J. Lopez, J. Lopez, J. A. Manceras, and A. Mana, "Neural Networks Applied to the Estimation of Object Orientation", *International Conference on Imaging Science, Systems, and Technology (CISST98)*, pp. 418-424, 1998.
NICS Lab. Publications: https://www.nics.uma.es/publications

NEURAL NETWORKS APPLIED TO THE ESTIMATION OF OBJECTS ORIENTATION

F. López, J. López
Dep. of Computer Science
University of Málaga. E.T.S.I. Informática.
Campus de Teatinos. 29071 MÁLAGA
e-mail: valverde@lcc.uma.es; jlm@lcc.uma.es

ABSTRACT

We present in this paper a first approach to the use of artificial neural as a tool to determine the orientation of objects moving on a conveyor belt in a car assembly line. The capability of neural networks to generalise is a key element in the calculation of an object's orientation. In this sense, a neural network with Competitive Hebbian Learning can identify the angle of a part never used in its training process. The equilibrium between exactitude and processing time is also studied.

1. INTRODUCTION

Most current industrial assembly lines use conveyor belts to provide the parts to be assembled. The parts on the belt can be in any position, and the determination of this position is extremely important for a robot arm to be able to grip and assemble parts correctly. The control system we have designed is a neural network[1] whose input is a video camera image of the part on the conveyor belt, and whose output is the angle of that part. In this first approach, images of the different parts have been simulated. In a further work, we'll apply our network to a real assembly system. This system consists of a Hebbian competitive learning neural network[2][3]. The aim of this study is to determinate, as accurately as possible, the orientation of an object on the belt, represented by a binary image. The complete system is showed in figure 1.

In section 2, image preprocessing and neural network design are studied; in section 3, how the neural network is trained and how we manipulate the network outputs, in section 4 we explain one of the most important problems found, symmetry, and finally, in section 5 we show the results obtained in the new system.

2. IMAGE PRE-PROCESSING AND NEURAL NETWORK DESIGN

This section shows the different processes that we apply to the image in order to obtain an input vector for the neural network. The image obtained by the video camera, that is, the network input, is a 320x200 two-tone (black and white) image, i.e., 64000 pixels. Each pixel is connected to an input unit, Therefore a network of 64000 input units is needed. This is quite out of proportion as regards space, but even more so as regards time because the system has to produce an answer in real time. So, in order to reduce the number of input units the number of pixels is decreased, obtaining a 30 x 30 pixel (quadrant) image[4]. A simple arithmetic mean is carried out for each quadrant until we obtain one of two possible values (black or white). An example of the image is shown in figure 2.

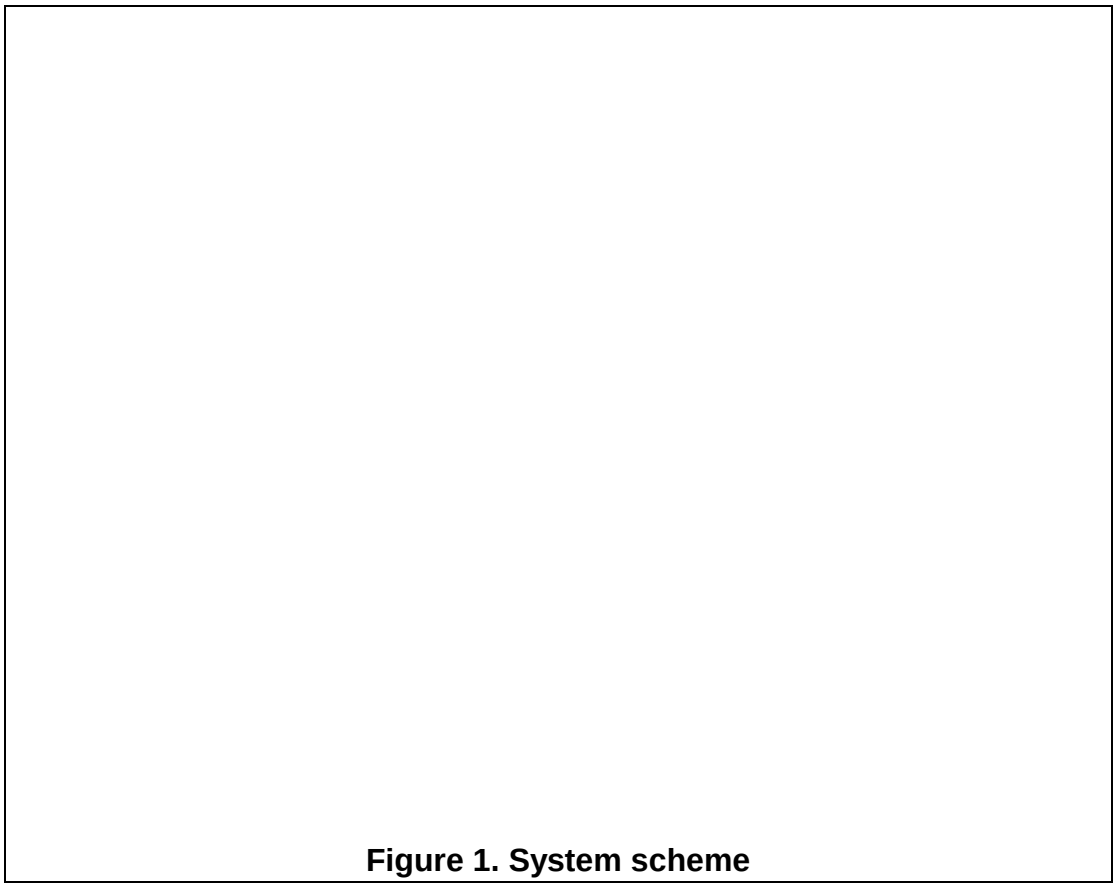

Figure 1. System scheme

Figure 2. Spark plug image after pre-processing

For each part and rotation algorithm we get nine different orientations that will be used to train and evaluate the neural network. Several rotation algorithms were considered. The better algorithms were long time computation, so it was chosen an algorithm with medium quality and fast: the rotation matrix of sines and cosines. The quality of the rotated image is not as good as those obtained using other methods, but it is much more fast, and that is one of our objectives. In every rotation, the original 320 x 200 two-tone image has been used and, once it has been rotated, the 30 x 30 pixel image used as input vector has been obtained.

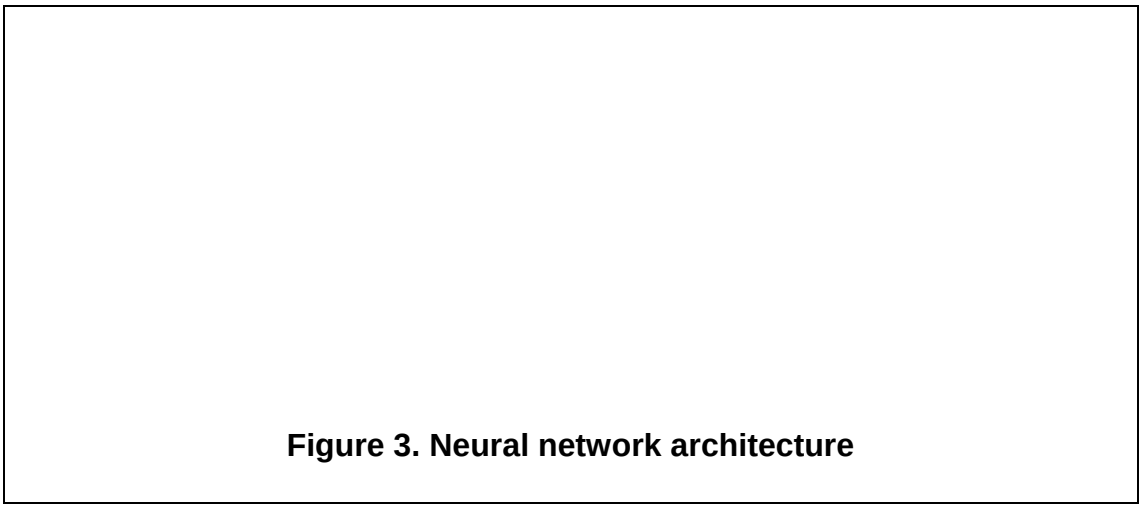

Figure 3. Neural network architecture

The control system that communicates the angle of the part to the robot[5] arm is an artificial neural network based on competitive Hebbian learning. The network architecture is quite simple: one layer with 30 x 30 inputs (one for each pixel of the processed image from the video camera); and 9 outputs (figure 3). Each output corresponds to a different angle. By dividing 360º in nine 40º sections we obtain the following angles (0º, 40º, 80º, ..., 280º, 320º). Originally, a division into eight 45º sections was studied, but some problems appeared in the neural network training process due to the intrinsic symmetry of this division. The use of one extra section eliminated this symmetry in most of cases and consequently the learning problems. This and other problems will be explained in section 4,

3. NEURAL NETWORK TRAINING

The aim of training a neural network is to assign a suitable value to its weights so that it fulfils the function it was trained for. The network uses a set of patterns for every training session, and once trained, its weights are not modified. The objective of this training is to determine the angle of any part located on the conveyor belt. An important factor in such task is that it must work correctly for any part whatsoever, even for those not included in the training patterns. This is essential to bear in mind in the choice of the set of training patterns because such a set must be the most representative for any future network input. So, in our study of a car assembly line, we needed to choose a part similar to any other part that might be used. A pendulum was chosen for this task (figure 4). Although this is not a real part from an assembly line, it is certainly similar to many of those used in car assembly. Successive 40º rotations were applied to the pendulum image to obtain nine patterns of the same part in different orientations (0º, 40º, 80º, 120º, 160º, 200º, 240º, 280º, 320º).

In this way, each input pattern corresponds to a single orientation and a single network output unit. During the learning process the neural network modifies the weight values as soon as different input patterns appear. This training system uses feedback, that is, weight modification is a function not only of their current values but also of the output values resulting from the different input patterns. Each network output has an associated orientation, and those outputs takes continuous values. The network operates as follows: given a part with an unknown orientation, as input, the output with a higher value (winner output) suggests its orientation. If input orientation matches to one of the orientations associated to outputs, then the value of that output (winner) is much higher than the rest. If input orientation doesn't match, let's say 60º (between 40º and 80º), the outputs associated to these two orientations takes similar values. Generally, to calculate the orientation of the input, the system uses a lineal factor of three outputs, the winner and its two adjacent.

Figure 4. Pendulum as representative part

A problem appeared in the beginning, when the network had eight input units and used 45º sections. In this situation there was a symmetry that caused some cases of confusion with the output associated to its reversed orientation (Xº +180º). As soon as the division into nine sections was used, both symmetry and confusion problems disappeared.. It must be noted that the orientations of Xº and its opposite, Xº + 180º, are different; the robot arm gripped the part well, but assembled it incorrectly. Chips are typical examples of this problem, which we look at in detail later.. One chip is totally symmetric and can be plugged into the board incorrectly because different pins have different functions. In these cases a simple mark on the part (using a sticker, for example) can solve the problem.

4. ESTIMATION ALGORITHM

A complete study has been done concerning the relationship of the output values of the winner neuron and the highest of the two adjacent ones. This study is used for the calculation of the part orientation on the conveyor belt. Different ways were considered for the comparison of these two values and it could be tested that the best of them was to calculate the quotient of these two magnitudes. In order to determine the strength of the answer for every winner output, the output values of the different winner neurons were compared introducing the nine training patterns. However, these values were found not to be equal; not even similar. When the two highest values of the neurons were compared, no logic relationship was found; that is, we realized it was not an useful information for our problem

As we have already suggested, the best, or even the unique correct way was to obtain the quotient of the two adjacent highest values of the output vector.

Taking any of the different 40º sector that contains one of the training patterns, a group of values for our quotient was carried out. When we affirm that any sector can be

used, we are implicitly deducing, and so has been tested, that a periodicity exists in the observation of the results obtained with these nine patterns.

The following task was to find an interpolation function valid for every possible orientation of a part to be assembled. If such a function could be found, the quotient would be introduced in order to obtain the resulting angle. Obviously, it was also necessary to keep a table to relate the output neurons and the angles that made them take the highest values. With this two data it was possible to determine a satisfactory result. Firstly of all, a lineal interpolation for each period was used. The error rates obtained were not excessive for those parts that were similar to the pendulum. For those parts whose similarity to the pendulum was lower, the error percentage turned out to be too high to be admissible. Finally, a parabolic interpolation was used.

Another point had to be considered for those parts with a certain symmetry. In order to act correctly, certain rules of election were established.For a few orientations of the patterns, the two neurons adjacent to the winner one could have very low outputs, and the two opposite neurons had relatively high outputs. It was necessary to make some purely empirical adjustments in order to determine the correct angle. These adjustments were based on the quotients of the mentioned five neurons: the winner, the two adjacent ones and the two opposite ones.

As a possible further work, other two neural networks could be designed , in order to complete and implement the whole system. One of them could locate and centre the image on the conveyor belt. The other one could determine the type of part on the belt so that the robot arm could assemble it correctly.

5. RESULTS

Once training was successfully completed, the system was ready for testing. A set of different parts used in car assembly was chosen for this.. Each of these parts was rotated from 0º to 360º and supplied to the neural network. An orientation value was obtained for every rotation and then compared to the real orientation of the part in order to calculate the error. The results obtained are shown in figures 5a, 5b y 5c. The worst orientation error did not exceed 25º. It must be taken into account that the network had not seen any of these parts prior to this test.

Figure 5a. Results.

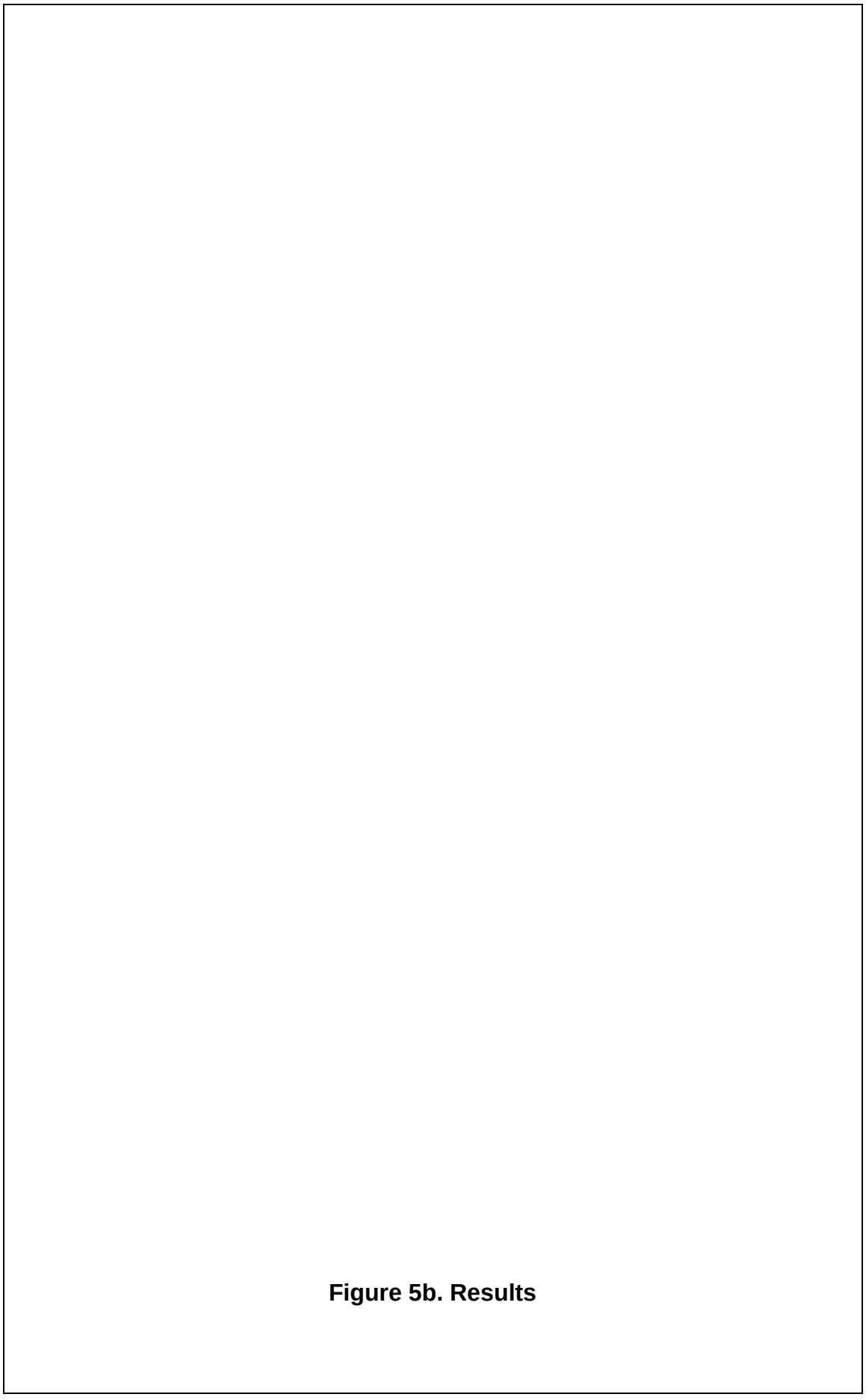

Figure 5b. Results

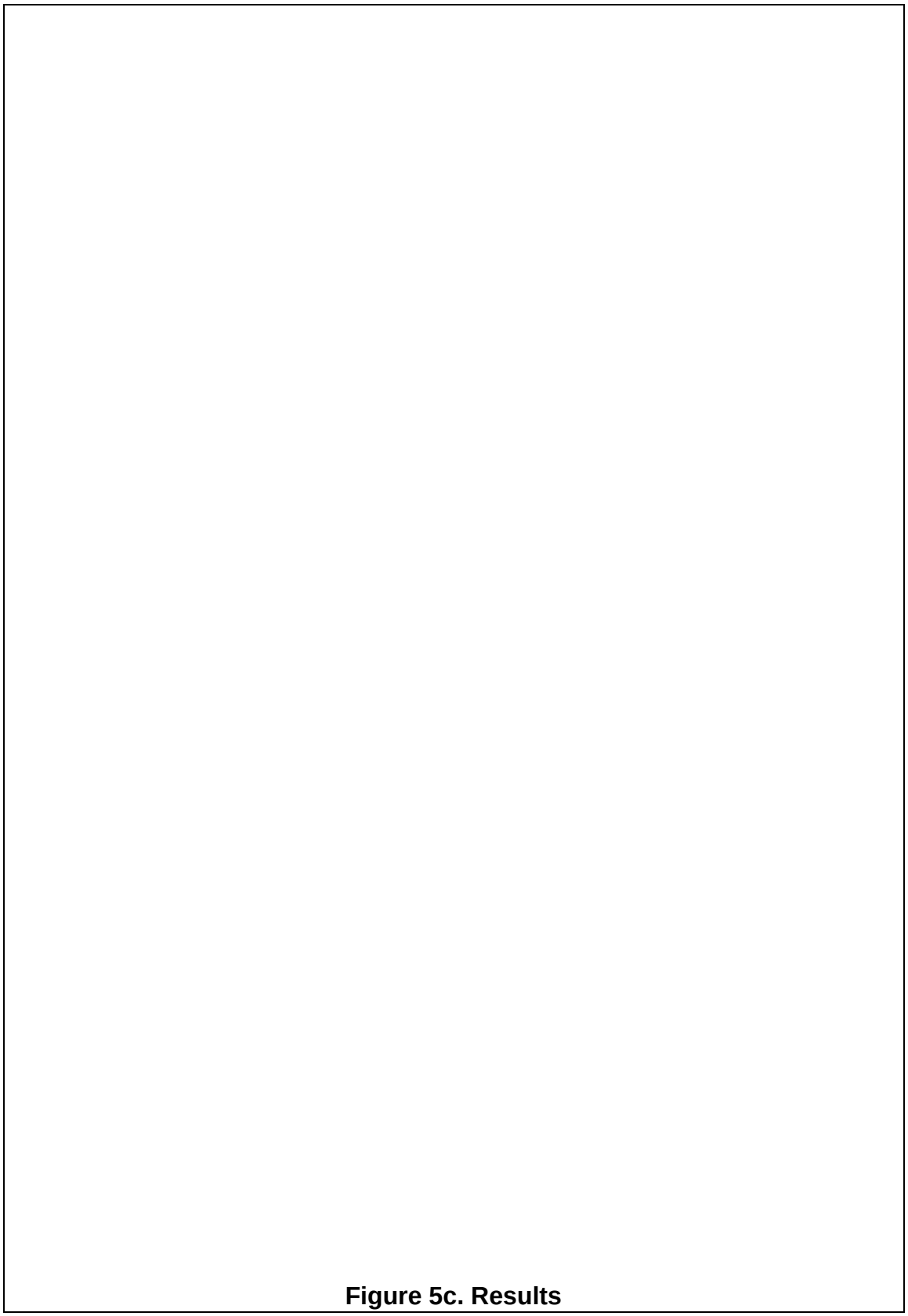

Figure 5c. Results

6. CONCLUSIONS AND FUTURE WORKS.

Although the margin of error may seem high (10^{o}), this value is acceptable in order for the part to be correctly assembled. This method has various advantages compared to other methods:

· it can calculate the angle of new parts not previously seen;
· the running time is very low (virtually negligible).

The only disadvantage concerns accuracy. With other methods the margin of error is about 1º.

To improve the system as a future work it could be added the two neural networks which were shown in section 4:

· one to locate and centre the image on the conveyor belt and
· another to determine the type of part on the belt.

REFERENCES.

[1] **Neurocomputing.** Hetch-Nielsen R.
Addison-Wesley, 1990.

[2] **Competitive Hebbian Learning : Algorithm and Demonstration.** Ray H. White
Neural Networks. Vol. 5, 1992.

[3] **Image Compression based on Competitive Hebbian Learning Neural Networks.**
F. López, J. López, C. Maravall. Brain Process, Theories and Models.
An International Conference in Honor of W.S. McCulloch, 25 years after his death.
MIT Press.

[4] **Image Pattern Recognition.** Kovalesky.
Springer-Verlag, 1980.

[5] **Robotics.** John J. Craig.
Addison-Wesley, 1986.