Image processing for positioning mechanical device with Backpropagation algorithm and separate handling of RGB components

Processamento de imagem para posicionamento de dispositivos mecânico com algoritmo de retropropagação e manipulação em separado de componentes RGB

Procesamiento de imágenes para posicionamiento de dispositivo mecánico con algoritmo de retropropagación y manejo separado de componentes RGB

Received: 01/14/2022 | Reviewed: 01/19/2022 | Accept: 01/21/2022 | Published: 01/23/2022

Mauricio Conceição Mario
ORCID: https://orcid.org/0000-0002-1973-2186
Santa Cecilia University, Brazil
E-mail: cmario@unisanta.br
Alzira Marques de Oliveira
ORCID: https://orcid.org/0000-0002-5344-1091
Santa Cecilia University, Brazil
E-mail: amo_yoga@hotmail.com
João Inácio da Silva Filho
ORCID: https://orcid.org/0000-0001-9715-8928
Santa Cecilia University, Brazil
E-mail: inacio@unisanta.br
Dorotéa Vilanova Garcia
ORCID: https://orcid.org/0000-0002-8769-0328
Santa Cecilia University, Brazil
E-mail: dora@unisanta.br
Heraldo Silveira Barbuy
ORCID: https://orcid.org/0000-0001-7249-2890
Santa Cecilia University, Brazil
E-mail: barbuy@unisanta.br

Abstract
Different approaches for the use of Artificial Neural Networks - ANNs, in the recognition of image patterns, have been used with variations ranging from the processing of the image data to the ANN architecture itself. This paper describes the development of a system that aims to recognize patterns of images with ANNs of three inputs that receive images decomposed into their RGB components. The ANNs have an architecture with two hidden layers of six neurons each, and use the algorithm Backpropagation. The built model normalizes RGB components with values between zero and one. The Backpropagation algorithm is used for the purpose of functional approximation of these components, and after training, the numerical arrangements obtained in the three outputs corresponding to the inputs are denormalized to form the resulting training image. Six image pattern had training in different ANNs, forming a system to recognized each pattern. The feasibility of using the model was verified with the tests for its generalization capacity. Images used to position a mechanical device, which did not participate in the training, were inserted into the system and from them the positioning of the device was performed, with a high degree of accuracy.
Keywords: Artificial neural networks; Automation; Digital images; Backpropagation Algorithm.

Resumo
Diferentes abordagens para o uso de Redes Neurais Artificiais - RNAs, no reconhecimento de padrões de imagem, têm sido utilizadas com variações que vão desde o processamento dos dados da imagem até a própria arquitetura da RNA. Este artigo descreve o desenvolvimento de um sistema que visa reconhecer padrões de imagens com RNAs de três entradas que recebem imagens decompostas em seus componentes RGB. As RNAs possuem uma arquitetura com duas camadas ocultas de seis neurônios cada, e utilizam o algoritmo Backpropagation. O modelo construído normaliza os componentes RGB com valores entre zero e um. O algoritmo Backpropagation é utilizado para fins de aproximação funcional desses componentes e, após o treinamento, os arranjos numéricos obtidos nas três saídas correspondentes às entradas são desnormalizados para formar a imagem de treinamento resultante. Seis padrões de imagem foram treinados em diferentes RNAs, formando um sistema para reconhecer cada padrão. A viabilidade de utilização do modelo foi verificada com os testes de sua capacidade de generalização. Imagens utilizadas para

posicionar um dispositivo mecânico, que não participou do treinamento, foram inseridas no sistema e a partir delas foi realizado o posicionamento do dispositivo, com alto grau de precisão.
Palavras-chave: Redes neurais artificiais; Automação; Imagens digitais; Algoritmo Backpropagation.

Resumen
Se han utilizado diferentes enfoques para el uso de Redes Neuronales Artificiales - ANN, en el reconocimiento de patrones de imagen, con variaciones que van desde el procesamiento de los datos de la imagen hasta la propia arquitectura ANN. Este artículo describe el desarrollo de un sistema que tiene como objetivo reconocer patrones de imágenes con ANNs de tres entradas que reciben imágenes descompuestas en sus componentes RGB. Las ANN tienen una arquitectura con dos capas ocultas de seis neuronas cada una, y utilizan el algoritmo Backpropagation. El modelo construido normaliza los componentes RGB con valores entre cero y uno. El algoritmo Backpropagation se utiliza con el fin de realizar una aproximación funcional de estos componentes y, después del entrenamiento, los arreglos numéricos obtenidos en las tres salidas correspondientes a las entradas se desnormalizan para formar la imagen de entrenamiento resultante. Seis patrones de imagen fueron entrenados en diferentes ANN, formando un sistema para reconocer cada patrón. La factibilidad de uso del modelo se verificó con las pruebas de su capacidad de generalización. Las imágenes utilizadas para posicionar un dispositivo mecánico, que no participaba en el entrenamiento, fueron insertadas en el sistema ya partir de ellas se realizó el posicionamiento del dispositivo, con un alto grado de precisión.
Palabras clave: Redes neuronales artificiales; Automatización; Imágenes digitales; Algoritmo de Retropropagación.

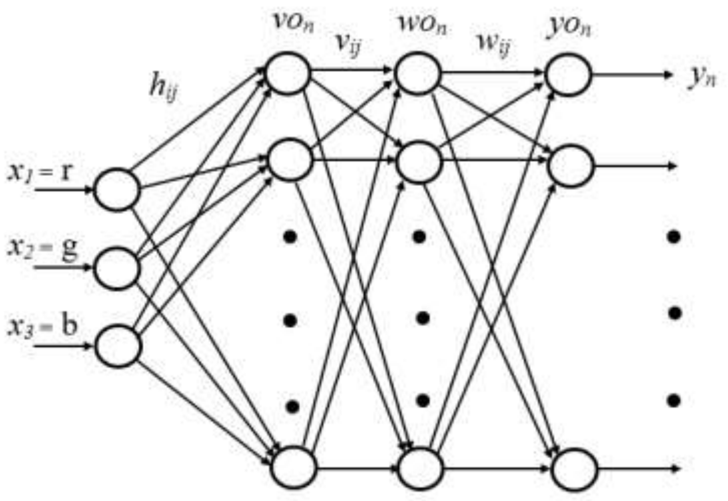
1. Introduction

The use of Artificial Neural Networks for the classification and recognition of image patterns in applications involving process automation has been increasingly widespread and in different areas. Regarding the architectures and algorithms of ANNs applied in digital image processing, recent works describe the use of feed forward networks to extract features such as image textures (Tuncer et al., 2019), and the use of Deep Convolutional Neural Networks to classify objects in 3D images and real-time identification of objects in video images or advanced automation applications (Lakhili et al., 2018; Newby et al., 2018; Sharma et al., 2018; Wei et al., 2021). Computer vision problems are characterized by the large amount of manipulated information, which require the treatment of high definition images with a large amount of additional information, such as coloring and positioning in three-dimensional space (Braga et al., 2012). This work describes a simpler alternative method for image processing, where the main approach is to decompose it into its RGB components (Gonzalez & Woods, 2010; Pedrini & Schartz, 2008; Solomon & Breckon, 2013), which enhances the characterization of a pattern as a function of color. The prototype developed in this work uses a passive sensor, a camera (Russel & Norvig, 2004), to observe and collect images of the environment. The positioning of the mechanical device from image processing through Neural Networks can serve, for example, in an Industrial Automation system, as an effector of a robot (Capelli, 2008). In the state of the art of computer vision, a technique used in this work, we can mention the mathematical model used to measure a 3D surface (Talon & Pellegrino, 2022) using angle measurements from embedded sensors; another type of computer vision problem, the synthesis of images from cross-views (Regmi & Borji, 2019) through the use of mapping points from one plane to another, homography; Deep Convolutional Neural Networks have also been relevant to the problem of image restoration with a high degree of accuracy (Sharma et al., 2018; Wang et al., 2020).

1.1 Artificial Neural Network for image processing

The image data processing of this work was initially elaborated from the Artificial Neural Network represented in Figure 1, which presents architecture with three inputs identified by x_i, two hidden layers with six neurons each, identified by vo_n and wo_n, and six outputs identified by y_n. Index n identifies the position of the neuron in its layer. In the context of machine learning (Baranauskas & Monard, 2000) the Artificial Neural Network uses the error correction learning rule, the Backpropagation algorithm (Faceli et al., 2019; Haykin, 2001; Luger, 2013), and the hidden layer neurons, as well as the outputs, use the bipolar sigmoid nonlinear activation function. In the image pattern recognition application described in (Mario et al., 2018), in image preprocessing it is segmented into its RGB components (Gonzalez & Woods, 2010; Pedrini & Schartz, 2008; Soloman, 2012; Solomon & Breckon, 2013), and these components are inserted into the network inputs, with the Red component inserted at input x_1, the Green component inserted at input x_2 and the Blue component inserted at input x_3. The network is fully connected, as a consequence, one neuron in any layer is connected to all neurons in the previous layer (Haykin, 2001). In the application described in (Mario et al., 2018) the outputs y_1, y_2 and y_3 converge respectively, after training, to inputs *r*, *g* and *b*. Thus, it is possible to reconstruct a pattern of the images submitted to training by recomposing the RGB components presented in the network outputs. Outputs y_4, y_5 and y_6 are respectively replicas of outputs $y1$, y_2 and y_3. The hx_{ij}, vij, and wij synapses are linked respectively to input neurons, vo_n neurons, wo_n neurons, and outputs. Indices *ij* identify which neuron the synapse is linked to and the order of its position.

Figure 1: Artificial Neural Network for image data processing.



Source: Authors (2022).

Network training is described below in *Algorithm 1*:

***Algorithm 1* – start:**

```
1.  Definition of synapse weight initial values $h_{ij}$, $v_{ij}$ and $w_{ij}$;
    $h_{ij} \leftarrow h_{ij}[0]$;                                   (1)
    $v_{ij} \leftarrow v_{ij}[0]$;                                   (2)
    $w_{ij} \leftarrow w_{ij}[0]$;                                   (3)

2.  Setting the initial values of the neuron bias value $vo_n$, $wo_n$ and $y_n$;
    $v_n \leftarrow v_n[0]$;                                         (4)
    $w_n \leftarrow w_n[0]$;                                         (5)
    $yb_n \leftarrow yb_n[0]$;                                       (6)

3.  Setting initial value of network learning rate – $\alpha$;
    $\alpha \leftarrow \alpha[0]$;                                   (7)
4.  Calculation of the additive junction of $vo_n$ neurons;
    $vo_n = \sum_{n=1}^{n} (x_i * h_{ij} + v_n)$                     (8)

5.  Calculation of neuron activation function $vo_n$;
    $\varphi_n(vo_n) = \frac{2}{1+ e^{-vo_n}} - 1$                   (9)
6.  Calculation of the activation function derivative $vo_n$;
    $\varphi'_n(vo_n) = \frac{1}{2} * (1 + vo_n) * (1 - vo_n)$       (10)
```

7. Calculation of additive junction of *wo_n* neurons;

$$wo_n = \sum_{n=1}^{n} \left(\varphi_n(vo_n) * v_{ij} + w_n\right) \quad (11)$$

8. Calculation of activation function of wo_n neurons;

$$\varphi_n(wo_n) = \frac{2}{1+e^{-wo_n}} - 1 \quad (12)$$

9. Calculation of derivative of activation function wo_n;

$$\varphi_n'(wo_n) = \frac{1}{2} * (1 + wo_n) * (1 - wo_n) \quad (13)$$

10. Updating ouputs y_n;

$$y_n = \sum_{n=1}^{n} \left(\varphi_n(wo_n) * w_{ij} + yb_n\right) \quad (14)$$

11. Calculation of activation function of outputs y_n;

$$\varphi_n(y_n) = \frac{2}{1+e^{-y_n}} - 1 \quad (15)$$

12. Calculation of derivative of output activation function y_n;

$$\varphi_n'(y_n) = \frac{1}{2} * (1 + y_n) * (1 - y_n) \quad (16)$$

13. Error calculation of outputs $y_{1,4}$;

$$e_{1,4} = x_1 - y_{1,4} \quad (17)$$

14. Error calculation of outputs $y_{2,5}$;

$$e_{2,5} = x_2 - y_{2,5} \quad (18)$$

15. Error calculation of outputs $y_{3,6}$;

$$e_{3,6} = x_3 - y_{3,6} \quad (19)$$

16. Calculation of error propagation $e_{1,4}$;

$$\delta_{1,4} = \alpha * e_{1,4} * \varphi_{1,4}'\left(y_{1,4}\right) \quad (20)$$

17. Calculation of error propagation $e_{2,5}$;

$$\delta_{2,5} = \alpha * e_{2,5} * \varphi_{2,5}'\left(y_{2,5}\right) \quad (21)$$

18. Calculation of error propagation $e_{3,6}$;

$$\delta_{3,6} = \alpha * e_{3,6} * \varphi_{3,6}'\left(y_{3,6}\right) \quad (22)$$

19. Calculation of the bias update rate of vo_n, wo_n and y_n neurons;

$$\Delta v_n = \alpha * e_n * \varphi_n(wo_n) \quad (23)$$
$$\Delta w_n = \alpha * e_n * \varphi_n(y_n) \quad (24)$$
$$\Delta yb_n = \alpha * e_n \quad (25)$$

20. Update of the bias values of the vo_n, wo_n, and y_n neurons;

$$v_{n(t+1)} = \Delta v_n + v_n \quad (26)$$
$$w_{n(t+1)} = \Delta w_n + w_n \quad (27)$$
$$yb_{n(t+1)} = \Delta yb_n + yb_n \quad (28)$$

21. Calculation of w_{ij} synapse weights update rate;

$$\Delta w_{ij} = \delta_{ij} * \varphi_n(wo_n) \quad (29)$$

22. w_{ij} synapse weight update;

$$w_{ij(t+1)} = \Delta w_{ij} + w_{ij} \quad (30)$$

23. Calculation of v_{ij} synapse weights refresh rate;

$$\Delta v_{ij} = \alpha * \varphi_n'(vo_n) * \varphi_n'(wo_n) * \varphi_n(vo_n) * \sum_{n=1}^{n} \left(e_n * w_{ij}\right) \quad (31)$$

24. Update of v_{ij} synapse weights;

$$v_{ij(t+1)} = \Delta v_{ij} + v_{ij} \quad (32)$$

25. Calculation of hxi_{ij} synapse weights refresh rate;

$$\Delta hxi_{ij} = \sum_{n=1}^{n} \left(\Delta v_{ij} * v_{ij} * \varphi_n'(vo_n) * x_i\right) \quad (33)$$

26. Update of hxi_{ij} synapse weights;

$$\Delta hxi_{ij(t+1)} = \Delta hxi_{ij} + hxi_{ij} \quad (34)$$

Algorithm 1 – end:

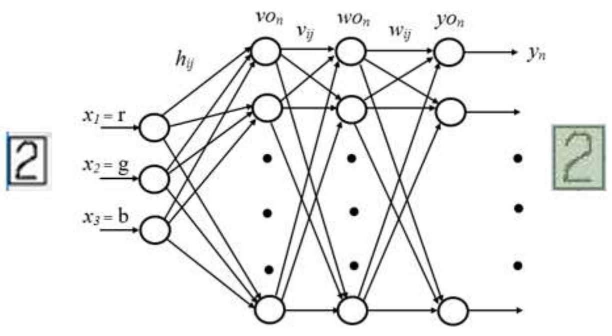
In the experiments described in (Mario et al., 2018) it was observed that the use of six-output architecture improves the accuracy of image reproduction compared to three-output architecture. Updating network synapse weights so that output data converge to input data occurs most effectively with the strategy of using duplicate outputs, propagating error minimization to synapses that are directly or indirectly linked to the outputs.

After image acquisition, the image is preprocessed to segment it into its RGB components. Image data is inserted into the application in the form of a numerical matrix in which the elements of the same are sequentially distributed as in the notation of equation (35) of Algorithm 2, where n is the dimension of the matrix, called *image_array*. The elements R, G and B of the *image_array* are respectively the Red, Green and Blue components of the image. Algorithm 2 described below demonstrates the creation of RGB component arrays from *image_array*:

```
Algorithm 2 – start:
    $image\_array = \{R_{[0]}, G_{[1]}, B_{[2]}, R_{[3]}, G_{[4]}, B_{[5]} \ldots R_{[n-2]}, G_{[n-1]}, B_{[n]}\}$    (35)
1. Initialization of RGB matrix element indexes;
    $p \leftarrow 0;$    (36)
    $q \leftarrow 0;$    (37)
    $r \leftarrow 0;$    (38)
2. Red component matrix formation;
    for (i = 0 to i = n)
        { $red[p] = matriz\_imagem[i]$    (39)
        $p = p + 1$    (40)
        $i = i + 3$ }    (41)
3. Green component matrix formation;
    for (j = 1 to j = n)
        { $green[q] = matriz\_imagem[j]$    (42)
        $q = q + 1$    (43)
        $j = j + 3$ }    (44)
4. Blue component matrix formation;
    for (k = 2 to k = n)
        { $blue[r] = matriz\_imagem[k]$    (45)
        $r = r + 1$    (46)
        $k = k + 3$ }    (47)
Algorithm 2 – end:
```

In experiments with this Neural Network (Mario et al., 2018) it was trained with a representative image of a numerical digit of dimensions 20 by 25 pixels and 54 training cycles were performed. In sequence the trained image was inserted into the network with the values of the synaptic weights obtained at the end of the training. After recomposing the data obtained at outputs y_1, y_2 and y_3, the resulting image can be recognized as an approximation of the trained image (Mario et al., 2018). The representation of the result of this phase of the experiment is shown in Figure 2.

Figure 2: representation of network response to trained image.

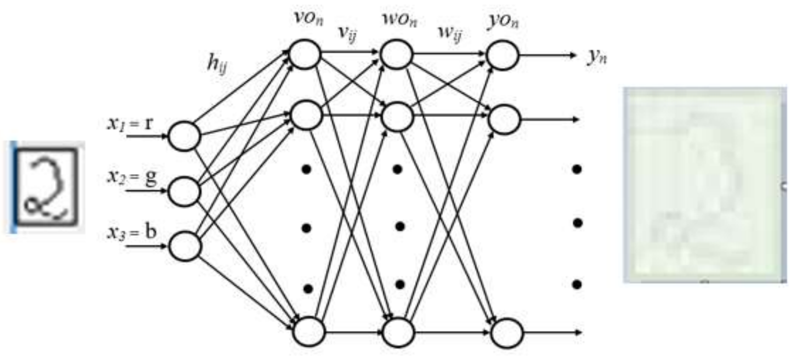


Source: Authors (2022).

The generalizability of the network was tested by inserting a similar image to the trained one, and checking the network response for the similarity of output and input network images. The representation of the result obtained in this test is shown in Figure 3.

The development of the system for positioning a mechanical arm of this work is based on the results of the experiments performed with the described Artificial Neural Network (Mario et al., 2018), with the difference that the system network outputs should converge to distinct numerical value ranges for each of the six desired positions for mechanical arm travel. The change in network functionality will be made through Algorithm 1 equations (17), (18) and (19), where the values of x_1, x_2 and x_3 will be replaced by numerical ranges equivalent to the positions of the mechanical arm.

Figure 3: representation of network response to trained image.



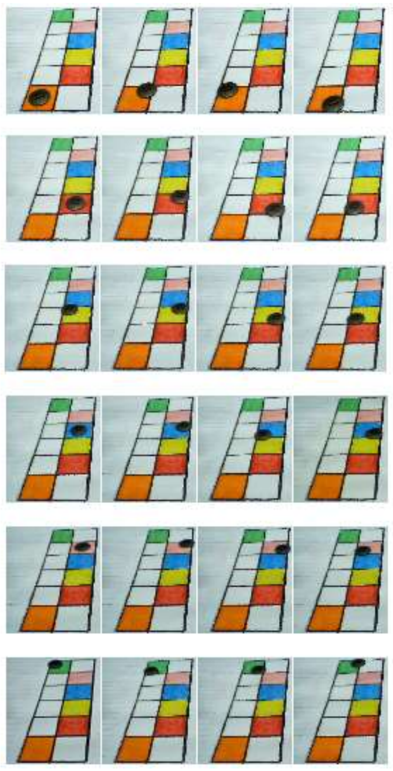
Source: Authors (2022).

2. Methodology

The development of this work is related to an experimental and quantitative research, based in results obtained in experiments carried out with two-layer Neural Networks for digital image recognition (Mario et al., 2018) and on the development of an electronic and mechanical prototype that is moved from a Backpropagation algorithm applied in Artificial Neural Networks.

Six image's patterns was used, different about as part position. The images relative to the positions of the piece were obtained through a digital camera, and were normalized to dimensions of 1.5 cm x 1.8 cm. Figure 4 shows the sets of images used for ANN training, respectively with the workpiece located in regions 1 through 6.

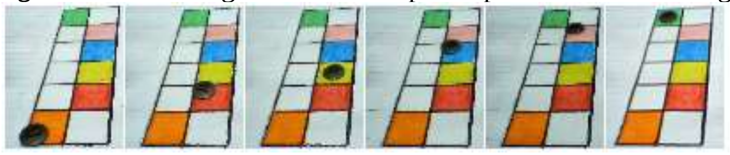
Figure 4: ANNs training image sets: positioning regions 1 to 6.



Source: Authors (2022).

The images used to test the generalizability of RNAs are shown in Figure 5. These images are different of test's images because the piece are positioned in different coordinates in each color's region.

Figure 5: set of images that did not participate in ANN training.

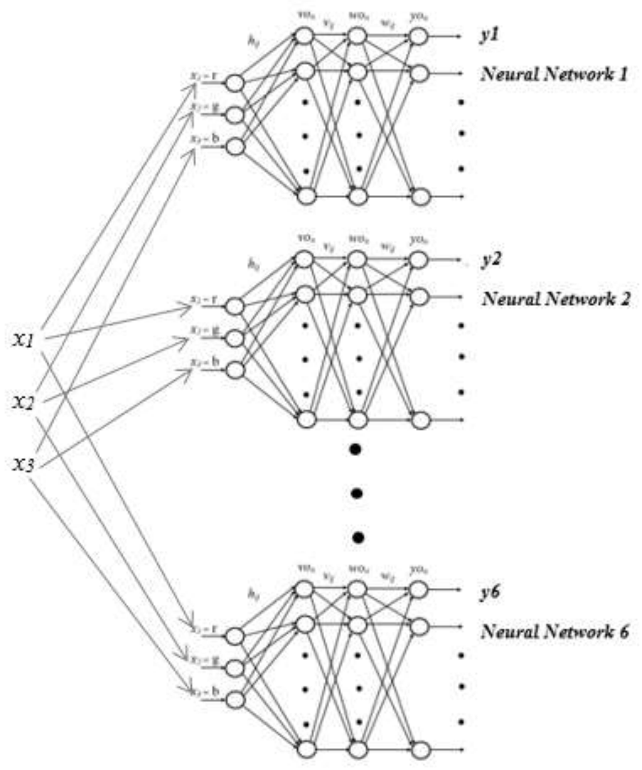


Source: Authors (2022).

2.1 Neural Networks for image pattern recognition

To recognize the patterns of the image sets presented in Figure 4, ANNs with the architecture described in the introduction are used. The programming language used for the construction of Artificial Neural Networks was Java, in the programming environment - IDE, NetBeans. The architecture of ANNs with common inputs is shown in Figure 6.

Figure 6: ANN structure for image pattern recognition.



Source: Authors (2022).

Neural Network 1 is trained to recognize the pattern of images in region 1, orange; Neural Network 2 is trained to recognize the pattern of images of region 2, red, and so on until Neural Network 6, trained to recognize the pattern of images of region 6, green.

In Neural Network training, algorithm 1 is changed so that the outputs y_1 through y_6 of each Neural Network in the structure converge to a single numerical value. The three error equations (17), (18) and (19) are then replaced by a single (48); notation of errors and outputs may be replaced respectively by $e_{1\text{-}6}$ and $y_{1\text{-}6}$. The variable s receives the numerical values for convergence of the outputs.

Output error calculation $y_{1\text{-}6}$;

$$e_{1-6} = s - y_{1-6} \tag{48}$$

2.2 Training of Artificial Neural Networks

The training was performed with different values of the variable s for each of the Neural Networks that will identify the positions of the piece through the regions from 1 to 6. The values are broken down in Table 1:

Table 1: performed with different values of the variable *s* for each ANN.

regions	*s values*
region 1	0,165
region 2	0,332
region 3	0,499
region 4	0,666
region 5	0,833
region 6	1,000

Source: Authors (2022).

2.3 Definition of the variation of the numerical value ranges of the Neural Networks outputs

Training defines the weight values of all synapses and biases of the Neural Networks. With these values set, the following Algorithm 1 steps are performed, which allow the recalculation of the output values considering the training weights:

***Code's piece from Algorithm 1*– start:**

4. Calculation of the additive junction of vo_n neurons;

$$vo_n = \sum_{n=1}^{n} (x_i * h_{ij} + v_n) \tag{8}$$

5. Calculation of neuron activation function vo_n;

$$\varphi_n(vo_n) = \frac{2}{1+ e^{-vo_n}} - 1 \tag{9}$$

7. Calculation of additive junction of wo_n neurons;

$$wo_n = \sum_{n=1}^{n} (\varphi_n(vo_n) * v_{ij} + w_n) \tag{11}$$

1. Calculation of activation function of wo_n neurons;

$$\varphi_n(wo_n) = \frac{2}{1+ e^{-wo_n}} - 1 \tag{12}$$

10. Updating ouputs y_n;

$$y_n = \sum_{n=1}^{n} (\varphi_n(wo_n) * w_{ij} + yb_n) \tag{14}$$

11. Calculation of activation function of outputs y_n;

$$\varphi_n(y_n) = \frac{2}{1+ e^{-y_n}} - 1 \tag{15}$$

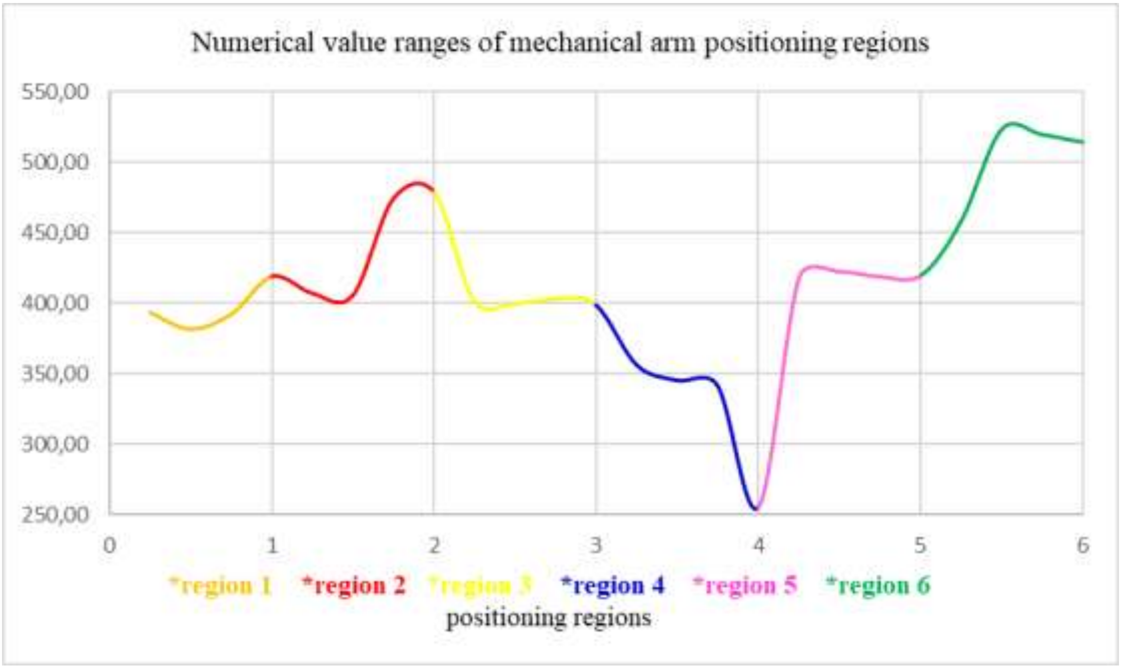
***Code's piece from Algorithm 1* – end:**

The numerical value ranges corresponding to the outputs of the Neural Networks are calculated by summing the values of each activation function component calculated in (15). Equation (49) describes the calculation of the ranges of numerical values equivalent to the outputs of each Neural Network.

$$\phi_n(y_n) = \sum_{n=1}^{n} \left(\varphi_n(y_n)\right) \tag{49}$$

Graph 1 shows the numerical value ranges of the outputs of the Artificial Neural Networks after training. The numeric ranges of the positioning regions represented in the graph are equivalent to the outputs of Neural Networks 1 to 6.

Graph 1: Equivalent numerical values of mechanical arm positioning regions.



Source: Authors (2022).

After defining the numerical value ranges of the Neural Networks outputs, the images used for training were inserted into the respective Neural Networks from 1 to 6, and then adjustments were made to the numerical value ranges and conditional order to mitigate the effect of overlapping values for same region, as can be seen in Graph 1.

2.4 Decision making system for defining the positioning of the mechanical device

The decision system uses to define the positioning of the mechanical arm in each region the numerical value ranges calculated for each Neural Network from 1 to 6, as described in Algorithm 3. In this algorithm, the variable *p* receives the numerical value corresponding to the displacement of the mechanical device and which is passed via the serial port to the stepping motors drive application. Its value is equivalent to the time and displacement of the horizontal stepping motor.

***Algorithm 3* – start:**

```
1.    $if\ (\ (\phi_1(y_1)\ \geq 390\ )\ and\ (\ \phi_1(y_1)\ < 390\ )\ )$
      $do\ \ p = 1$                                   (50)

2.    $else\ if\ (\ (\phi_2(y_2)\ \geq 370\ )\ and\ (\ \phi_2(y_2)\ < 480\ )\ )$
      $do\ \ p = 4$                                   (51)

3.    $else\ if\ (\ (\phi_3(y_3)\ \geq 380\ )\ and\ (\ \phi_3(y_3)\ < 470\ )\ )$
      $do\ \ p = 6$                                   (52)

4.    $else\ if\ (\ (\phi_5(y_5)\ \geq 360\ )\ and\ (\ \phi_5(y_5)\ < 430\ )\ )\ and\ (\ \phi_6(y_6)\ < 513\ )$
      $do\ \ p = 11$                                  (53)

5.    $else\ if\ (\ (\phi_4(y_4)\ \geq 250\ )\ and\ (\ \phi_4(y_4)\ < 356\ )\ )\ and$
      $((\ \phi_6(y_6)\ < 584\ )\ and\ (\ \phi_6(y_6)\ < 598\ )\ )$
      $do\ \ p = 9$                                   (54)

6.    $else\ if\ (\ (\phi_6(y_6)\ \geq 450\ )\ and\ (\ \phi_6(y_6)\ < 820\ )\ )$
```

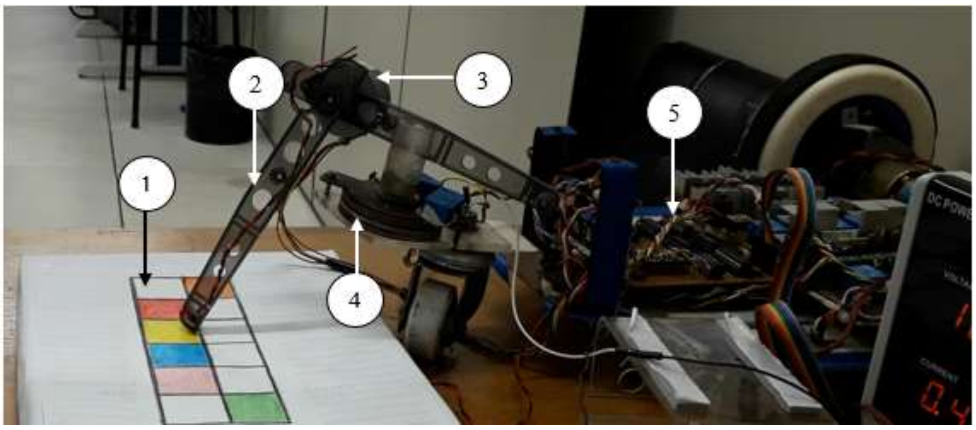
```
        do   $p = 13$                                   (55)
7.          else
        do   $p = 0$                                    (56)
```

***Algorithm 3* – end:**

2.5 Mechanical device drive system

The mechanical device consists of two stepper motors, one for horizontal displacement and one for vertical displacement. A plastic rod is fixed to the vertically moving stepper motor shaft, and through the rod the arm approaches the position of the part. A plastic rod is fixed to the vertically moving stepper motor shaft, and through the rod the arm approaches the position of the part. The horizontal plane where the mechanical arm moves has dimensions of 18 cm by 6 cm and in this plane six regions of 9 cm2 were defined for workpiece displacement, four of these regions positioned side by side and two vertically displaced in relation to the others. Each region of the piece has a specific color, being region 1 orange, region 2 red, region 3 yellow, region 4 blue, region 5 purple and region 6 green. Figure 7 shows the mechanical device composed by the rods (2) and attached to the vertical displacement step motor (3), the horizontal displacement step motor (4) and the horizontal plane with the regions (1) where the mechanical device is positioned after the displacements. The electronic circuits for selecting, controlling and driving stepper motors can also be seen in figure below, position (5).

Figure 7: Mechanical device displacement system components.



Source: Authors (2022).

2.6 Tests of Neural Networks

Neural Network tests were performed with the image patterns used for the training and also with images similar to the positioning regions, but not participating in the training. Test results are obtained from the decision system described. For this generalization test an extension of the application was built, which loads the images in the neural network inputs.

3. Results

Following are the results of the tests with the image patterns used for the training of the Neural Networks and the results related to the generalizability of the networks, performed with untrained images.

3.1 Test results with training image standards

The training of Neural Networks 1 to 6 was performed with 300 cycles each. At the end of the training, the Neural Networks to recognize the image patterns of regions 1 to 6 were built with the final values of the synapse weights. The training images were then inserted into the Neural Network system, and each Neural Network system feeds the decision system; thus Neural Network 1 gives the decision system the value of $\phi 1$ (y_1), Neural Network 2 gives the decision system the value of $\phi 2$ (y_2), and this sequence is followed to Neural Network 6. Results are shown in Table 2.

Table 2: Results of the training of Neural Networks 1 to 6.

regions	*accuracy*	*identified pattern*
orange	75%	red
red	100%	
yellow	100%	
blue	75%	purple
purple	100%	
green	75%	purple
total accuracy	87,50%	

Source: Authors (2022).

3.2 Test results with untrained images

Neural Networks 1 to 6 were then submitted to the generalization test, that is, images that did not participate in the training were inserted in the Neural Networks, but with the image patterns of the respective regions of the mechanical arm positioning. Results are presented in Table 3.

Table 3: Results of generalization test.

regions	*accuracy*
orange	100%
red	100%
yellow	100%
blue	100%
purple	100%
green	100%
total accuracy	100,00%

Source: Authors (2022).

4. Discussion

The experiments performed with the Neural Network described by Algorithm 1 (Mario et al., 2008) provided the perspective that it would be possible, through its architecture with the use of the six outputs, to enhance convergence to a data set equivalent to a particular image pattern. Network response enables the reconstruction of an image from the sets of RGB components displayed in the outputs. The expectation of this work was to transform the subjective connotation of image analysis as a response to a pattern into a concrete response that could be expressed through numerical values. Through these same experiments it can be verified that the architecture with one output did not produce the expected results using similar methodology for the training of the networks, that is, there was no convergence for different numerical value ranges that could identify different image patterns.

Regarding the results, in tests with trained images, errors are characterized by network responses pointing to images from neighboring regions that would be the correct ones. Since neural networks use the RGB components of the images, it is relevant to mention that there is similarity between the data from the orange-red and salmon-blue regions, where two of the three region identification errors occurred.

Adjustments made following conditionals as well as decision system ranges after the training step resulted in improvements in mechanical device positioning system responses. The performance in generalization tests with images that did not participate in training, with correctness of all images, is a tendency to the first tests and qualifies the system composed by Artificial Neural Networks as viable for the recognition of image patterns.

5. Conclusions

The objective proposed at the beginning of this work was achieved by demonstrating the feasibility of using image pattern recognition, using a less complex Neural Network architecture than those currently used for this purpose.

The studies and experiments that inspired and supported this work also show relevant results, such as the feasibility of recognizing and reconstructing image patterns through their RGB components.

The prototype built to demonstrate the development of this work is characterized by the predominant use of open source hardware and software, resulting in the easy obtaining of all components, as well as the low cost involved. On the other hand, the methodology used in this work for the analysis of image patterns, allows its use using other programming languages, while the use can be extended to applications where the condition is determined by image.

Noting the similarity of all images used in the experiment, where the difference occurs only by the presence of the circular piece over one of the regions 1 to 6, it should be noted the performance of the tests of the Neural Networks, with accuracy of 87.5 % overall for the images that participated in the training, and 100% accuracy for the generalization tests.

Final considerations: the developed model that covers both software and hardware can be adapted in future works for robotics applications where the positioning of mechanical artifacts is necessary from the interaction with image recognition, with an approach to image recognition that highlights the treatment of its attributes from the RGB components, differing from works where the images are segmented and evaluated by parts. This type of approach to image recognition also allowed a lower complexity for the Artificial Neural Network used, since its architecture is simpler when compared to the state of the art, in which the use of Deep Artificial Neural Networks has been frequent for manipulation of digital images.

References

Baranauskas, J. A. & Monard, M. C. (2000). "*Reviewing some Machine Learning Concepts and Methods"*. Relatório Técnico 102, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos. ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_102.ps.zip.

Braga, A. P., de Carvalho, A. P. de Leon, Ludermir & T. B. (2012). "*Redes Neurais Artificiais – Teoria e Aplicações*". (2a ed.), 181 – 182, 198. Editora gen LTC.

Capelli, A. (2008). "*Automação Industrial – Controle do movimento e processos contínuos*". (2a ed.), 210. Editora Érica.

Faceli, K., Lorena, A. C., Gama, J. & De Carvalho, A. C. P. L. F. (2019). "*Inteligência Artificial – Uma Abordagem de Aprendizado de Máquina*". (2a ed.), 117. Editora gen LTC.

Gonzalez, R. C. & Woods, R. E. (2010). "*Processamento Digital de Imagens"*. (3a ed.), 265. Editora Pearson.

Haykin, S. (2001). "*Redes Neurais – Princípios e Prática*", (2a ed.), 183 - 259. Editora Bookman.

Newby, J. M., Schaefer, A. M., Lee, P. T., Forest, M. G. & Lai, S. K. (2018). "*Convolutional neural networks automate detection for tracking of submicron-scale particles in 2D and 3D*". National Academy of Sciences. https://doi.org/10.1073/pnas.1804420115. 115 no. 36 9026-9031.

Lakhili, Z., El Alami, A., Mesbah, A., Berrabou, A. & Qjidaa, H. (2018). "*Deformable 3D Shape Classification Using 3D Racah Moments and Deep Neural Networks*". Second International Conference on Intelligent Computing in Data Sciences. Procedia Computer Science. 148, 12 – 20.

Luger, G. F. (2013). "*Inteligência Artificial*". (6a ed.), Editora Pearson.

Mario, M. C., Da Silva Filho, J. I., Garcia, D. V., Fernandes, L.A. & Fernandes, C.L.M. (2018). "*Artificial Neural Network with Backpropagation algorithm applied to pattern recognition of digital images*". 50, Editora Paralogike.

Pedrini, H. & Schartz, W. R. (2008). "*Digital image analysis: principles, algorithms and applications* ". Publisher Thomson Learning. 22, 471.

Regmi, K. & Ali Borji, A. (2019). "*Cross-view image synthesis using geometry-guided conditional GANs*". Computer Vision and Image Understanding, 187, 102788. https://doi.org/10.1016/j.cviu.2019.07.008.

Russel, S., Norvig, P. (2004). "*Inteligência Artificial*", (2a ed.), Editora Campus - Elsevier.

Sharma, N., Jain, V. & Mishra A. (2018). "*An Analysis Of Convolutional Neural Networks For Image Classification* ". Proceeded by Computer Science 132. International Conference on Computational Intelligence and Data Science. 377 – 384.

Solomon, C. & Breckon, T. (2013). "*Fundamentals of digital image processing – a practical approach with examples in MATLAB*". Translated by José Rodolfo Souza, 1st edition, publisher LTC. 5, 8 - 10.

Soloman, S. (2012). "*Sensores e sistemas de controle na indústria*". (2a ed.), Editora gen LTC.

Wang, S., Hu, L., Li, L., Zhang, W., Huang, Q. (2020). "*Two-stream deep sparse network for accurate and efficient image restoration*". Computer Vision and Image Understanding. 200, 103029. https://doi.org/10.1016/j.cviu.2020.103029.

Wei, L., Jin, C., Ping, W., Cheng, J., Yongheng, M. & Keshiting. C. (2021). "*Dynamic Characteristics and Anti-slip Grasping of Two-Finger Translational Manipulator*". Frontiers in Neurorobotics. 15, https://www.frontiersin.org/article/10.3389/fnbot.2021.684317. 10.3389/fnbot.2021.684317.

Talon, T. & Pellegrino, S. (2022). "*Inextensible Surface Reconstruction Under Small Relative Deformations from Distributed Angle Measurements*". Int J Comput Vis. https://doi.org/10.1007/s11263-021-01552-x.

Tuncer, T., Dogan, S. & Ertam F. (2019). "*A novel neural network based image descriptor for texture classification*". Physica A: Statistical Mechanics and its Applications. 526, 120955.