of the proposed model considering, both “parts images” and “accurate” identifying of finished product parts crossing the line towards ware house or storage using ANN and image processing systems. The most important aim of the paper that we have been seeking for-are- to increase the efficiency of the model for faster and more accurate detection of the parts or part families which are finished and as we mentioned they are either ready to be used as individual unit or assemble ones to the other production parts as well as kept to be used as spare parts or for any other application processes.

Keywords: Classification, Identification, Edge Extraction, Image Processing, Morphologic Operations, ANN, Ware house.

1. INTRODUCTION

Nowadays, companies encounter with a hardly changing environment and higher degree of competition. Thanks to the recent developments in data acquisition, processing, and process control systems, efficiency of many of the industrial applications (i.e. automobile, electronics, rock, and metal industries) has been improved with the help of automated visual processing and classification systems [1]. Computer vision is the science and technology of machines that is able to visualize. The machine is made by integration of many parts to extract information from an image in order to solve some tasks. As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from



images. Each of the application areas described above employ a range of computer vision tasks; with more or less well defined measurement or processing problems, which may be solved using proper methods. Some examples of typical computer vision tasks are presented below.

Recognition is the classical problem in image processing, and machine vision. It is related to the determination of whether or not the image contains some specific objects, features, or activities. This operation can normally be solved robustly and without effort of a human, but is still not satisfactorily solved in machine vision for the general case, involving arbitrary objects in arbitrary situations. The existing methods for dealing with this task can at best, solve it only for specific objects, such as simple geometric objects, human faces, printed or hand written characters, or vehicles, and in specific situations, typically described in terms of well-defined illumination, background, and pose of the object relative to the camera [1],[2],[4],[6], [8], [11]. ANN, as a problem-solving tool which mimics the process of human brain reactions has become an alternative method to recognize an image through training [7]. MATLAB is the abbreviation of matrix laboratory, which has several hundred built-in functions packages and thirty kinds of tool kits. Many design, training and simulation functions of the NN are provided in a NN toolbox.

In this paper, we use the MATLAB ANN Toolbox with LM¹ astray algorithm; also Feed Forward architecture will be examined. In ANN training process, the LM training function has less iteration than traditional BP² and other improved algorithms while the convergence rate is faster and the precision is higher [15].

In feature extraction process, we extract five features for each object [6-7], [10], [12],[14]to recognize objects. Proposed method will use lower inputs to ANN and tends to higher efficiency of vision system. This method is suitable for real time recognition systems compared with previous research [14]; because we can get better iteration time, speed of belt conveyor and accuracy.

2. RELATED LITERATURE

Recognition has an important role in our lives. When a human sees an object, first of all gathers information about the object and compares its properties and behavior with existing knowledge stored in the brain. Recognition of an item mostly done in three important phase: gathering information, feature extraction and classification. As a fundamental role of classification, the features choose the ratio of perimeter squared to the area and ratio of

¹Levenberg-Marquardt



length of the object. Vision system was evaluated in MATLAB using an image database of 45, 80, and 150 images and achieved a recognition rate of 95%, 86%, and 79% respectively. Reference [16] developed a sorting system for bolt and nut using back propagation neural network. The features in that system chose only the shape of the object; therefore the neural network needed one thousand and two hundred data images for the learning process and took more than nine thousand iteration times. Recognition rate was less than 90% and the maximum speed was 5 cm/sec. In this paper, we have developed a neural network for sorting parts (i.e. bolt, nut and screw driver).

There has been developed a software framework for image processing namely the real-time intelligent visual inspection [12] in recognizing the parts in a crossing finished parts line. In this paper the edge detection result is used for morphologic and detection process (finished part line).

Matlab NN toolbox was used to classify the ripeness of papaya based on their mean RGB value components [12]. This system involved image processing and classification using ANN and the threshold rule. In terms of the success rate, ANN can classify up to 93.8% correctly, while the threshold rule only can perform 84.4%.

Finally, Feed Forward was used by means of illustrating special search on pattern identification and curve imitation. This paper also introduces tool functions available in Matlab for Feed Forward and explains how to program within the function. This way It avoids the complexity of advanced language and it is quite easy for users to design and simulate, obtaining satisfactory results. Feed Forward Neural Network has been chosen to determine analyzing module results from edge detection in three status levels for example Bolt, Nut and Screwdriver. In this experiment the acquisition accuracy for moving objects in average is 94% at a maximum speed of 15 cm/s.

3. METHODOLOGY

3.1. Information Collection and Processing

Proposed system collects images on the conveyor belt by high speed video camera, selects bolt, nut and screwdriver as a recognition objects. Because of image quality, real time belt identification and the accuracy requirements, image preprocessing is necessary. First, color image is converted to grayscale color-map, then, to reduce redundant computations the original image gray level is compressed before generating the image co-occurrence matrix [12]. Because we utilize edge detection and morphological techniques to extract feature for objects, we can compress image from 256 to 16 levels.

3.2. Hardware Structure

We use a web-cam (Microsoft Life Cam HD-6000) to capture the objects in conveyor belt. The captured image then will be processed by a program (developed in C#.NET environment) by calling Matlab API's to use image processing and Feed Forward neural network algorithms implemented by Matlab functions, finally sending the output signal to the microcontroller ATmega8. As an actuator, the microcontroller which is connected to a personal computer via USB port in Hd² transfer mode, implemented by (Objective Development Software GmbH, 2011), orders a servo motor's angle of operation, so it can differentiate the bolt, nut and screwdriver in separate places as shown in Fig. 1.

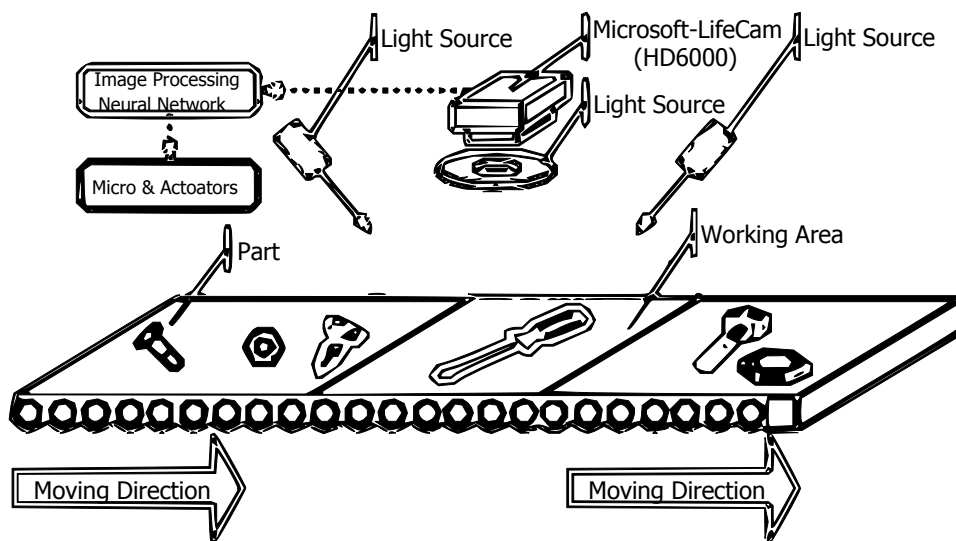


Fig.1- Hardware Structure

3.3. Software Structure

First level in the machine vision is the image processing algorithm [3]. The algorithm is made by adopting an artificial version of human mental action that should be done by humans. Finally it would analyze and extract useful information from the image. The model of software structure is shown in Fig.2.

² Human Interface Device

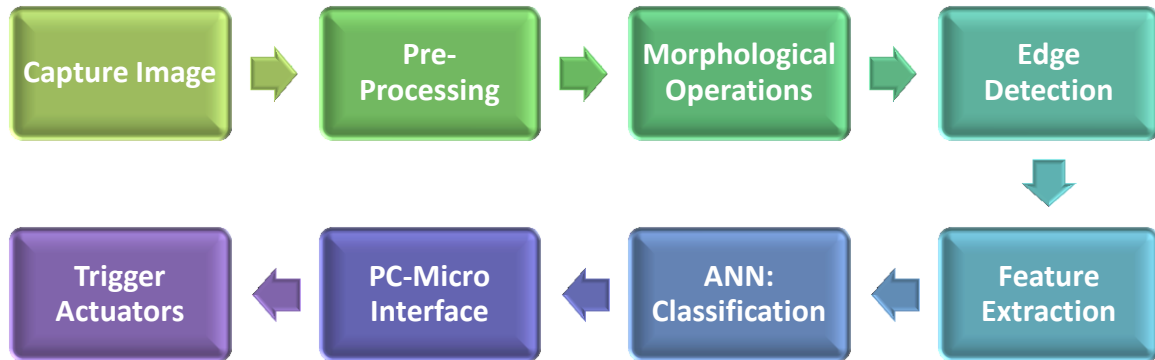


Fig.2. Software Structure Block Diagram and Chart

Fig.2 shows the chart process done through the sequences of algorithms described as following sections.

3.3.1 Capturing Image

Images captured by web-cam are in the RGB³ format, and because we do not need color information, the image is converted into grayscale color map. Mentioned web-cam deliver 4MP images, such an image is so large to be processed; so we decrease image size to a suitable standard image size (i.e. 512*512 pixels). Using the adaptive threshold algorithm, the image is changed into binary form, which is suitable for a further recognizing process.

3.3.2 Image Pre-Processing

Sometimes the target image obtained by camera will produce distortion, which is due to the complex variety of unpredictable circumstances or conditions. Therefore, it is very important to have image preprocessed, and its purpose is to maximize image quality to be processed. In this respect our essential methods include noise reduction, filtering, sharpening, and edge enhancement, etc. After preprocessing, the quality of image brightness, contrast, and visibility has been improved significantly; its features are clear and easy to be identified. The main processing steps used in this article are outlined as follows: Image noise generally produces random, and its distribution and size show irregular. The existence of noise causes image distortion, and directly affects the following treatment. Median filtering, which can effectively suppress noise, is a nonlinear signal processing technology based on the sorting statistical theory. Median filtering, as its name implies,

³ Red-Green-Blue



replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel. It has characters such as fast processing speed and simple operation process, and also shows excellent filtering performance with additive white noise and long tail additive noise. Median filtering can preserve edge and makes away out noise, thus the image can be recovered well [5-9].

The aim is to make the edges and contours blurred image become clear, highlight the details of the image and enhance the details which are fuzzy. We usually use image sharpening treatment based on Laplace transform in time domain. A plan image can be viewed as a function $f(x, y)$ of two variables, its Laplace transform is defined as:

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (1)$$

The partial second-order derivative in the x-direction:

$$\frac{\partial^2 f}{\partial x^2} = f(x + 1, y) + f(x - 1, y) - 2f(x, y) \quad (2)$$

Similarly in the y-direction:

$$\frac{\partial^2 f}{\partial y^2} = f(x, y + 1) + f(x, y - 1) - 2f(x, y) \quad (3)$$

Adding (2) and (3), we obtain:

$$\nabla^2 f(x, y) = [f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y)] \quad (4)$$

This equation can be implemented using the filter mask shown in Fig.3, which gives an isotropic result for rotations in increments of 90°.

| | | |
|---|----|---|
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

Fig.3 - Filter Mask for Isotropic Increment of 90°

Expanding (4) to the 45° isotropic increment, we obtain:



$$\nabla^2 f(x, y) = [f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) + f(x - 1, y - 1) + f(x - 1, y + 1) + f(x + 1, y - 1) + f(x + 1, y + 1) - 8f(x, y)] \quad (5)$$

| | | |
|---|----|---|
| 1 | 1 | 1 |
| 1 | -8 | 1 |
| 1 | 1 | 1 |

Fig.4 - Filter Mask for Isotropic Increment of 45° considering Equation (5)

After the filter mask processed image (shown in Fig. 4), the light gray edges and mutation points are added to the dark background. The original image is superimposed on the Laplace image, thus we not only protected the Laplace sharpening effect, but also restored the original image information, and we obtained sharp image as well. Then we can get the following equation,

$$g(x, y) = f(x, y) - \nabla^2 f(x, y) \quad (6)$$

Where $g(x, y)$, is the new image enhanced and sharpened by Laplace filter mask [13].

3.3.3 Morphological Operations

After pre-processing phase, in order to connect interrupted lines, we apply mathematical morphology dilation command using a SE^4 (line: 7pixel). After this process we clear the holes in image. To eliminate noise coming from the outside line of the conveyor belt, we apply “Image Clear Border Command”, with a value of 5. We have also applied mathematical morphology erosion command, “Image Erosion Detection Command”, using a diamond SE, with a value of 6 for thinning the shape and eliminating the noise. In this stage we can calculate first feature of the object. We define “area” of object to white pixels exists in the image after morphological operations. Morphological operation processes are shown in Fig.5, 6, 7.

⁴ Structural Element

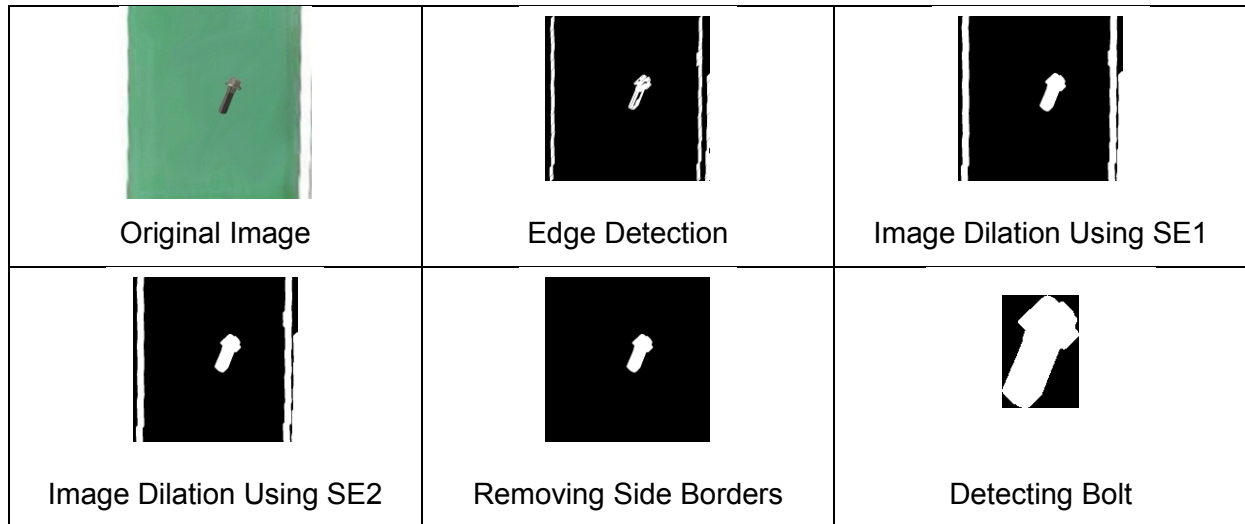


Fig.5. Morphological Operations Process on a Bolt

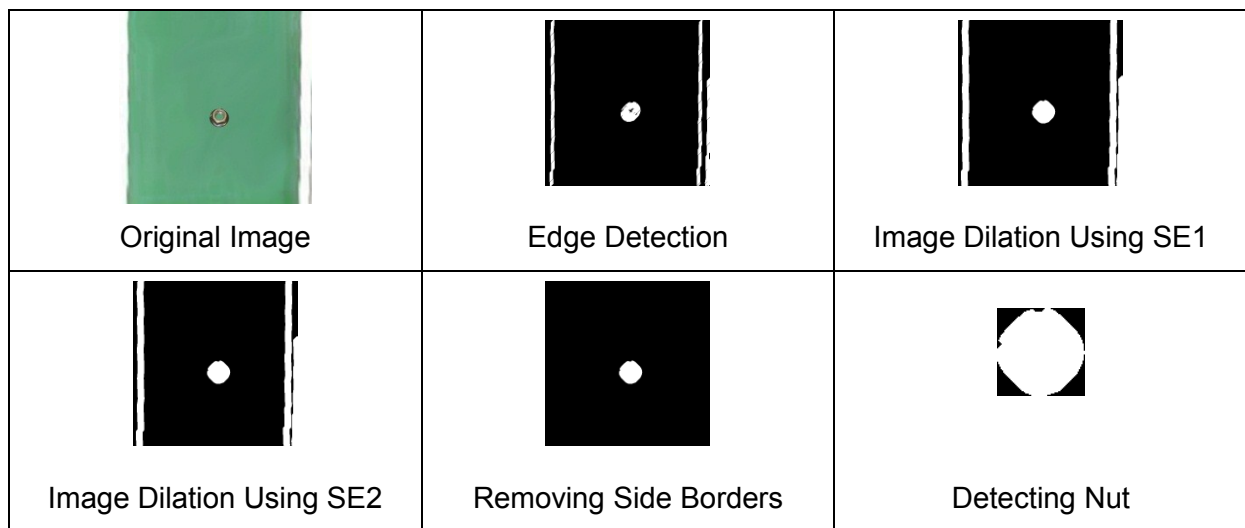


Fig.6. Morphological Operations Process on a Nut

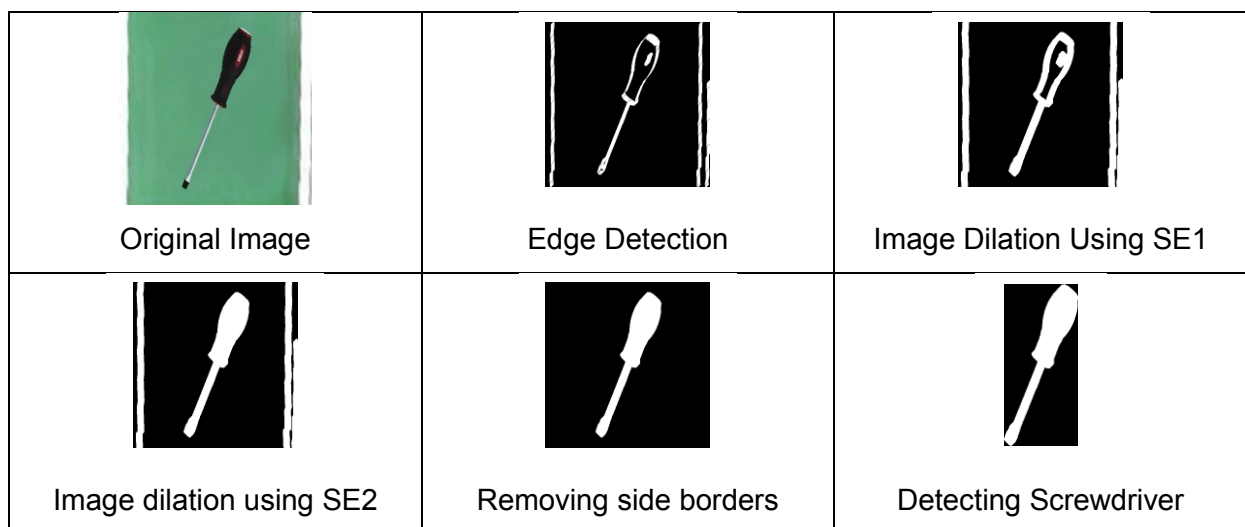


Fig.7– Morphological Operations Process on a Screwdriver



3.3.4 Edge Detection

Image which is processed by the first order differential equation usually produce relatively wide edge, so we use the gradient method based on the first derivative to enhance the edge of image. The approximate gradient of the image $f(x, y)$ is:

$$\nabla^2 f(x, y) \approx |G_x| + |G_y| = \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right| \quad (7)$$

The above equation can be described by 3×3 filter mask shown in Fig.8, and the approximate result is:

$$\nabla f(x, y) \approx \{|(a_7 + 2a_2 + a_9) - (a_1 + 2a_2 + a_3)| + |(a_3 + 2a_6 + a_9) - (a_1 + 2a_4 + a_7)|\} \quad (8)$$

In the above equation, we can find that the difference between the third row and first row is close to the differential in the x-direction, and the difference between the third column and first row is close to the differential in the y-direction, this can be expressed by 3×3 filter mask shown in Fig.9. This filter mask is called Sobel operator [6-7,17]. After treatment by Sobel operator mask, the edge of image will be significantly more intuitive and the processed image is conducive to feature extraction.

| | | |
|-------|-------|-------|
| a_1 | a_2 | a_3 |
| a_4 | a_5 | a_6 |
| a_7 | a_8 | a_9 |

Fig.8 - 3×3 filter mask structure

| | | |
|----|----|----|
| -1 | -2 | -1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |

| | | |
|----|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

Fig.9 - Sobel Operator Filter Mask (left for vertical and right for horizontal edge detection)

Edge detection is done by using the Sobel operator to show the second shape of the object [6-7, 17]. After that we can extract two other features from object. If we calculate count of white pixels in image, we can define second feature as "PERIMETER", also by dividing



“area” to “perimeter” we obtain third feature for objects. Then we define two other parameters as “length” and “width” by using length and width of detected object.

3.3.5 Neural Network Architecture

To implement a Feed Forward Architecture, MATLAB neural network toolbox supplies a new, professional function “newff ()” for neural network establishment. Its grammar is as follows:

$$\text{net}=\text{newff} (\text{Xr}, [\text{S1 S2 ... SN1}], \{\text{TF1 TF2 ... TFN1}\}, \text{BTF}, \text{BLF}, \text{PF}) \quad (9)$$

In the equation above, Xr is an input vector, [S1 S2 ... SN1] expresses, in turn, the unit numbers of the hidden layers and output layers in the Feed Forward network; {TF1 TF2 ... TFN1} represents respectively the transfer functions in the hidden layers and output layer. Functions such as “tansig”, “logsig” and “purelin” and so on, can be used and ‘tansig’ is the default; BTF, which expresses the back train function in the network, is a character string variable and “trainlm” is the default; BLF, which represents the back weigh learning function, is a character string variable and ‘learngdm’ is the default; PF, which expresses the performance function concluding mae (calculating network average absolute error), mse (calculating mean-square error), msereg (calculating mean-square error and the weighting of weight or threshold value) and sse (calculating network mean-square sum), is a character string variable calculating network output error to provide a criterion for training, which chooses ‘mse’ as default; net is a newly creating Feed Forward neural network.

BTF, BLF and PF will be set in terms of requirement, or omitted. The overall Flowchart for Feed Forward training and application is shown in Fig.10. We define an input vector of size 1*5 for neural network, which are features generated in image processing phase. We examine some structures for our Network by some parameters shown in Table1, according to table x, the best results for our network achieved in method No.9, by mse=2.0785e-12.

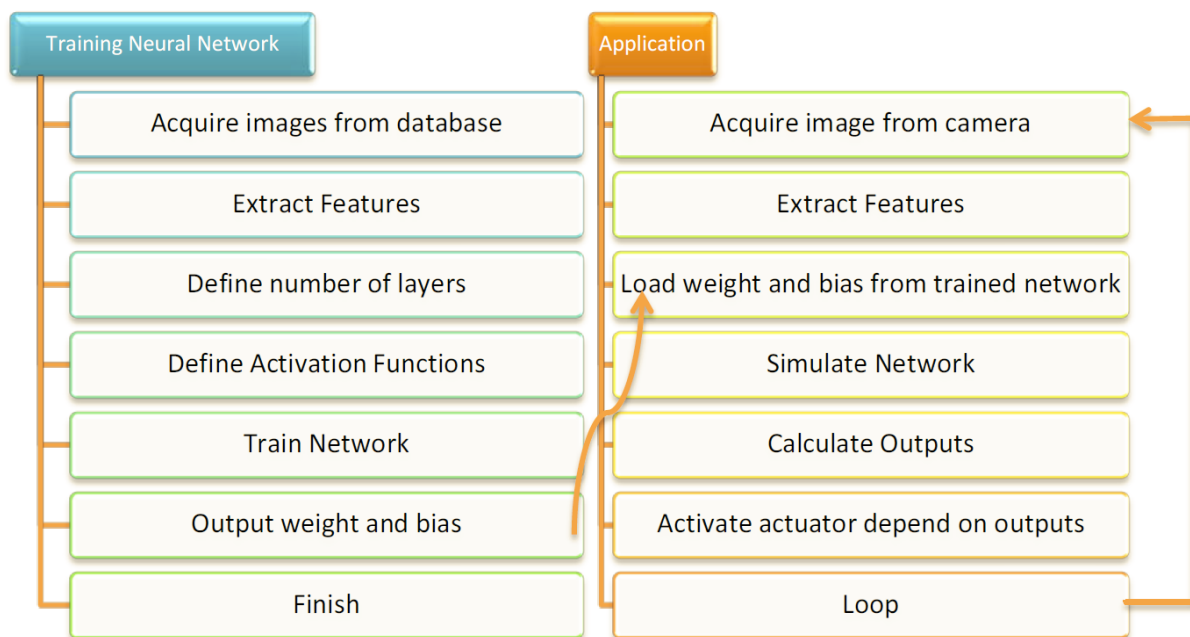


Fig.10 - Training and application flowchart of Neural Network

Table1-The Heuristic Results for Different Hidden Layers and Activation Functions

| Method | Hidden Layers | Neurons | Activation Functions | Iteration | MSE |
|-----------|---------------|--------------|------------------------------|-----------|-------------------|
| 1 | 1 | 5-3 | tansig- purelin | 10 | 0.01118 |
| 2 | 1 | 10-3 | tansig- purelin | 15 | 0.01089 |
| 3 | 1 | 15-3 | tansig- purelin | 23 | 0.00378 |
| 4 | 1 | 20-3 | tansig- purelin | 16 | 0.01932 |
| 5 | 1 | 25-3 | tansig- purelin | 10 | 2.034e-5 |
| 6 | 1 | 30-3 | tansig- purelin | 14 | 0.01009 |
| 7 | 2 | 15-5-3 | tansig-tansig-purelin | 20 | 3.79e-12 |
| 8 | 2 | 10-5-3 | tansig-tansig-purelin | 14 | 0.00774 |
| 9* | 2 | 5-5-3 | tansig-tansig-purelin | 18 | 2.0785e-12 |
| 10 | 2 | 5-3-3 | tansig-tansig-purelin | 21 | 1.765e-7 |
| 11 | 2 | 10-3-3 | tansig-tansig-purelin | 50 | 3.295e-7 |
| 12 | 2 | 15-3-3 | tansig-tansig-purelin | 15 | 2.977e-7 |
| 13 | 1 | 15-3 | logsig-tansig | 13 | 7.73e-12 |
| 14 | 1 | 15-3 | logsig-logsig | 18 | 0.00076 |
| 15 | 1 | 15-3 | purelin - logsig | 27 | 2.18e-10 |

Validation Performance Graph during Training process shown in Fig.11.

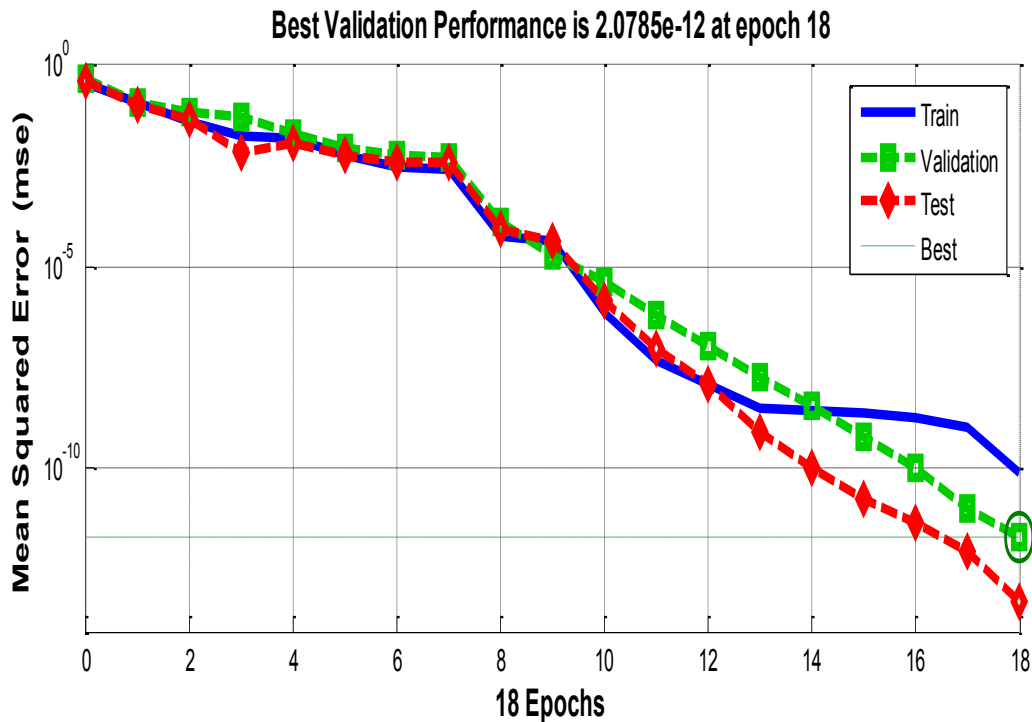


Fig.11 – Best Validation Performance Achieved by method #9 defined in Table 1

4. Conclusion

In comparison with the previous researches, this experiment shows better results in terms of success rate, speed of conveyor belt, and types of bolt and nut and screw drivers. The success rate is 96%, at 10 cm/sec speed, and involving different types of bolt, nut and screw drivers. The success rate can be improved up to 98% if we had used an additional spotlight on the object and better conveyor belt (steady color and movement), because by using those, a clearer and sharpen image with less noise and shadows obtained by camera. Also speed of conveyor belt may be increased in case of using a better servo actuator with lower response time. With this method we can develop an online intelligent classification system, that is able to classify products in groups and also training neural network for new products don't detain production line because training can be done in another place and neural network's weights can be applied in our system to identify new products.



ACKNOWLEDGMENTS

This paper was prepared within the framework of the TAMOP-4.2.1.B-10/2/KONV-2010-0001 project having title “Improvement of the quality of higher education through the development of research-development-innovation-education”. Project takes place by the support of European Union, co-financed by the European Social Fund.

REFERENCES

1. **R. Mattone et al. (2000)**. “Sorting of items on a moving conveyor belt,” Part 1: A technique for detecting and classifying objects. *Robot Comput Integr Manuf*, pp. 16:78–80.
2. **R. Mattone et al. (2000)** . “Sorting of items on a moving conveyor belt,” Part 2: Performance evaluation and optimization of pick-and-place operations. *Robot Comput Integr Manuf*, pp.16:81–90.
3. **Hi. Et al. (2002)**. “Visual processing and classification of items on a moving conveyor,” a selective perception approaches. *Robot Comput Integr Manuf*;18(2), pp. 25–133.
4. **M .Unser. (1986)**. “Sum and difference histograms for texture classification,” *IEEE Trans Patt Anal Mach Intell*; PAMI-8(1), pp.118–25.
5. **BG. Batcheloret et al. (1992)**. “Commercial vision systems. *Computer Vision*,” Theory and Industrial Applications. New York: Springer;.
6. **R .Chellappa. et al. (1985)**. “Digital image processing and analysis,” *IEEE Catalog No: EHO232-9*;
7. **M. Sonka et al. (1993)**. “Image processing, analysis and machine vision,” New York Chapman & Hall.
8. **TS. Newman. Et al. (1995)**.“A survey of automated visual inspection,” *Computing Vision Image and Understanding*, pp. 61–62.
9. **EN. Malamas, et al. (2003)**. “A survey on industrial vision systems,” applications, pp. 21–88
10. **H. Golnabi. Et al. (2007)**. “Design and application of industrial machine vision systems,” *Robot Compute Integr Manuf*, 23(6), pp. 630–7.
11. **Qiao.Tiezhu. (2010). et al.** “A Research of Image Identification Signal Processing for Longitudinal Rip of the Transport Belt,” *International Conference on Optoelectronics and Image Processing*.
12. **Wang. Kun. et al. (2010)**. “Identification Method of Waste Based on Gray Level Co-occurrence Matrix and Neural Network,” *IEEE*,



13. **I. Topalova. et al. (2010).** "Increasing the Image Recognition Accuracy in Machine Vision Systems with Added Noise due to Technological Issues," IEEE 26-th Convention of Electrical and Electronics Engineers,
14. **T. Muhammad. et al. (2011).** "Recognition of Bolt and Nut using Artificial Neural Network," International Conference on Pattern Analysis and Intelligent Robotics.
15. **M. GOPIKRISHNAN. Et al. (2011).**"Effect Of Different Neural Networks On The Accuracy In Iris Patterns Recognition,"International Journal of Reviews in Computing
16. **Dewanto S. Et al. (2005).**"Sistem sortir mur dan baut menggunakan jaringan syaraf tiruan," Jurnal Teknik Komputer, vol. 13, no. 1, pp. 1-13
17. **Rafael C.Gonzalez. Et al. (2002).** "Digital Image Processing", Prentice-Hall Inc., New Jersey