Journal of Engineering and Technology for Industrial Applications, 2020. Edition. 23. Vol: 06
https://www.itegam-jetia.org
ISSN ONLINE: 2447-0228
DOI: https://dx.doi.org/10.5935/2447-0228.20200017

RESEARCH ARTICLE OPEN ACCESS

OBJECT DEFECT DETECTION BASED ON A VISION SYSTEM WITH A MICROCONTROLLER AND AN ARTIFICIAL NEURAL NETWORK

Ingrid Martins Valente Costa[1], Lorena Cândida Mendonça[2], Miguel Gonçalves de Freitas[3], Talles Marcelo Gonçalves de Andrade Barbosa[4] and Symone Gomes Soares Alcalá[5]

[1, 2, 3, 5] Federal University of Goiás – UFG, Aparecida de Goiânia – Goiás, Brazil.
[4] Pontifical Catholic University of Goiás – PUC Goiás, Goiânia – Goiás, Brazil.

Email: ingridmvalente@gmail.com, lorena.candida12@gmail.com, miguel.goncalves.freitas@gmail.com, talles.ucg@gmail.com, symonesoares@gmail.com

Received: Jun 22th, 2019

Accepted: Jun 15th, 2020

Published: June 30th, 2020

ABSTRACT

Vision systems have been widely employed in industries to automate the inspection process in products. Their use provides standardized, reliable and accurate inspections when compared to a human operator. Vision systems pass to machines the ability to view and automatically extract features in order to indicate abnormalities in products. This paper proposes a vision system for capturing and preprocessing digital images, besides classifying objects with defect and objects without defect using an Artificial Neural Network model. As a case study, digital images of boxes are acquired and classified on a conveyor belt. Tests reveal that the proposed system is able to classify accurately a box with defect and a box without defect in real time. The main contribution of this paper is the proposal of a system that performs automated inspections in products, in order to detect abnormalities, and it can be easily coupled, modularly, to the existing industrial platforms.

Keywords: Neural network applications, Machine vision, Quality assurance, Microcontroller, Image classification.

I. INTRODUCTION

In an industry, ensuring the quality of products and processes, through the optimization and automation of production lines, is indispensable for its success. In most cases, the quality of products is mainly related to the absence of defects and their occurrence can affect the credibility of a company. In this sense, one way of verifying product conformity is by performing visual inspection.

The visual inspection allows detecting abnormalities in the products in order to attend the rules and expectations of consumers. Generally, it is performed manually, resulting in high costs, failures and difficulties in standardization. With the evolution of the new technologies, vision systems have made possible the realization of automated inspections [1, 2]. Vision systems can automatically extract characteristics and indicate abnormalities in products. For example in [3] proposed a vision system to identify humans and cars in a real time video surveillance system.

In manufacturing systems, vision systems enable accuracy and repeatability in non-contact measurements by suppressing factors such as subjectivity, fatigue, vagueness and costs inherent to the human inspection [4]. Therefore, they are more efficient, safer and faster because they certify the quality of the product, since they go beyond the human capacity for visual detection.

A vision system can be divided into the following phases: image acquisition, segmentation, image improvement, feature extraction and pattern recognition [5]. Acquisition involves capturing the digital image of product. Segmentation and improvement are employed, respectively, to highlight the relevant portion and to increase image quality. Both steps eliminate aspects such as noise, which could jeopardize later steps. Feature extraction involves capturing relevant information about the segmented image that is essential for the product classification. The last phase is done by classification algorithms, which use the characteristics extracted to classify products according to a set of categories, such as, product with defect and product without defect.

The image acquisition step can be done using microcontrollers [6]. Microcontrollers have made it possible to create low-cost computing applications. They are a kind of “computers” that can be programmed to control circuits and perform specific tasks using previously elaborated commands. In addition to being easily programmable most of the times and taking up little physical space, microcontrollers are being increasingly targeted because of the low cost.

Therefore, professionals with knowledge on the handling of these devices are being increasingly demanded. In addition, microcontrollers have been used to control apparatus, answering machines, photocopiers, medical instruments, among others. Although microcontrollers are feasible in many applications, few studies in the production engineering contemplate their use.

Classification algorithms are mainly developed using artificial intelligence methods. Through these methods, intelligent agents can be created to perform tasks that require intelligence when done by humans [7]. According to [8], the artificial intelligence methods used for pattern recognition are mainly implemented using statistical approaches, syntactic/structural approach, Artificial Neural Networks (ANNs), fuzzy logic model and hybrid models. Among these, ANNs stand out for their fault tolerance. For example, in [9] developed an ANN to classify defects in the color and texture of oranges. Currently, most vision systems recognize color patterns and few are able to recognize defects.

Thus, this paper proposes a vision system for capturing and preprocessing digital images, besides classifying objects with defect and objects without defect using an ANN model, trained by the Scaled Conjugate Gradient Backpropagation (SCGB) algorithm. For this purpose, the following phases of the vision system are performed: image acquisition using a Logitech HD Webcam C270, segmentation using the Otsu's method, improvement by median filtering, feature extraction using co-occurrence matrix, and pattern recognition by the ANN model. As a case study, digital images of boxes are acquired and classified on a conveyor belt. Experimental results reveal that the proposed system vision is able to classify accurately the box with defect and box without defect in the conveyor belt.

The main contribution of this paper is to propose a vision system that performs automated inspections in products, in order to detect abnormalities, and it can be easily coupled, modularly, to the existing industrial platforms, as it only needs to incorporate a simple Webcam to the existing machinery. For this purpose, a small scale vision system was built to simulate the environment and conditions of the conveyor belt operation. Another contribution is that the proposed system does not require calibration, since through machine learning and ANN training, it does not require human interventions for maintenance purposes. Moreover, this paper performs a survey about the use of microcontrollers in production engineering applications in Brazil.

The rest of this paper is organized as follows. Section II describes the use of microcontrollers in production engineering applications in Brazil, and details the Arduino board. Section III describes the ANN model, its main architectures and the SCGB algorithm. Section IV details the proposed vision system and its main steps. Section V shows the experimental results of the proposed vision system using images acquired from a conveyor belt. And finally, Section VI presents some concluding remarks.

II. MICROCONTROLLERS

This Section discusses the use of microcontrollers in the production engineering area in Brazil, as well as the main concepts of the Arduino board

II.1 THE USE OF MICROCONTROLLERS IN PRODUCTION ENGINEERING APPLICATIONS

Microcontrollers appeared around the 70s with the development of the Intel 4004 microprocessor [10]. Currently, Raspberry and Arduino are the most common. Raspberry and Arduino focus on teaching basic computer skills in schools, but Arduino is also used for automation applications industrial [11]. Applications that use microcontrollers are often called embedded systems, because they are programmable computers designed to perform a specific task, such as controlling the temperature of an air conditioner, printers, and other tasks [12].

According to [13], microcontrollers are basically composed of the following peripherals: initialization managers, input and output ports, serial communication, timers, analog comparators and, finally, EEPROM (Electrically-Erasable Programmable Read-Only Memory).

In Brazil, most of the production engineering applications involving the use of microcontrollers fall into the areas related to Innovation Management and Technology Management. For example, in [14] proposed the use of microcontrollers to control Automated Guided Vehicles (AGVs) in industrial transportation processes; and [15] created interfaces between microcontrollers and other software, which together can easily perform the acquisition of industrial sensor data.

Although microcontrollers are feasible in a range of applications in production engineering, in Brazil, few papers portray their use and importance, and highlight their advantages. To demonstrate this, a survey was performed on October/28/2018 on the basis of the annals of the National Meeting of Production Engineering (ENEGEP, most well-known production engineering conference in Brazil) from the years 2010 to 2017, by the website of the Brazilian Association of Production Engineering (ABEPRO), using the strings “microcontrolador” (in English, “microcontroller”), “Arduino”, “Raspberry” and “sistema embarcado” (in English, “embedded system”). From the search, only 3 papers were found: one article in 2012 and two papers in 2016.

The paper of the ENEGEP 2012 was found with the string “embedded system”, and it uses MINI2440 microcontroller to monitor electrical energy in industrial systems [16]. On the other hand, the two papers of the ENEGEP 2016 were found with the word ``Arduino", and the first paper optimizes the capture of light energy from solar tracking to reduce energy costs [17]; while the second paper optimizes the use of street lamps [18]. Therefore, all the papers are related to the electric power area.

Through this survey, it is noted the importance of developing this study with microcontrollers in the production engineering area, and of showing the relevance of this device that is little explored in the production engineering area in Brazil.

II.2 THE ARDUINO BOARD

According to [18], the Arduino is a prototyping board based on the Atmel AVR microcontrollers, which can be programmed from a specific programming language based on C and C++ languages. According to [19], a development platform, such as Arduino, is a physical computing platform and in it, digital systems linked to other components measure variables in the physical environment, perform numerical calculations and can even make logical decisions, generating new variables in the environment physicist.

The abilities to control other physical devices and to receive and process data through an interface are advantages of using an electronic prototyping platform, such as Arduino. In addition, according to [20], the existence of several circuit boards, known as shields, that can be integrated with the Arduino to increase its capacity, is an advantage, since this capacity of expansion allows a range of applications of simple form and fast.

According to data from Eletrogate [21-24], Table 1 was developed. It presents the main Arduino boards available in the market and their corresponding microcontrollers, memory capacities, digital ports, analogue ports and clock speed (or CPU speed - Central Processing Unit).

As shown in Table 1, the Arduino Mega 2560 has the largest memory capacity and the largest number of digital and analog ports. Its larger number of ports, compared to the other Arduino boards, allows users to connect to a large number of devices, which can be advantageous in projects with a certain level of complexity.

Table 1: Comparison between the Arduino boards available on the market.

	NANO	**UNO**	**LEONARDO**	**MEGA 2560**
Micro-controller	ATmega 328	ATmega 328	ATmega32u4	ATmega25 60
FLASH memory	32KB	32KB	32KB	256KB
EEPROM memory	1KB	1KB	1KB	4KB
SRAM memory	2KB	2KB	2.5KB	8KB
Number of digital ports	14	14	20	54
Number of analog ports	8	6	12	16
Clock velocity	16MHz	16MHz	16MHz	16MHz

Source: Author, (2019).

Therefore, this paper uses Arduino MEGA 2560 for the development of the vision system. This Arduino is based on the ATmega2560 microcontroller and has 5 volts of voltage. Figure 1 shows the Arduino MEGA 2560 board.

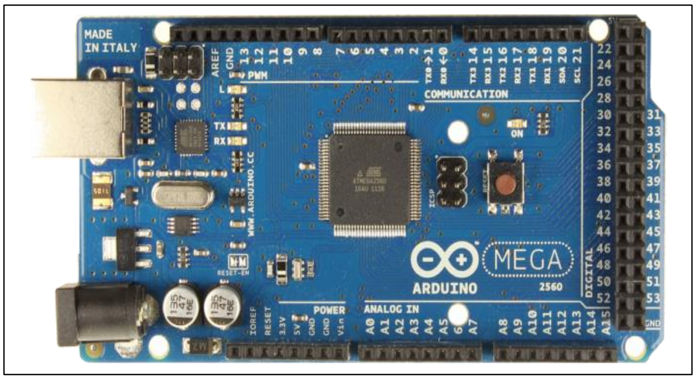

Figure 1: The Arduino MEGA 2560 board.
Source: Author, (2019).

III. ARTIFICIAL NEURAL NETWORKS

ANNs are artificial intelligence models inspired by biological neuron behaviors and have processing units, also known as neurons, and connections/weights between them, an architecture and a learning algorithm [7, 8]. After these steps, an output is generated and propagated to other neurons or to the environment [25].

The main characteristics of the ANN models are: *learning* - the ability of an ANN to initiate a learning without knowledge and be able to be trained using data; *generalization* - ability to generate the best output for an example not used in training; *massive potential parallelism* - neurons are triggered parallel throughout data processing; *robustness* - even if some neurons do not perform well, yet the whole system can work well; and *partial match* - which detonates that the known data does not exactly match with new events [26].

In literature, there are several types of ANN architectures, but Multilayer Perceptrons (MLPs), are the most popular ANN. They have one input layer, one or more hidden/intermediate layers and one output layer. In MLP, during estimation, the data propagates forward, from the input layer to the output layer. Researches have proved that MLPs are universal approximators [27]. For example, [28] demonstrated that any continuous function can be approximated by a continuous ANN having one hidden layer with neurons that use a sigmoidal non-linear activation function.

Single-hidden Layer Feedforward Networks (SLFNs) are MLPs with one hidden layer, and they have good approximation capabilities in many problems [29]. Therefore, the ANN models developed in this paper implement the SLFN architecture.

III.1 SINGLE-HIDDEN LAYER FEEDFORWARD NETWORK ARCHITECTURE

The SLFN architecture has an input layer with n_0 input neurons, an intermediate/hidden layer with n_1 hidden neurons, and an output layer with an output neuron, as shown in Figure 2.

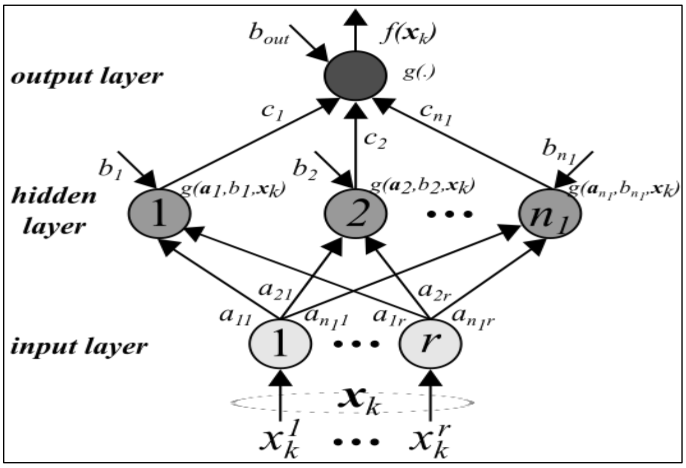

Figure 2: The SLFN architecture.
Source: Author, (2019).

Hidden neurons and output neurons have activation functions defined as $g(x)$. There are many activation functions including the *sigmoid* activation function, $g(x) = 1/(1 + \exp(-x))$, and the *linear* activation function $g(x) = x$. They perform a transformation as the last step to obtain the neuron's output value. To obtain the output value, consider a data set $\boldsymbol{D} = \{(\boldsymbol{x}_k, y_k)\}_{k=1}^{K}$ with K samples (with $x_k = [x_k^1, \cdots, x_k^R] \epsilon \mathbb{R}^R$, and

$y_k \in \{0,1\}$), a SLFN model with n_1 hidden neurons can be represented as:

SLFN model

$$f(\boldsymbol{x}_k) = g\left(\left(\sum_{j=1}^{n_1} g\left(\sum_{r=1}^{R} a_{jr}\, x_k^r + b_j\right) c_j\right) + b_{out}\right) = \hat{y}_k, (1)$$

for $k = [1, \cdots, K]$; where $\boldsymbol{a}_j = [a_{j1}, \cdots, a_{jR}]^{\mathrm{T}}$ is the weight vector connecting the r input neuron and the j-th hidden neuron (for $j = [1, \cdots, n_1]$; $\boldsymbol{c} = [c_1, \cdots, c_{n_1}]^{\mathrm{T}}$ is the weights vector connecting the n_1 hidden neurons and the output neuron; b_j is the bias of the j-th hidden neuron; b_{out} is the bias in the output layer; and $\hat{y}_k$ is the output prediction. As it can be seen, the operation the neurons is guided by the introduction of the inputs, calculation of the weighted sum and application of activation functions. After these steps, an output is generated and propagated to other neurons or to the environment [25].

The weights of an ANN should be obtained using a learning algorithm in order to minimize the error in all the K samples. This process is performed in several iterations in order to obtain the final weights. This process, also known as learning process, stops when an established stop criterion is reached.

Many learning algorithms have been developed for the learning process. The most known algorithm is the Back-Propagation (BP) algorithm. It uses a gradient descent method to define the NN parameters [30]. But, in most problems, the BP algorithm may generate overfitting. To overcome this limitation, other algorithms were developed, such as the Scaled Conjugate Gradient Backpropagation (SCGB) algorithm [7]. Below, the SCGB algorithm is introduced.

III.2 THE SCALED CONJUGATE GRADIENT BACKPROPAGATION ALGORITHM

The SCGB algorithm is a faster version of the BP algorithm, where second order partial derivatives are used to change the learning rate [31]. The main objective of the SCGB algorithm is to associate the confidence region approach with the conjugate gradient approach. Thus, SCGB has a step size scaling feature, which avoids a delayed line search by learning iteration. The SCGB algorithm indicates a super linear convergence in many problems [31].

As other learning algorithms, the SCGB algorithm learns when a set of training samples (from the initial input data) is processed iteratively. The process is done by comparing the ANN's output prediction and the real output. This is performed using the mean squared error, so that to minimize the error, it is necessary to change the weights for each training sample [32].

The learning process of the SCGB neural network is performed using Equations (2) and (3):

SCGB neural network – step 1

$$Z_{i,k}^{m} = b_{0,i}^{m} + \sum_{j=1}^{n_{m-1}} w_{j,i}^{m}\, O_{j,k}^{m-1} \qquad (2)$$

SCGB neural network – step 2

$$O_{j,k}^{m} = g\left(Z_{i,k}^{m}\right) \qquad (3)$$

where $Z_{i,k}^{m}$ is the obtained output in the layer m and the neuron i using a sample k; $b_{0,i}$ is the threshold (bias) of a neuron i in a hidden layer m; $w_{j,i}^{m}$ is the weight between the layer m with the neuron j and the layer $m-1$ with the neuron i; $O_{j,k}^{m-1}$ is the output value of the layer $m-1$ and the neuron i using sample k [33].

The error between the real output and the predicted output of the ANN model can be calculated according to the formula:

ANN error

$$E = \frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{n_2}\left(T_{j,k} - O_{j,k}^{2}\right)^2, \qquad (4)$$

where $T_{j,k}$ is the expected/real output of the output layer in neuron j using a sample k; and $O_{j,k}^{2}$ is the predicted output of the output layer in neuron j using a sample k.

IV. PROPOSED VISION SYSTEM FOR DETECTING OBJECT DEFECTS

The proposed vision system automatically captures digital images of an object moving on a conveyor belt. Figure 3 shows the interaction between the components of the system. The hardware consists of a Logitech HD Webcam C270, a Notebook equipped with Intel Core i7 3.4GHz and 8GB RAM, an Arduino Mega 2560 microcontroller connected to a protoboard, conveyor belt and infrared sensor module.

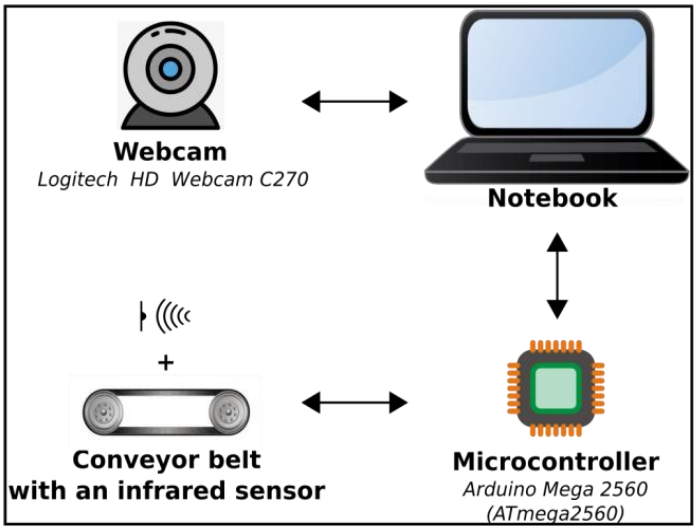

Figure 3: The main components of the proposed vision system.
Source: Author, (2019).

The software was developed in a MATLAB R2016a multi-platform environment, using the MATLAB Support Package for Arduino Hardware tool, which allows the integration and interaction between MATLAB and a microcontroller Arduino. The MATLAB R2016a also enables the processing of the acquired images of the objects, so that later they can be classified as object with defect and object without defect.

When the conveyor belt is activated, it transports an object (in this paper, a box), which is detected by an infrared sensor. When this occurs, a digital image of the object is captured and stored on the computer. If no object is on the conveyor belt, no action is performed.

Figure 4 shows the prototype of the vision system. It should be pointed that, if an object is detected, the software of the proposed vision system performs the following steps: image acquisition, segmentation, improvement, feature extraction and pattern recognition, as presented in Figure 5. The next Subsections detail each step of the proposed methodology.

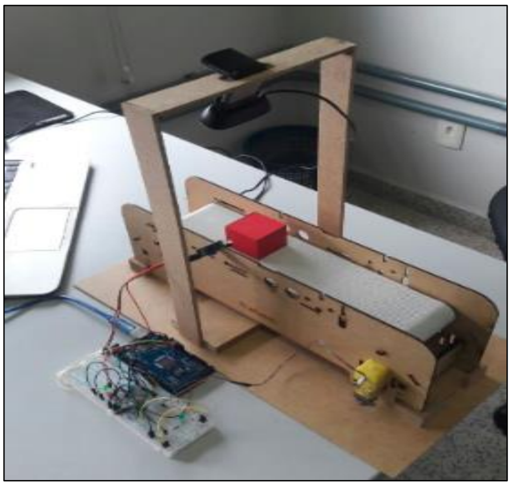

Figure 4: The prototype of the proposed vision system.
Source: Author, (2019).

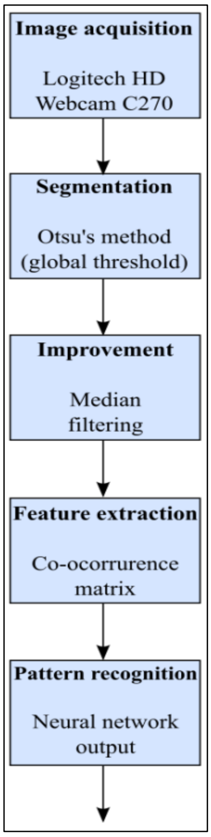

Figure 5: The steps the proposed vision system.
Source: Author, (2019).

IV.1 IMAGE ACQUISITION

The first step, *image acquisition*, obtains digital images of the boxes using the Logitech HD Webcam C270, with a resolution of 1280 × 960. The captured images are in the Red, Green and Blue (RGB) color system, which are then converted to the YCbCr color space.

In this case, the *Y* channel contains the luminance, the *Cb* channel contains the blue chrominance and the *Cr* channel contains the red chrominance. Moreover, the *Y* channel represents the light intensity and the *Cb* and *Cr* channels represent the colorization. The change in the color space is necessary to extract characteristics related to the luminance and chrominance of the digital images [9]. These characteristics are essential for segmenting images.

IV.2 SEGMENTATION

The *segmentation* step extracts the object from the portion considered as the background of the digital image. To do this, the Otsu's method is used [34]. It obtains a global threshold from the *Cr* channel a digital image. The threshold acts as a border, delimiting the space filled by the object and the space known as the background.

IV.3 IMPROVEMENT

The next step, *improvement*, generates digital images with higher quality, allowing a greater efficiency of the ANN learning algorithm. In this process, remaining noises of the segmentation process are eliminated using median filter, where each pixel of the image is replaced by the median of the neighborhood. In this paper, the adopted neighborhood is 5×5.

IV.4 FEATURE EXTRACTION

An efficient technique for extracting image texture information is the Gray Level Co-occurrence Matrix (GLCM), where the texture is classified by the spatial distribution of gray levels in a neighborhood. The matrix retracts the combinations of pixel brightness values (gray levels) in tabular form. It shows how often a pixel value with gray level u value occurs in relation to another pixel value, known as the neighbor pixel, with gray level v value. GLCM can be used to calculate texture characteristics such as contrast, entropy, energy and homogeneity.

The next step is to extract characteristics of the image. This is done by using GLCM, where contrast is the adopted characteristic [9]. In the GLCM method, for each channel, four co-occurrence matrices are determined employing distance equal to 1 and 4 directions, which are 0º, 45º, 90º and 135º.

Afterward, a contrast value is obtained using each matrix, totaling 12 characteristics. In addition, the average value of each channel is calculated. The calculated averages correspond to the general values of each channel in YCbCr of the segmented image. Thus, using an image, 15 characteristics (features) are obtained, and they will be used as inputs in the classification algorithm.

For the ANN training algorithm, it is necessary to create a data set consisting of two matrices, $\boldsymbol{X}$, the input matrix, and $\boldsymbol{X}$, the output matrix, where each column of them contains information about an acquired image. The input matrix $\boldsymbol{X}$ contains the features of all acquired images; while the output matrix $\boldsymbol{Y}$ contains the classes of all captured images.

The sizes of $\boldsymbol{X}$ and $\boldsymbol{Y}$ are $15 \times K$ and $2 \times K$, respectively, where K is the number of acquired images. For each image c, its 15 features, $x_{1,c}, \cdots, x_{15,c}]^{\mathrm{T}}$, are stored in a column of $\boldsymbol{X}$; and its class, $y_{1,c}, \cdots, y_{2,c}]^{\mathrm{T}}$, in a column of $\boldsymbol{Y}$. Being that, there are two distinct classes: class i - box without defect, and class ii - box with defect. For an image c belonging to the class i, the outputs are set as $y_{1,c} = 0$ and $y_{2,c} = 1$; while for an image c belonging to the class ii, the outputs are set as $y_{1,c} = 1$ and $y_{2,c} = 0$.

IV.5 PATTERN RECOGNITION

The pattern recognition step was developed using a SLFN trained with the SCGB algorithm, described in Subsection III.2. To do this, the data set was randomly divided in training data set (65%), testing data set (20%) and validation data set (15%); where the training data is used for the ANN training, the validation data is employed to verify ANN accuracy during the training and the test data set is used to authenticate ANN accuracy after the training, using a data set which was not used in training.

After this process, the best number of neurons in the hidden layer (n_1) is determined. It was selected using the 10-fold cross-validation method that randomly divides a set of data $\boldsymbol{D}$ into 10 mutually exclusive subsets, $\boldsymbol{D}_1, \cdots, \boldsymbol{D}_{10}$. The hidden layer activation function is hyperbolic tangent sigmoid (tansing), while the output layer activation function is soft maximum (softmax).

The number of hidden neurons is selected by varying it in the interval of [2, 20]. This value is chosen based on the best performance on a 10-fold cross-validation using the training data set, where the best number of hidden neurons is chosen as the one that maximizes the mean testing performance on the 10-folds using the Mean Squared Error (MSE) [29].

For a specific value of hidden neuron, 10-fold cross-validation obtains the average of the predictive error values of the 10 subsets (each one being used as testing data set), while the other 9 sets are randomly divided into a set of data of training (75%) and validation (25%) [35, 36]. It is noted that the method selects the best number of hidden neurons that first shows the lowest percentage of incorrect classifications. Then an ANN with SLFN architecture is trained with the SCGB algorithm and the best number of hidden neurons.

V. EXPERIMENTAL RESULTS

This Section reports the main experimental results obtained with the ANN model (Subsection V.1), and the proposed vision system in real time operation (Subsection V.2).

V.1 ARTIFICIAL NEURAL NETWORK TRAINING

In this Subsection, the results described were obtained using MATLAB R2016a software. In the first step of the proposed methodology, digital images of boxes were acquired on a conveyor belt. Then, the camera (Logitech HD Webcam C270) acquired 430 images of the boxes, being that 215 images were obtained of a box with defect (defective box) and 215 images were captured of a box of without defect (nondefective box); where both boxes have shape of a cube, with edges of 3.5cm, and images captured in different angle and illumination conditions. The box with defect has paint defect on top surface.

Figures 6 and 7 show some images of the box without defect and some images of the box with defect, respectively. Figures 8 and 9 show the result of two images after the segmentation and improvement steps.

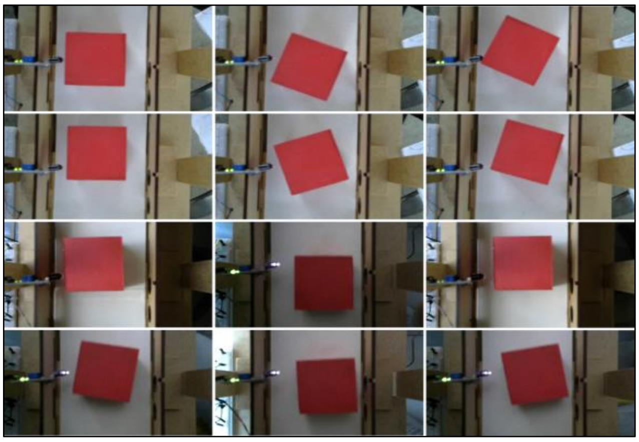

Figure 6: Some acquired images of the box without defect.
Source: Author, (2019).

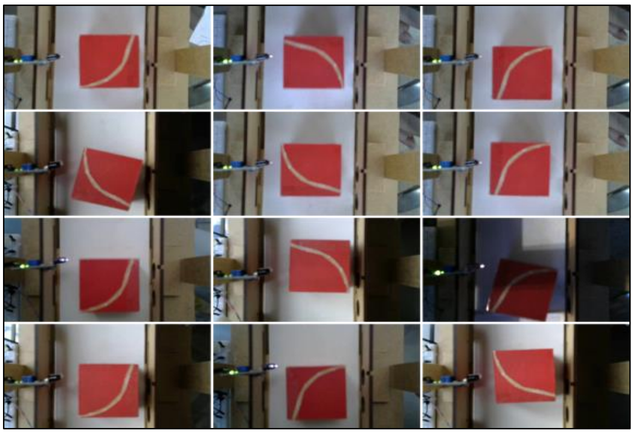

Figure 7: Some acquired images of the box with defect.
Source: Author, (2019).

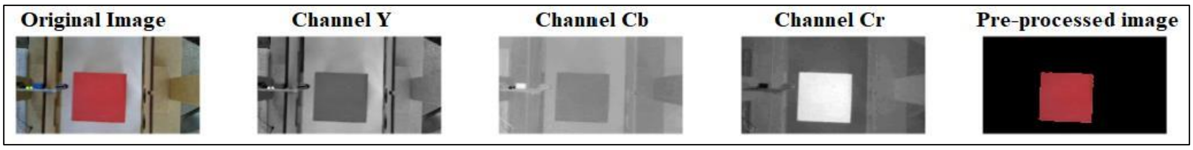

Figure 8: A preprocessed image of the box without defect.
Source: Author, (2019).

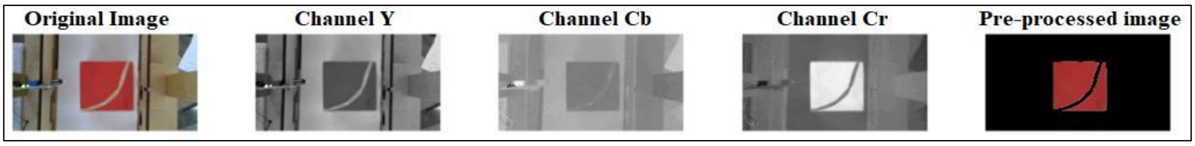

Figure 9: A preprocessed image of the box with defect.
Source: Author, (2019).

After the improvement step, the feature extraction step was performed and the data set resulting from the extraction was stored on an input matrix $\boldsymbol{X}$ of size 15×430 and an output $\boldsymbol{Y}$ matrix of size 2×430. Then, $\boldsymbol{X}$ and $\boldsymbol{Y}$ were used to training (65%), validate (15%) and test (20%) the ANN model. Furthermore, to determine the best number of hidden neurons in the hidden layer, the 10-fold cross-validation method was employed using only the training data. For further information, see Subsection IV.5.

The performance from 10-fold cross-validation method for each hidden neuron number is shown in Figure 10. The performance was measured using the average percentage (in the 10 test subsets) of incorrectly classified images.

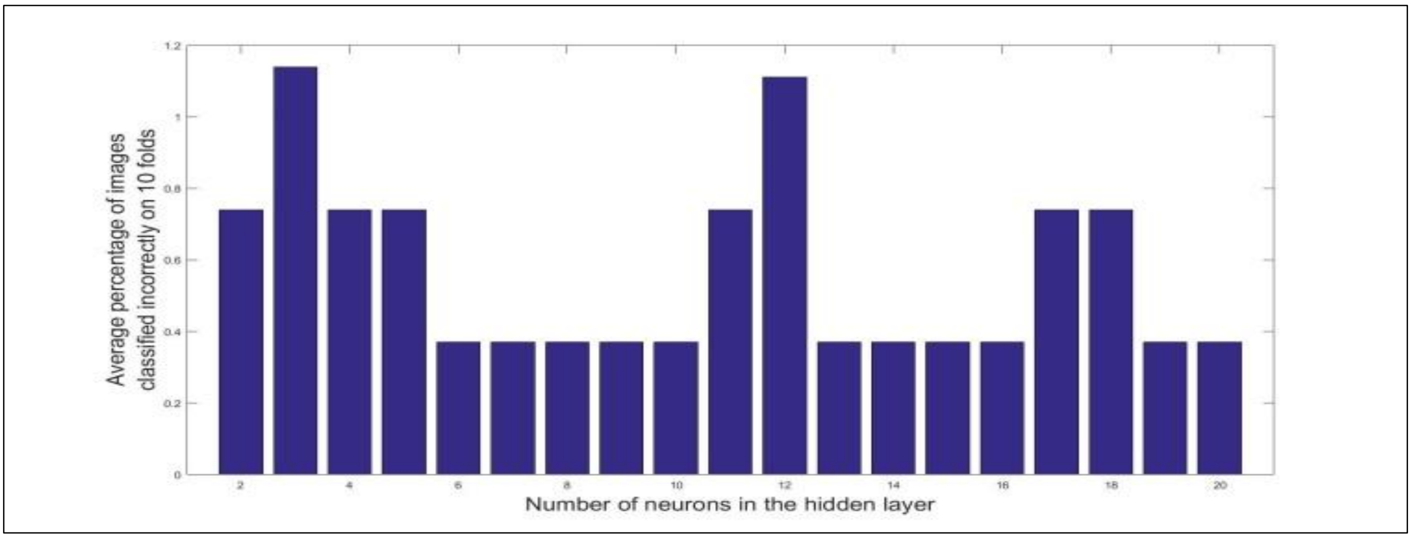

Figure 10: ANN performance in the 10-fold cross-validation method.
Source: Author, (2019).

In Figure, it was verified that the smallest error was obtained with numbers of hidden neurons equal to 6, 7, 8, 9, 10, 13, 14, 15, 16, 19, 20. Then, the smallest number of hidden neurons (i.e. 6 hidden neurons) was chosen. Figure 11 shows the architecture of the designed ANN model. As it can be seen, the ANN architecture is composed by 15 neurons in the input layer (15 features), 6 neurons in the hidden layer, and 2 neurons the output layer (2 classes in the formats of {0,1} and {0,1}).

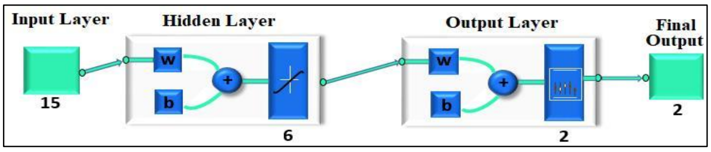

Figure 11: Architecture of the ANN model with 6 hidden neurons in the hidden layer.
Source: Author, (2019).

As described previously, the activation functions are sigmoid hyperbolic tangent (tansig) for the hidden layer and softmax for the output layer. The ANN training was done using the SCGB algorithm with the standard parameters of the MATLAB software: maximum of iterations equal to 1000, performance goal equal to 0, among others [26].

The ANN performance during the training was measured using the cross-entropy function [25]. Figure 12 shows the ANN performance in training and validation data over time. The ANN training was automatically stopped after 56 iterations.

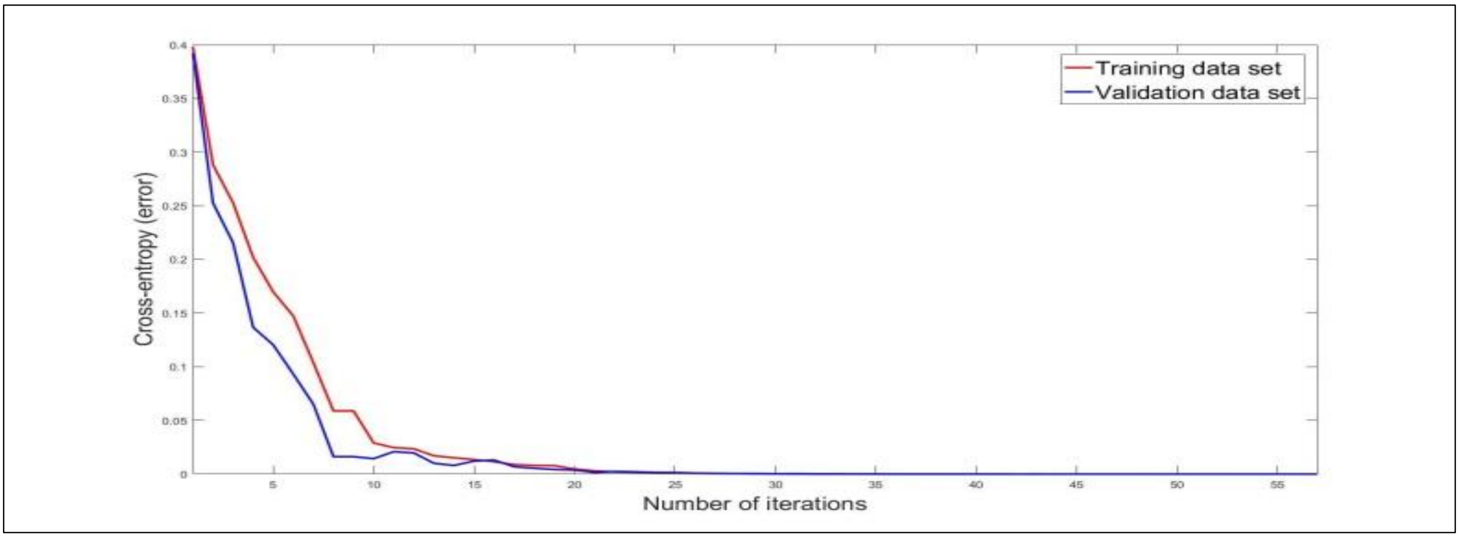

Figure 12: Performance of the ANN model in the learning process.
Source: Author, (2019).

Table 2 shows the percentage of correct and incorrect ANN classifications in training, validation, and test data. It was noticed that ANN performance was 100% in all data, such that ANN was able to classify efficiently images of the box with defect and the box without defect in all data. Therefore, the proposed methodology has effectiveness in the classification of the box without defect and the box with defect.

Table 2: Percentages of correct and incorrect classifications of the ANN model.

	Correct classifications	Incorrect classifications
Training	100%	0%
Validation	100%	0%
Test	100%	0%

Source: Author, (2019).

It should be pointed that, in this testing scenario, the boxes classifications were performed using images acquired previously, that is, the images were stored in a database. This testing scenario is necessary to create and validate the ANN model for the vision system.

V.2 REAL-TIME PERFORMANCE OF THE PROPOSED VISION SYSTEM FOR OBJECT DEFECT DETECTION

In order to analyze the real-time performance of the vision system, 60 digital images were acquired, one by one, by the proposed vision system. The acquired images consist of 30 images of a box with defect and 30 images of a box without defect, being that all the images were obtained in different angles and lighting conditions.

The main difference of this experiment (testing scenario) is that after acquiring the image of a box, it is automatically classified (with or without defect) by the vision system. While in the previous experiment (testing scenario), a set of images was simultaneously acquired and applied to train the ANN. Figure 13 shows some images acquired by the camera in this experiment.

Table 3 shows the percentages of boxes classified correctly and incorrectly by the vision system. Therefore, all the boxes were classified accurately by the proposed vision system. Thus, even in a real-time operation, the ANN model is efficient to classify boxes with and without defects in painting.

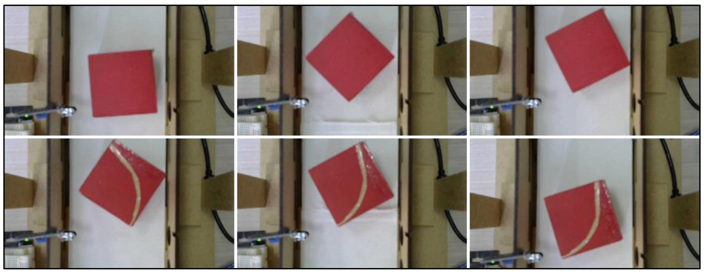

Figure 13: Some acquired images from proposed vision system in real time operation.
Source: Author, (2019).

Table 3: Percentages of correct and incorrect classifications of the proposed vision system in real time operation.

	Correct classifications	Incorrect classifications
Non-defect box	100%	0%
Defect box	100%	0%

Source: Author, (2019).

VI. CONCLUSION

The use of digital images of objects to control the quality of products is indispensable in order to avoid failures and additional costs resulting from manual inspection. This paper proposed a vision system which performs image acquisition using a Logitech HD Webcam C270, segmentation using the Otsu's method, improvement by median filtering, feature extraction using co-occurrence matrix, and pattern recognition using an ANN model. The proposed vision system was able to capture and classify accurately, in real time, images of a box with defect and a box without defect on a conveyor belt.

However, the prototype has a limitation. The used infrared sensor has problems when exposed to high level of illumination. In this situation, the infrared sensor detects an obstacle, even if an object is not in the conveyor belt. Therefore, further works should be devoted to test other infrared sensors.

Moreover, in order to make the proposed vision system more efficient, other improvements can be done, such as the integration of a Radio-Frequency Identification (RFID) model, a WiFi wireless communication module and a robotic arm in the vision system. In addition, as future work, it is expected the real application of the proposed vision system in an industry with the purpose of verifying its accuracy in a real production line.

VII. REFERENCES

[1] C. Iglesias, J. Martínez, J. Taboada. Automated vision system for quality inspection of slate slabs. Computers in Industry, vol. 99, pp. 119-129, 2018.

[2] O. Nir, Y. Parmet, D. Werner, G. Adin, I. Halachmiad. 3D Computer-vision system for automatically estimating heifer height and body mass. Biosystems Engineering, vol. 173, pp. 4-10, 2018.

[3] A. A. Shafie, A. B. M. Ibrahim, M. M. Rashid. Smart Objects Identification System for Robotic Surveillance. International Journal of Automation and Computing, vol. 11, no. 1, pp. 59-71, 2014.

[4] F. F. Feliciano, I. L. Souza, F. R. Leta. Visão Computacional Aplicada à Metrologia Dimensional Automatizada: Considerações Sobre sua Exatidão. Engevista, vol. 7, no. 2, 2010.

[5] A. R. Backes, J. J. M. S. Junior. Introdução à Visão Computacional Usando MATLAB, Rio de Janeiro: Alta Books Editora, 2016.

[6] M. Tresanchez, A. Pujol, T. Pallejá, D. Martínez, E. Clotet, J. Palacín. A proposal of low-cost and low-power embedded wireless image sensor node for IoT application. Procedia Computer Science, vol. 134, 2018.

[7] S. J. Russel, P. Norvig. Artificial Intelligence: A Modern Approach. Malaysia: Pearson Education Limited, 2010.

[8] N. Mohanan, A. Ahmad, A. A. Ghani, M. Al-Nadabi, L. Khriji, S. Al-Busaidi. A Brief Description of Pattern Recognition Techniques. Imperial Journal of Interdisciplinary Research, vol. 2, no. 8, 2016.

[9] G. Capizzi, G. L. Sciuto, C. Napoli, E. Tramontana, M. Wozniak. A Novel Neural Networks-based Texture Image Processing Algorithm for Orange Defects Classification. International Journal of Computer Science & Applications, vol. 13, no. 2, pp. 45-60, 2016.

[10] M. M. Breve, M. A. C. Bernuy. Introdução ao Desenvolvimento de Sistemas com Microcontroladores. In Seminário de Perquisa Jr., Universidade Tecnológica Federal do Paraná, Cornélio Procópio, vol. 3, 2012.

[11] N. A. Martins. Sistemas Microcontrolados. Editora Novatec, 2005.

[12] A. B. Nunes. Análise de Resultados Benchmarks para Escalonamento de Tarefas de Tempo Real em Máquinas Heterogêneas, B. Sc. dissertation, Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Ponta Grossa, PR, Brazil, 2016.

[13] U. Lopes, T. R. Bonfim. Implantação de Laboratório Acadêmico de Microcontrolador em Instituição de Ensino Superior. Revista de Ciências Exatas e Tecnologia, vol. 2, no. 2, pp. 53-61, 2007.

[14] A. D. Nogueira, V. Teixeira, W. R. de Freitas. Aplicação de Robótica em Processos de Transporte Industrial. Technical Report, Curso de Pós Graduação em Automação e Controle Industrial do Centro Universitário de Lins (Unilins), Lins, SP, Brazil, 2015.

[15] A. O. Tucci, M. J. da Cunha, A. S. de Morais, J. S. de Morais. Proposta de Desenvolvimento de uma Plataforma Computacional de Integração entre o Matlab e o Arduino Aplicado em Processos Industriais. In Conferência de Estudos em Engenharia Elétrica, vol. 12, pp. 1-4, 2014.

[16] J. A. Back, L. P. C. Tedesco. Proposta de um Sistema para Monitoração e Análise de Energia Elétrica em Ambientes Industriais. In Encontro Nacional de Engenharia de Produção (ENEGEP), vol. 32, pp. 1-13, 2012.

[17] C. A. S. de Miranda, S. L. A. Viera, A. P. C. Belem, L. F. V., N G. D. Gomes. Desenvolvimento de Sistemas Híbridos de Iluminação. In Encontro Nacional de Engenharia de Produção (ENEGEP), vol. 36, pp. 1-13, 2016.

[18] D. C. F. da Fonseca, U. E. F. Alves, F. A. G. Matos, M. K. A. Silva, I. G. A. M. da Silva. Eficiência Energética da Iluminação Pública Automatizada: Estudo de Caso na Cidade de Caruaru. In Encontro Nacional de Engenharia de Produção (ENEGEP), vol. 36, pp. 1-12, 2016.

[19] I. B. Q. da Araújo, F. V. Souto, A. G. da Costa Junior, C. P. de Sousa. Desenvolvimento de um Protótipo de Automação Predial/Residencial Utilizando a Plataforma de Prototipagem Eletrônica Arduino. In Congresso Brasileiro de Educação em Engenharia, vol. 40, pp. 1-9, 2012.

[20] A. R. da Silva. Sistema Inteligente de Controle e Monitoramento de Ambiente de Laboratórios de Análises Químicas, Master thesis, Software engineering, Instituto Metrópole Digital, Universidade Federal do Rio Grande de Norte, Natal, RN, Brazil, 2016.

[21] Eletrogate. Nano V3.0 + Cabo USB para Arduino, [Online], Available: https://www.eletrogate.com/nano-v3-0-cabo-usb-para-arduino, March 21, 2019.

[22] Eletrogate. Uno R3 + Cabo USB para Arduino, [Online], Available: https://www.eletrogate.com/uno-r3-cabo-usb-para-arduino, March 21, 2019.

[23] Eletrogate. Leonardo R3 + Cabo USB para Arduino, [Online], Available: https://www.eletrogate.com/leonardo-r3-cabo-usb-para-arduino, March 21, 2019.

[24] Eletrogate. Mega 2560 R3 + Cabo USB para Arduino, [Online], Available: https://www.eletrogate.com/mega-2560-r3-cabo-usb-para-arduino, March 21, 2019.

[25] Y. Uzun, K. Bicakci. A second look at the performance of neural networks for keystroke dynamics using a publicly available dataset. Computers & Security, vol. 31, no. 5, pp. 717-726, 2012.

[26] N. K. Kasabov. Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering, 2nd ed., Massachusetts Institute of Technology: A Bradford book, pp. 18, 1998.

[27] K. Hornik, M. Stinchcombe, H. White. Multilayer Feedforward Networks are Universal Approximators. Neural Networks, vol. 2, no. 5, pp. 359-366, 1989.

[28] G. Cybenko. Approximation by Superpositions of a Sigmoidal Function. Mathematics of Control, Signals and Systems, vol. 2, no. 4, pp. 303-314, 1989.

[29] S. G. Soares, R. Araújo. An adaptive ensemble of on-line extreme learning machines with variable forgetting factor for dynamic system prediction. Neurocomputing, vol. 171, pp. 693-707, 2016.

[30] D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning Internal Representations by Error Propagation. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, pp. 318-362, 1986.

[31] M. F. Moller. A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks, vol. 6, no. 4, pp. 525-533, 1993.

[32] S. Ekinci, K. Çarman, H. Kahramanli. Investigation and modeling of the tractive performance of radial tires using of road vehicles. Energy, vol. 93, pp. 1953-1963, 2015.

[33] D.-J. Li, Y.-Y. Li, J.-X. Li, Y. Fu. Gesture recognition based on BP neural network improved by chaotic genetic algorithm. International Journal of Automation and Computing, vol. 15, no. 3, pp. 267-276, 2018.

[34] N. Otsu. A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, vol. 9, no. 1, pp. 62-66, 1979.

[35] S. Soares, C. H. Antunes, R. Araújo. A Genetic Algorithm for Designing Neural Network Ensembles. In Proceedings of the 14th International Conference on Genetic and Evolutionary Computation Conference (GECCO'12), p. 681-688, 2012.

[36] R. Kohavi. A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. The International Joint Conference on Artificial Intelligence (IJCAI), vol. 14, no. 2, pp. 1137-1145, 1995.