www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 6 Issue 2 Feb. 2017, Page No. 20254-20260
Index Copernicus Value (2015): 58.10, DOI: 10.18535/ijecs/v6i2.17

Increase accuracy the recognition of objects using artificial neural network technology

Mahmoud Mohamed Hamoud Kaid* Muawia Mohamed Ahmed Mahmoud
AL Neelain University, Faculty of Engineering, Control Engineering Department,
Khartoum, Sudan
Mahmoudkaid80@gmail.com

Abstract: *This paper aims to* find offers the possibility of building system software used two techniques to identify the objects one of these techniques technique of artificial intelligence process of nerve cells called artificial neural networks applications, and other technical digital image processing technology using a torque is changing (moments Hugh), where the system will be able to forms of discrimination Engineering regular and irregular artificial neural networks users to reduce the proportion of the mis-identification of objects process and integrate it online with the identification using the torque is changing techniques. The network is trained on those forms for the first time and then the output of the system by giving congruence of any of these forms at high speed and accuracy. When using each technique separately there was a mis ratio clearly and when merging two technologies with some of the proportion of wrongdoing, for objects that have been training completely absent and has to recognize these objects process meticulously.

Keywords: Pictures for digital processing, for moments is variable (Hue moments), artificial neural networks

1. Introduction

Summed up the way distinguish objects and recognition how monitoring systems for the middle of the environment , and its ability to image processing with high accuracy in order to recognize the objects and distinguish them from the background , giving a reasoned decision about the affiliation or non- affiliation shots of samples studied[1].

There are many ways to recognize and distinguish objects in different areas, and using different techniques, but the best are the following methods:

Classic method , analytical , matching templates, and method of artificial neural networks, these methods do not necessarily have to be independent , but can be integrated with each other , so that the technique discrimination is affected by the transition , change size, and rotation required to distinguish the samples.

Selection has occurred on the classic manner and method of artificial neural networks, which relies classic style method of moments non changing in the vector extraction features to resolve the problem of discrimination three-dimensional objects.

Non Changing moments is offering a complete set of descriptors for the image , so it has a fundamental importance in distinguishing objects theory , supports the concept of non- changing moments that deals with the properties of specific varieties of algebraic expressions that remain constant under the influence of general linear transformations.

These features are obtained from two-dimensional images, but when the process of applying this plucks some pictures and found that it does not recognize them as has previously been training them, and over they recognize some of the nearby shapes of images of objects that have been training and this is a problem again , so we used artificial neural networks as a way complementary to it, to solve these problems thus achieved two methods with each other a high rate in recognizing pictures of objects process and found solution for the previous problems.

As the researchers studied cared intense study in recent years and can process a tremendous amount of information Parallel, at the same time can solve problems that can be solved using conventional algorithms also learn from the examples provided to them. Artificial Neural Networks used recently in digital imaging engineering and decision-making, and proved to be effective in the areas of identification and classification of data arrangements [3].

2. Image processing

Identify digital image: as an image $f(y,z)$ has been cut in each of the spatial coordinates and lighting levels. So it can be considered as evidence of an array determines where the line and column position of the point of the image and the value of the matrix element specifies the amount of gray normal at that point. The elements of such a digital matrix called picture elements (Image elements) or (Pixels) or (Picture elements) or Abbreviation (Pels) [2].

Symbolizes the image of a two-dimensional function $f(y,z)$ of the intensity of light, referred to as the aftershock, as the value f at the same spatial coordinates (y,z) of the point gives the intensity (illumination) image at that point. Because the light form of energy, the value of the function $f(y,z)$ must be limited and not equal to zero, it must meet the following relationship number (1):

$$0 < f(y,z) < \infty \quad (1)$$

The basic nature of the function $f(y,z)$ vectors: The first is the amount of light received on the scene that look at it, and the other is the amount of light reflected from objects in the scene, it is applicable to call this the rovers illumination, and the reflectance. It referred to $i(y,z)$, and $r(y,z)$, respectively, as the product of these two functions is according to the relationship's number (2), knowledge of the function:

$$f(y,z) = i(x,y)r(y,z) \quad (2)$$

The picture from the storage file is the image of a true colors with subsequent jpg are represented in the MATLAB environment to represent RGB In this representation is the color of each picture element combination of colors in a single point from three matrices are matrices of primary colors, it is converted to a gray image in order to facilitate processing and enhancing the edges of objects to draw necessary to distinguish them from the surrounding environment information [2].

3. Geometric Moment Invariants

Hue (1962) presented a theory of two-dimensional moment invariants for planar geometric figures based on the work of the 19th century mathematicians Boole, Cayley and Sylvester. In this theory, a set of invariants based on combinations of regular moments using algebraic invariants were derived. These seven moments are invariant to translation, scale, and orientation. He also implemented these moments to recognize the Latin alphabetic characters [5].

Many studies on deriving different types of invariants and using invariant
Moments for object recognition have been carried out since the introduction of Hu's Theory, Alt et. Al compared the performance of moment invariants and Fourier descriptors in recognizing planar shapes [2].

3.1 Object Recognition by Geometrical Moment Invariants

Hue defined seven descriptors which computed from central moments through order three that are independent to object translation, scale and orientation. Translation invariance is obtained by computing moments that are normalized with respect to the center of gravity. The size invariance can be computed from algebraic invariants. From the second and third order values of the normalized central moments, a set of seven invariant moments can be computed which are independent of rotation .Given a two-dimensional image , $f(x,y)$, the moments of order (p q) are defined as relationship number (3):

$$m_{pq} = \sum_{Z=0}^{M-1} \sum_{Y=0}^{N-1} z^p \cdot y^q \cdot f(y,z) \quad (3)$$

Where:

$$p,q = 0,1,2,..,\infty$$

M, N - horizontal and vertical dimensions of the image.

$f(y,z)$: Intensity (gray class) at the point (y,z) of the picture, we get zero torque put p, q = 0 in the relationship (4), we get the following equation:

$$m_{00} = \sum_{Z=0}^{M-1} \sum_{Y=0}^{N-1} f(y,z) \quad (4)$$

The relationship represents the total values of the pixels in the image, and named an area of the image. If the binary image $f(y,z)$ then the image area is equal to the flat area of the sample. It determines the torque of the first class the following equations:

$$m_{10} = \sum_{Z=0}^{M-1} \sum_{Y=0}^{N-1} x \; f(y,z)$$

(5)

$$m_{01} = \sum_{Z=0}^{M-1} \sum_{Y=0}^{N-1} y \; f(y,z)$$

Sample Center is one of the important barometers to determine the position of the sample, is the point at which owns coordinates x', y' which is the sum of squares of which the distance of each other points within the small body. By expressing the center of moments as follows:

$$y' = \frac{m_{10}}{m_{00}}, \qquad z' = \frac{m_{01}}{m_{00}}$$

3.2 Central moments

Correspond to moments of inertia moments sample m_{20} , m_{02}.These moments variable depending on location for the center and scale draws the studied sample, and are therefore of limited use. So, it was a collection of moments to get to know is changing. This group can be derived beginning calculates the central moments μ_{pq} given by the following equation:

$$\mu_{pq} = \sum_z \sum_y (z - z')^p \, (y - y')^q \; f(y,z) \quad (6)$$

According to the previous equation No.6 can be all of the following moments Account:

$$\mu_{10}, \mu_{01}, \mu_{11}, \mu_{10}, \mu_{20}, \mu_{02}, \mu_{12}, \mu_{21}, \mu_{03}$$

You can write a sentence following equations:

$$\mu_{00} = m_{00} \quad ,\mu_{10} = 0 \ , \ \mu_{01} = 0$$
$$\mu_{20} = m_{20} - z'.m_{10}$$
$$\mu_{02} = m_{02} - y'.m_{01}$$
$$\mu_{11} = m_{11} - y'.m_{10}$$
$$\mu_{12} = m_{12} - 2y'.m_{11} - z'.m_{02} + 2y'^{2}.m_{10} \quad (7)$$
$$\mu_{21} = m_{21} - 2z'.m_{11} - y'.m_{02} + 2z'^{2}.m_{01}$$
$$\mu_{03} = m_{03} - 3y'.m_{02} + 2y'^{2}.m_{01}$$
$$\mu_{30} = m_{30} - 3z'.m_{20} + 2z'^{2}.m_{10}$$

Then the development of the central moments measured η_{pq} as follows:

$$\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{\lambda}} \ ; \ \lambda = \frac{p+q}{2} + 1 \ ; \ p+q \geq 2 \qquad (8)$$

3.3 Moments Hue Non-changing

These parameters can define a set of moments non-changing as follows [5].

$$\phi_1 = \eta_{20} + \eta_{02}$$
$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$
$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$
$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$
$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + \qquad (9)$$
$$(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$$
$$\phi_6 = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$
$$\phi_7 = (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})\right]$$
$$+ (3\eta_{12} - \eta_{30})(\eta_{03} + \eta_{21})\left[3(\eta_{30} + \eta_{12}) - (\eta_{21} + \eta_{03})\right]$$

These moments non-changing form a complete set to describe the image, which provides the possibility to recognize the samples with high accuracy. The torque account for a set of images of samples trainer by the program, and a safe in advance to be a library, the program compares the moments calculated for the current image coming from the image capture with moments stored and based on the result of the comparison program gives decision to recognize or failure to identify the sample, which is done by calculating the distance.

3.4 Mechanism of action to recognize and discernment algorithm

After obtaining the previous moments, the recognition algorithm and discrimination compared between the vector moments of Hue to the image of a sample with a vector of income plucks each image sample stored in a comprehensive public library, using the technique discrimination Euclidean distance given to the following relationship.

$$D_{I} = \sqrt{\sum_{j=1}^{7}\left(\phi_j - \phi_j^{I}\right)^{2}} \qquad (10)$$

Where:

I - Represents a line guide (Image guide) in the library.

D_I - The distance between the vector and the image vector moments income reference ϕ_j^I stored in the line I of the library.

Φj - Moments image income vector $\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7$.

The distance between the vector moments calculated image input ϕj and all of the reference vector

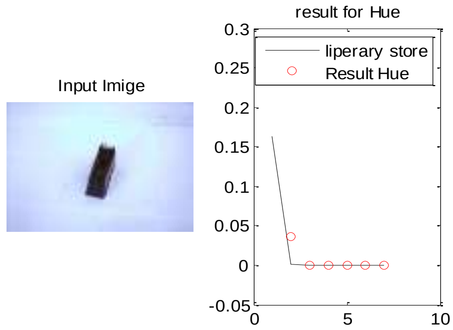


Figure (3)

ϕ_j^I stored in the library and then Minimal distance from the vector and compares with a certain distance threshold is taken , and based on the result of the comparison with the decision taken was a picture of input belongs to the samples intern them or not.

If the sample belongs to the objects that have been trained under the program, the program gives a signal that it was arrested on the target as shown in Figure (1), unless the program is due to take a picture of new income and processes it.

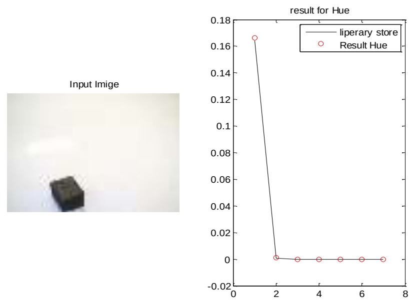


Figure (1)

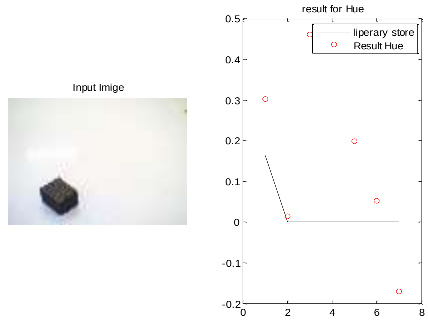


Figure (2)

Through the application of the program and found that it inaccurately high percentage, we noticed the

DOI: 10.18535/ijecs/v6i2.17

presence of the samples did not recognize the program although it was trained them as shown in Figure (2), and more it gives positive results for some of the forms that have not been trained them as shown in Figure (3), and to overcome these problems and improve the work of the program and increase the accuracy of the results our use of artificial neural networks, where this technique has a high ability to solve these problems and give excellent results.

4. Artificial neural network

Among countless number of neural network structures, there are two that are used more often than all others: the multilayer perceptron (MLP) and the radial basis function (RBF). These networks have been extensively studied, empirically tested on a broad spectrum of applications, and hence their properties are now well known. Furthermore, these two types of networks have proven to be universal approximates a term referring to the ability of these networks to approximate any decision boundary of arbitrary dimensionality and arbitrary complexity with arbitrary precision, given adequate amount of data [6].

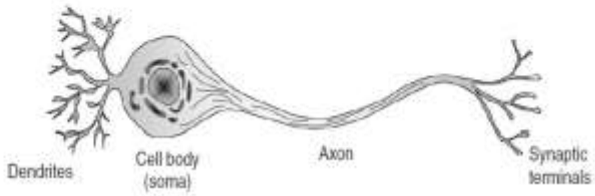


Figure (4) Schematic illustration of the neuron

4.1 Neuronal Model and the Multilayer Perceptron.

The artificial neural networks ANN or simply neural networks are designed to mimic the decision-making ability of the brain, and hence their structure resembles that of the nervous system, albeit very crudely, featuring a massively interconnected set of elements. In its most basic form, a single element of the nervous system, a neuron, consists of four basic parts, as schematically illustrated in Figure(4): the dendrites that receive information from other neurons, the cell body (the soma) that contains the nucleus and processes the information received by the neuron, the axon that is used to transmit the processed information away from the soma and toward its intended destination [6].

4.2 Transfer Functions

Many transfer functions are used in neural network. All of the mathematical transfer functions can be realized with a function having the same name. The sigmoid transfer function shown below takes the input, which may have any value between plus and minus infinity, and squashes the output into the range 0 to 1. This transfer function is commonly used in backpropagation networks, in part because it is differentiable [8].

4.3 Neuron with Vector Input

A neuron with a single R-element input vector is shown below. Here the individual element inputs p_1, p_2, ... p_R are multiplied by weights w_1, 1, w_1, 2, ... w_1, R and the weighted values are fed to the summing junction. Their sum is simply Wp, the dot product of the (single row) matrix W and the vector p.

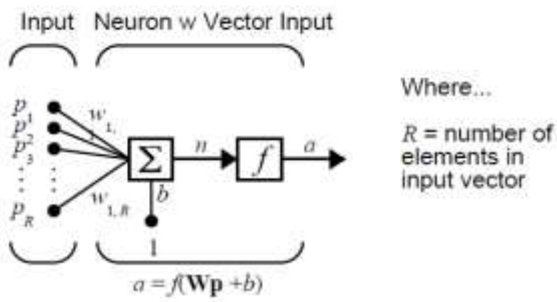


Figure (5)

The neuron has a bias b, which is summed with the weighted inputs to form the net input n. This sum, n, is the argument of the transfer function f [7].

4.4 Training a Network

Conceptually, a network forward propagates activation to produce an output and it backward propagates error to determine weight changes (as shown in Figure 6). The weights on the connections between neurons mediate the passed values in both directions.

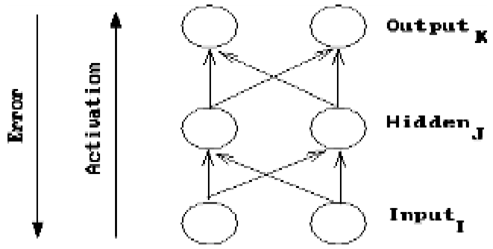
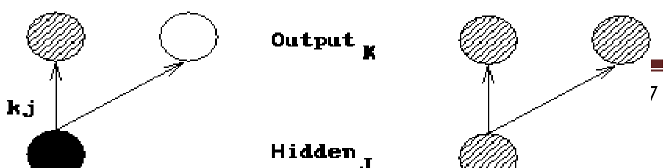


Figure (6)

The Backpropagation algorithm is used to learn the weights of a multilayer neural network with a fixed architecture. It performs gradient descent to try to minimize the sum squared error between the network's output values and the given target values.

Figure (7) depicts the network components which affect a particular weight change. Notice that all the necessary components are locally related to the weight being updated. This is one feature of backpropagation that seems biologically plausible. However, brain connections appear to be unidirectional and not bidirectional as would be required to implement backpropagation.

The change to a hidden to output weight depends on error (depicted as a lined pattern) at the output node. While the change to a input to hidden weight depends on error at the hidden node (which in turn depends on error at all the output nodes) and activation at the input node.



4.4.1 Notation

For the purpose of this derivation, we will use the following notation:

- The subscript **k** denotes the output layer.
- The subscript **j** denotes the hidden layer.
- The subscript **i** denotes the input layer.
- ***w_{kj}***: denotes a weight from the hidden to the output layer.
- ***w_{ji}***: denotes a weight from the input to the hidden layer.
- ***a***: denotes an activation value.
- ***t***: denotes a target value.
- ***net:*** denotes the net input.

4.4.2 Gradient Descent on Error

We can motivate the backpropagation learning algorithm as gradient descent on sum-squared error (we square the error because we are interested in its magnitude, not its sign). The total error in a network is given by the following equation (11) (the 1/2 will simplify things later).

$$E = \frac{1}{2}\sum_k (t_k - a_k)^2 \qquad (11)$$

We want to adjust the network's weights to reduce this overall error [4].

$$\Delta W \propto -\frac{\partial E}{\partial W} \qquad (12)$$

We will begin at the output layer with a particular weight.

$$\Delta w_{kj} \propto -\frac{\partial E}{w_{kj}} \qquad (13)$$

However error is not directly a function of a weight. We expand this as follows [9].

$$\Delta w_{kj} = -\varepsilon \frac{\partial E}{\partial a_k}\frac{\partial a_k}{\partial net_k}\frac{\partial net_k}{\partial w_{kj}} \qquad (14)$$

Let's consider each of these partial derivatives in turn. Derivative of the error with respect to the activation is given by the equation $\frac{\partial E}{\partial a_k} = \frac{\partial(\frac{1}{2}(t_k - a_k)^2)}{\partial a_k} = -(t_k - a_k)$ (15)

a term in the formula for δ_k, the error signal :

$$\frac{\partial a_k}{\partial net_k} = \frac{\partial(1+e^{net_k})^{-1}}{\partial net_k} = \frac{e^{-net_k}}{(1+e^{-net_k})^2} \qquad (16)$$

We'd like to be able to rewrite this result in terms of the activation function. Notice that

$$1 - \frac{1}{1+e^{-net_k}} = \frac{e^{-net_k}}{1+e^{-net_k}} \qquad (17)$$

Using this fact, we can rewrite the result of the partial derivative as:

$$a_k(1 - a_k)$$

4.4. 3 Derivative of the net input with respect to a weight

Note that only one term of the net summation will have a non-zero derivative: again the one associated with the particular weight we are consider in.

$$\frac{\partial \mathrm{net_k}}{\partial \mathrm{w_{kj}}} = \frac{\partial(\mathrm{w_{kj}a_j})}{\partial \mathrm{w_{kj}}} = \mathrm{a_j} \qquad (18)$$

4.4.5 Weight change rule for a hidden to output weight

Now substituting these results back into our original equation we have:

$$\Delta W_{kj} = \varepsilon(t_k - t_a)a_k(1 - a_k)a_j \qquad (19)$$

$$\delta_k = (t_k - t_a)a_k(1 - a_k) \qquad (20)$$

Notice that this looks very similar to the Perceptron Training Rule. The only difference is the inclusion of the derivative of the activation function. This equation is typically simplified as shown below where the $\delta_k$term repesents the product of the error with the derivative of the activation function.

$$\Delta W_{kj} = \varepsilon\delta_k a_j \qquad (21)$$

4.4.5 Weight change rule for an input to hidden weight

Now we have to determine the appropriate weight change for an input to hidden weight. This is more complicated because it depends on the error at all of the nodes this weighted connection can lead [9].

$$\Delta w_{ij} \propto -[\sum_k \frac{\partial E}{\partial a_k}\frac{\partial a_k}{\partial net_k}\frac{\partial net_k}{\partial a_j}]\frac{\partial a_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}} = \varepsilon[\sum_k (t_k - t_a)a_k(1 - a_k)w_{kj}]\, a_j(1 - a_j)a_i \qquad (22)$$

$$= \varepsilon[\sum_k \delta_k w_{kj}]\, a_j(1 - a_j)a_i$$

$$\Delta w_{ji} = \varepsilon\delta_j a_j$$

5. Create a feed-forward backpropagation network with Matlap

Purpose create a feed-forward backpropagation network

Syntax net = newff

net = newff (PR,[S1 S2...SNl],{TF1 TF2...TFNl},BTF,BLF,PF)

Description net = newff creates a new network with a dialog box.

Newff (PR,[S1 S2...SNl],{TF1 TF2...TFNl},BTF,BLF,PF) takes,

PR — R x 2 matrix of min and max values for R input elements

Si — Size of ith layer, for Nl layers

TFi — Transfer function of ith layer, default = 'tansig'

BTF — Backpropagation network training function, default = 'trainlm'

BLF — Backpropagation weight/bias learning function, default = 'learngdm'

PF — Performance function, default = 'mse'

The transfer functions TFi can be any differentiable transfer function such as tansig, logsig, or purelin [7].

In either case, calling train with the resulting network will train the network with trainlm .

5.1 trainlm

Purpose Levenberg-Marquardt backpropagation

Syntax[net,TR]= trainlm(net,Pd,Tl,Ai,Q,TS,VV,TV)

info = trainlm(code)

Description trainlm is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization.

trainlm(net,Pd,Tl,Ai,Q,TS,VV,TV) takes these inputs,
net — Neural network.
Pd — Delayed input vectors.
Tl — Layer target vectors.
Ai — Initial input delay conditions.
Q — Batch size.
TS — Time steps.
VV — Either empty matrix [] or structure of validation vectors.
TV — Either empty matrix [] or structure of validation vectors.
and returns,
net — Trained network.
TR — Training record of various values over each epoch:
Training occurs according to the trainlm's training parameters shown here with their default values [7].

5.2 Algorithm

Using algorithms that were studied in the MATLAB program in the field of artificial neural networks after training and application we are obtained for shapes where as illustrated in Figure (10) how they were arrested on the body the recognized it and that the program has been training beforehand, either Figure (11) explains the arrest and identification of body image and the one who did not moments Hu unable to identify it. As for the forms that have not been previously trained by the program, it does not deal with the launching and gives the message that it is not recognized body and that body is not a target as shown in Figure (12).

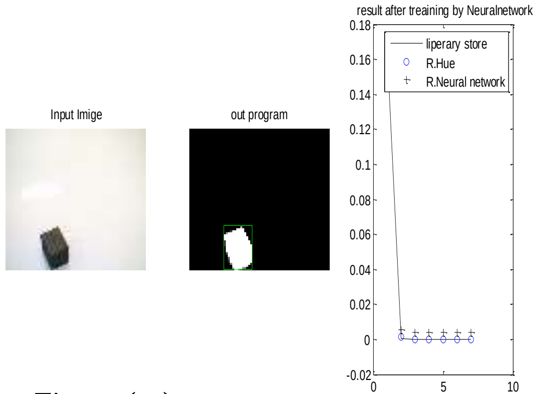


Figure (10)

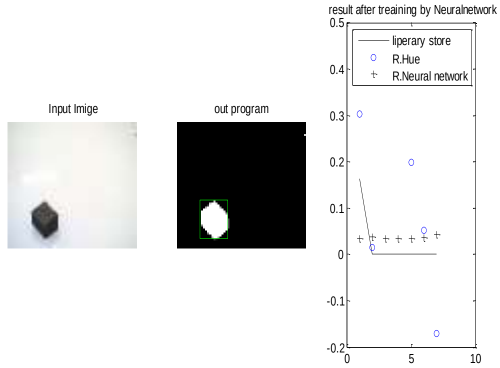


Figure (11)

```
the Object is not target
>>
```

Figure (12)

6. Conclusions

The aim of this work was to develop the software part of a Program imaging system that can automatically identify object in the field. This study resulted in developing a new recognition algorithm which uses image processing techniques based on **artificial neural network** ANN to identify shapes which we are training the program for it.

Tow recognition methods, Hue moments, ANN methods were used in this recognition system.

Note through a previous study that artificial neural networks were the best in terms of accuracy in the identification and discrimination, where we have achieved a high percentage of known objects and the proportion of very few ratio virtually non-existent compared to the method of moments Hu. As characteristic of these networks of high capacity for training and learning approach and previous values with subsequent....

7. References

[1] GHassan.Satouf, Mahmoud.Kaid," **The Control Of Robot In Real Time",** Engineering Science Series, Res. J. of Aleppo,26/2/2009.

[2] G.satoof, S. Jaklk," **Use techniques of digital images to determine the coordinates of the Centre for Spatial body weight for robot vision system",** Engineering Science Series, Res. J. of Aleppo,2008.

[3] Y. Wu, et al," **Simulation studies of data classification by artificial neural network",** Potential applications in medical imaging and decision making, Proc. SPIE Med. Imaging1090,1998,163-154.

[4] K. Prasad, D.C. Nigam, A. Lakhotiya, D. Umre " **Character Recognition Using Matlab's Neural Network Toolbox",** *B.I.T Durg, India,* Vol. 6, No. 1, February, 2013.

[5] S. M. ASHAGHATHRA, Dr. Paul Weckler,"**IDENTIFICATION OF PECAN WEEVILS THROUGH IMAGE PROCESSING**", Oklahoma State University, May 2008.

[6]ROBI POLIKAR, Rowan University, Glassboro, New Jersey**," PATTERN RECOGNITION"**, Wiley Encyclopedia of Biomedical Engineering, Copyright *&* 2006 John Wiley & Sons, Inc.

[7] *Howard Demuth, Mark Beale*," **Neural Network Toolbox**", *User's Guide*
COPYRIGHT, 1992 - 2002 by The MathWorks, Inc.

[8] M.H Beale, *M.T Hagan ,H.B Demuth,* " ***Neural Network Toolbox™ User's Guide***", © COPYRIGHT 1992–2015 by The MathWorks, Inc.

[9] David .S Touretzk ," **Backpropagation Learning**", 15-486/782: Artificial Neural Networks, Fall 2006.