Expert Systems With Applications 85 (2017) 114–122

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

# Deep learning for biological image classification

Carlos Affonso[a,*], André Luis Debiaso Rossi[a], Fábio Henrique Antunes Vieira[a], André Carlos Ponce de Leon Ferreira de Carvalho[b]

[a] UNESP - Universidade Estadual Paulista, Julio de Mesquita Filho, Brazil
[b] ICMC - USP Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brazil

ARTICLE INFO

*Article history:*
Received 3 June 2016
Revised 11 May 2017
Accepted 16 May 2017
Available online 17 May 2017

*Keywords:*
Wood classification
Deep learning
Image classification
Machine learning

ABSTRACT

A number of industries use human inspection to visually classify the quality of their products and the raw materials used in the production process, this process could be done automatically through digital image processing. The industries are not always interested in the most accurate technique for a given problem, but most appropriate for the expected results, there must be a balance between accuracy and computational cost. This paper investigates the classification of the quality of wood boards based on their images. For such, it compares the use of deep learning, particularly Convolutional Neural Networks, with the combination of texture-based feature extraction techniques and traditional techniques: Decision tree induction algorithms, Neural Networks, Nearest neighbors and Support vector machines. Reported studies show that Deep Learning techniques applied to image processing tasks have achieved predictive performance superior to traditional classification techniques, mainly in high complex scenarios. One of the reasons pointed out is their embedded feature extraction mechanism. Deep Learning techniques directly identify and extract features, considered by them to be relevant, in a given image dataset. However, empirical results for the image data set have shown that the texture descriptor method proposed, regardless of the strategy employed is very competitive when compared with Convolutional Neural Network for all the performed experiments. The best performance of the texture descriptor method could be caused by the nature of the image dataset. Finally are pointed out some perspectives of futures developments with the application of Active learning and Semi supervised methods.

## 1. Introduction

Quality analysis activities are often used by industries to ensure the quality of their products. These activities are usually carried out by human inspection, mainly by visually scanning the products in a production line. The inspection allows correction of problems and discards of defective products, resulting in a better quality of the final production. However, the use of human beings in the quality assessment adds subjective factors to this process and, due to problems like distraction, stress, and fatigue, can accept products whose quality is below the desired level. These problems show the importance of the use of efficient image classification techniques to improve the quality control in production lines (Affonso, Sassi, & Barreiros, 2015).

Frequent obstacles that arise when using these techniques are the design and tuning of automated image classification system since various aspects must be taken into consideration. Besides, considering that wood presents, as natural raw material, a variety of macroscopic and physical features, such as weight (different moisture content), color (variation), odor, hardness, texture, and surface appearances, its distinction becomes even harder.

In recent years, important efficiency gains have been achieved by machine vision systems, due to the development of high technology camera sensors and advances in processing capacity. Meanwhile, the price of systems based on cameras has decreased, enabling a cost-efficient classification solution environment for the quality of an extensive variety of products.

In complex problems as image classification, the capture of the essential features must be carried out without a priori knowledge of the image. Therefore, modeling by traditional computational techniques is quite difficult, considering the complexity and non-linearity of image systems.

Although texture has not a clear definition, such descriptors have a wide application on image classification, computer vision, and similar fields. Hossain and Serikawa (2013) surveyed a group of texture datasets from related to different areas of medicine and natural environment.

[*] Corresponding author.
*E-mail addresses:* affonso@itapeva.unesp.br, affonso.unesp@gmail.com (C. Affonso), alrossi@itapeva.unesp.br (A.L.D. Rossi), fhavieira@itapeva.unesp.br (F.H.A. Vieira), andre@icmc.usp.br (A.C.P.d.L.F. de Carvalho).

http://dx.doi.org/10.1016/j.eswa.2017.05.039
0957-4174/© 2017 Elsevier Ltd. All rights reserved.

Local binary patterns (LBP) is one of the most used descriptor considering its resistance to light changes, low computational cost and ability to classify using fine details (Nanni, Brahnam, & Lumini, 2012). Different texture descriptors have been proposed such as, Histograms of Oriented Gradient (Dalal & Triggs, 2005), wavelets (Unser, 1995) and Gabor filter (Mehrotra, Namuduri, & Ranganathan, 1992). However the most traditional is Haralick's texture descriptor (Haralick, Shanmugam, & Dinstein, 1973).

Due to the shortcomings of the manual process, Machine Learning (ML) algorithms have been widely used for classification and clustering of wood materials (Gonzaga, de Franca, & Frere, 1999). The representation of the data provided as the "experience" to these algorithms has strong influence on their performance (Bengio, 2009).

A number of features usually requires computational complexity and even greater runtime. Moreover, the noise in the database caused by excessive image features can cause a reduction in its capacity of representation.

In recent studies, Deep Learning (DL) techniques presented better predictive performance than state-of-the-art algorithms in many domains, including image classification (Krizhevsky, Sutskever, & Hinton, 2012). DL deals with the problem of data representation by introducing simpler intermediate representations that allow to combine them in order to build complex concepts. Therefore, it is unnecessary to apply many preprocessing techniques to extract features, which represent the image data (Bengio, 2009).

Artificial Neural Networks (ANN) are a quintessential example of DL model. Although ANN dates back to the 1950s, researchers now are able to train deeper structures than it had been possible before. The idea of using a higher number of layers, the multilayer network, is justified by the learning algorithm used to train a network in image classification (Al-Allaf, Abdalkader, & Tamimi, 2013).

On the other hand, because of its complex structure, DL needs a large volume of data to generate models with high predictive performance and, consequently, has high computational cost. Whereas recent works suggest that deep architectures might be more accurate, their training was unsuccessful until the recent uses of unsupervised pre-training (Bengio, 2009). That could happen because the gradient-based training of Convolutional Neural Network (CNN) achieves some local minimum and additionally for deeper architecture became more difficult to obtain satisfactory result (Bengio, Lamblin, Popovici, Larochelle et al., 2007; Erhan, Manzagol, Bengio, Bengio, & Vincent, 2009).

A successful DL algorithm based on ANN is the CNN. Some studies suggest CNN is superior to traditional learning algorithms, such as K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), and Support Vector Machines (SVM) for image classification (Chen, Xiang, Liu, & Pan, 2014; Ferreira & Giraldi, 2017; Makantasis, Protopapadakis, Doulamis, Doulamis, & Loupos, 2015; Neubauer, 1998; Norlander, Grahn, & Maki, 2015; Park, Kwon, Park, & Kang, 2016).

The literature points out DL Architecture as the best performance solution for image classification. However, its computational cost is much higher than the traditional techniques using texture descriptors (Gibert, Patel, & Chellappa, 2017; Krizhevsky, Sutskever, & Hinton, 2012).

The main objective of this paper was to analyze CNN to classify wood boards regarding their quality. The models were trained on a image data set, where each instance is a collection of pixel values representing a wood board image in grayscale. This data set presents three classes, according to their quality, with restricted examples.

Moreover, the CNN performance is compared to traditional learning algorithms, namely DT, ANN, and KNN. Each instance of the training data for these algorithms is a set of texture descriptors extracted from a wood board.

Since there is a "limited" number of instances to train the CNN, it is hypothetical that CNN achieves similar predictive performance compared to these algorithms using texture descriptors.

In this way, the traditional techniques using texture descriptors seems to be the smart choice for this investigation problem, once they present the advantage of lower computational cost.

In order to test the hypothesis, the classification accuracy for these two approaches was validated, such as DL techniques (through CNN) and Texture Descriptors (through Haralick's descriptors).

The next sections are organized as follows. Section 2 gives a brief introduction of DL techniques and focuses on relevant similar studies. Section 2.3 describes the texture descriptor methods proposed and investigated in this paper. Section 4 presents the experimental evaluation of DL architectures and discusses the results. Finally, Section 6 presents the main conclusions and future research directions.

## 2. Deep learning

The recent revival of DL techniques was triggered by the works on learning representations, or more traditional models (Hinton et al., 2012). DL architectures appear to solve problems that require complex highly-varying functions. Besides that, they usually involve such problems with very large, and in most cases, nonlabeled data set.

In order to deal with it, DL techniques learn characteristic hierarchies with features from higher levels of hierarchy formed by a composition of lower level features (Bengio, 2009).

DL assimilates complex behaviors with expansive information sets to select effective characteristics automatically by neural network structures in quite profound layers. The model achieves such goals adopting unsupervised layers succeeded by supervised ones, applying learning-teaching to signal data (Kim, Choi, & Lee, 2015).

### *2.1. Convolutional neural network*

CNN has attracted a high interest in the image and speech classification scientific communities, since its topology is more similar to biological systems. Another main characteristic is its receptive fields, which was inspired by the cat's visual cortex (Hubel & Wiesel, 1962).

The CNN topology is based on three main concepts, namely: local receptive fields, shared weights and spatial or temporal sampling (LeCun, Bengio, & Haffner, 1998). CNN can eliminate the feature extraction process imputing the network directly with normalized images. Typically, an image data set contains many hundred pixels.

If a full network is considered, each neuron is connected to every pixel. Therefore, the computational cost and the memory requirements would be unfordable. Another deficiency on the unstructured fully connected network for image classification is its non-acceptance for local distortions on receptive fields.

The spatial invariance is obtained through the shared weight across the image (LeCun et al., 1990). Subsampling is an important strategy in object recognition, as it helps achieve invariance to distortions of the visual image.

Because of its own nature, image data set has a strong spatial correlation. In order to deal with that, CNN restricts the receptive fields. Another characteristic of CNN is the shared weights. In this way, a set of pixels in a receptive field located at different places on an image, has identical weight vectors, which outputs constitute a feature map.

This operation could be considered as the convolutional transformer. Each feature map is followed by a layer that performs a lo-

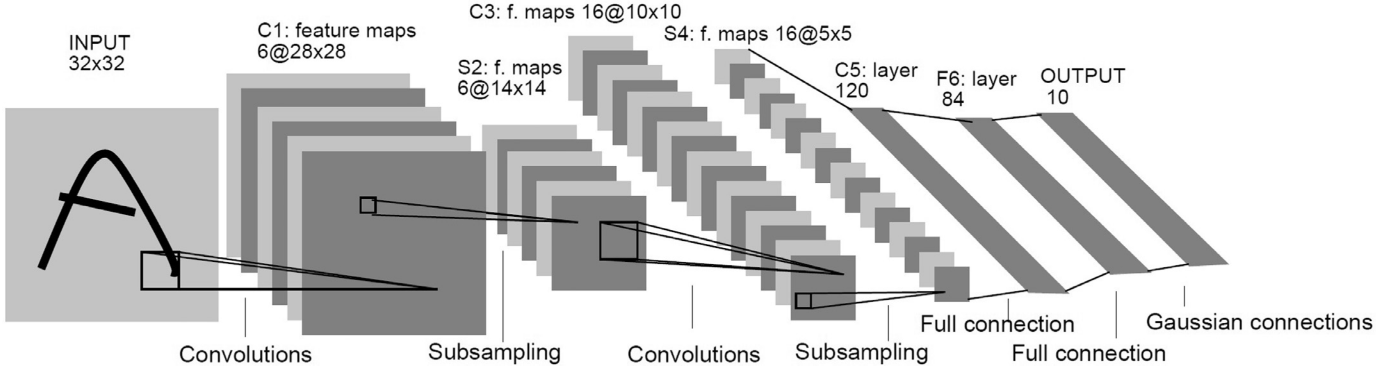

**Fig. 1.** Architecture of LeNet-5. Source (LeCun, Bengio, & Haffner, 1998).

cal subsampling. Thus, CNN is composed of convolutional and subsampling layers alternatively (LeCun et al., 1990).

The main characteristics of CNN architecture are sparse connectivity, convolutional layers and max-pooling. The spatial dependence of the pixels on the image is exploited by local connectivity on neurons on adjacent layers. Max-pooling partitions divide the receptive fields into a set of non-overlapping rectangles and output the maximum value (Bengio, 2009).

Features maps are obtained by convolution operations across the image. The object used as input, in this case, an image is convolved with three filters and biases, which could be trained, as in Fig. 1, to form three feature maps at the C1 level.

Every set of four pixels in the feature maps are summed, weighted, associated with a bias, and crossed by a sigmoid function to create the three feature maps, described as S2 in Fig. 1. They go through a new round of filtering to form the C3 level. Then, the hierarchy is responsible for developing S4 in a corresponding way S2 was produced.

At last, the combination of all pixel values is rasterized to present a unique vector input to the conventional neural network at the output (LeCun et al., 1990).

Considering the q examples $e = (\mathbf{x}, y_b)$, the function $Y^q = F(x, w)$, where $w$ represents the adjustable weights. We could define a *loss function*:

$$E^q(w) = D(|Y^q - y_b|, F(x, w)) \tag{1}$$

This function $E_q(w)$ measures the distance between the real data and the output produced by CNN. The most used criteria for minimizing the error is the Minimum Mean Squared Error (MSE) with penalties for uncorrected classes:

$$E^q(w) = \frac{1}{q}\sum_{q=1}^{q} y^q(F(x, w)) + \log\left(e^{-j} + \sum_{i} e^{-y^q(F(x,w))}\right) \tag{2}$$

Where $y^q$ is the output of q-th layer.

### 2.2. Artificial neural network

The ANN architecture MLP typically consists of a specification of the number of layers, one type of activation function of each unit, and the weights of connections between the different units (Rossi & Carvalho, 2008).

The algorithm used in training the MLP is the error back propagation, and in this work, the pattern was the prototype vector and its label.

This pattern is processed layer by layer until the output layer provides a rendered response, $f_{MLP}$, calculated as shown below:

$$f_{mlp}(x) = \phi\sum_{j=1}^{n} v_j\phi\left(\sum_{i=1}^{m} w_{ij}x_j + b_0\right) + b_1 \tag{3}$$

where $w_{ij}$ are synaptic weights; $b_l$ and $b_0$ are the biases; $\phi$ is the activation function, usually specified as the sigmoid function.

### 2.3. Texture descriptor

The texture is an internal property of almost every natural surfaces such as wood, the weave of a fabric, patterns in sand, leaves, etc. It contains information about the structural arrangement of surfaces and their relationship to the environment.

Nevertheless, the texture is easy to be identified by human eyes, it is hard to be defined in mathematical terms.

Haralick and their colleagues (Haralick, Shanmugam, & Dinstein, 1973) specifically define texture on a more rigorous way, considering it as a set of features extracted from spatial domain for a given probability distribution of grayscale on an image.

There are two possible approaches to texture description: structural and statistical. In both cases, some requirements must be considered as invariance to position, scale and rotation.

The main example of structural is the Fourier transform of the image. The most usual statistical approach is the co-occurrence matrix (Haralick, Shanmugam, & Dinstein, 1973), thanks to its best performance.

### 2.4. Statistical texture descriptors

Suppose a discretized image $I^{m.n} = [i_{x,y}]$ is assumed to be a Gaussian random field, where $i_{x,\,y}$ denotes the gray level of a pixel at location $x, y \in \mathbb{Z}$, with the quantized pixel $i_{x,\,y} < 2^8$ , $i_{x,y} \in \mathbb{N}$.

The co-occurrence matrix contains elements, which counts the pixels with the same brightness, according to certain distance and angle.

For $n$ distinct gray level partitions;

$$b_i = \frac{2^8}{i}, i = 1, 2, \ldots, n \tag{4}$$

There is the Spatial Gray Level Dependence matrix $SGLD = [p_{i,j}]$

$$p_{i,j} = \sum_{1}^{n}\sum_{1}^{n}(i_{x,y} \in b_i).(i_{x,y} \in b_j) \tag{5}$$

and the normalized form: $p_{i,j} = \frac{p_{i,j}}{|p_{i,j}|}$

Considering the averages $u_x$ and $u_y$ and standard deviations $s_x$ and $s_y$:

$$u_x = \sum_{i}\sum_{j} p_{i,j};\, u_y = \sum_{j}\sum p_{i,j} \tag{6}$$

$$s_x^2 = \sum (1-u_x)^2 \sum p_{i,j};\ s_y^2 = \sum (1-u_y)^2 \sum p_{i,j} \tag{7}$$

A feature can be extracted from SGLD through its entropy, energy, max intensity, correlation, and inverse difference moment:

$$entropy = -\sum p_{i,j} \log_2 p_{i,j} \tag{8}$$

$$energy = \sum \sum p_{i,j}^2 \tag{9}$$

$$max = Max(p_{i,j}) \tag{10}$$

$$correlation = \sum \sum \frac{(1-u_x)(1-u_y)p_{i,j}}{s_x s_y} \tag{11}$$

$$IDM = \sum \sum \frac{p_{i,j}}{1+|i-j|} \tag{12}$$

## 3. Related work

The success of DL techniques, and more specifically CNN, is mainly due to their superior predictive performance when compared with other ML techniques. Commonly image processing tasks successfully addressed by CNN are handwritten and image recognition problems (LeCun, Bengio, & Haffner, 1998; LeCun et al., 1990; Neubauer, 1998).

LeCun, Bengio, and Haffner (1998) compared different CNN, with different architectures (e.g., LeNet-1 and LeNet-5), to other classifiers, such as Support Vector Machine (SVM) and K-Nearest Neighbor (KNN), for a handwritten database. In that study, the “Boosted LeNet-4”, an ensemble of three LeNet-4[1], closely followed by LeNet-5 achieved the smaller error rates. However, SVM had excellent accuracy, which is remarkable, according to the author, as it does not include *a priori* knowledge about the problem. On the other hand, SVM required a considerable amount of memory and computational time.

Besides the handwritten problem, Neubauer (1998) compared a variation of convolutional networks, namely Neocognitron (NEO), to other traditional classifiers for face recognition task. The author obtained error rates of 4.5%, 12.9% and 6.4% for NEO, KNN, and MLP, respectively.

According to the author, these positive rates for KNN and NEO decline significantly when the classifiers are tested under more unconstrained conditions, as those related to pose and illumination, However, the background does not present great influence. Other preprocessing steps, such as contour extraction or high pass filtering were not applied since CNN already performs edge detection.

Other researchers focus on problems similar to the one we are addressing on this paper. For instance, a visual automated system using CNN was evaluated by Makantasis, Protopapadakis, Doulamis, Doulamis, and Loupos (2015) to visual tunnel inspection.

Indeed, CNN was employed to hierarchically construct high-level features from low-level ones aiming to describe defects, and a MLP to perform detection task. Unlike, in the present paper, CNN was employed to generate high-level features and performs the visual inspection of timber woods.

Moreover, the authors also compared CNN to existing ML techniques following a conventional paradigm of this area, which consists of using features extracted from images instead of raw pixels.

The textures were used as input of these ML techniques. Due to particular characteristics of wood and tunnel inspection problems, the set of features extracted from images were different. Makantasis, Protopapadakis, Doulamis, Doulamis, and Loupos (2015) compared their proposed approach to MLP, SVM, K-NN and DT. As in the other studies, CNN outperformed the conventional ML techniques by at least 12%.

Gibert, Patel, and Chellappa (2017) used CNN during railway track inspection for defects on crossties and rail fasteners. The cross-ties may be of ten different materials (e.g., wood, plastic, metal, or concrete) and the fastener could be of different types (e.g., elastic clips, bolts, or spikes).

For the material classification task, the CNN predictive performance was evaluated and compared to a fast K-NN algorithm (Muja & Lowe, 2009).

In this case, the input were the features obtained by a Local Binary Pattern (LBP) with two variations. Experimental results showed that the best CNN accuracy 95% against 83% of the best combination of feature extractor and K-NN.

The study of Park, Kwon, Park, and Kang (2016) investigated CNN applied to defect detection in many materials, such as silicon wafer, solid paint, stone, and wood. The main objective of this work was to compare CNN to other techniques, like PSO-ICA, Gabor Filter (Mehrotra, Namuduri, & Ranganathan, 1992) and Random Forest (RF). For the latter, a variance of features was used Kwon, Won, and Kang (2015), while CNN does not need a feature extraction process since it has an embedded module.

Experimental results showed CNN achieved the smallest error rates for five out of seven different materials. The two materials for which CNN was outperformed were wafer and solid paint. For wood, CNN obtained an error rate of 4.21% against 10.73% of Gabor filter, the second best.

In Norlander, Grahn, and Maki (2015), a CNN was used to find wooden knots in images of oak boards. This problem requires a visual inspection system able to deal with knots that can have different aspects, such as size, color, and noise images.

One of the challenges in that study was the limited amount of data to train CNN. Thus, the author initialized CNN using a pre-trained network on a bigger (different) data set, i.e., they used a transfer learning strategy.

CNN experimental results were compared to SVM using HOG (Dalal & Triggs, 2005) features. The former presented a significantly better predictive performance and the transfer learning could improve these results even considering CNN was pre-trained in a different data set domain.

An approach to granite tiles classification using CNN is presented in Ferreira and Giraldi (2017). However, instead of employing CNN for the whole classification process, only the feature extraction method embedded into CNN is used. For such, Ferreira and Giraldi (2017) proposed a methodology where the last CNN layer is removed and the feature vectors are the input.

DL techniques have used for several tasks of classification, in Chong, Han, and Park (2017) offer a systematic analysis of use of DL networks for market analysis and predictions. On the same way unsupervised extractive using auto encoders was used with good results (Yousefi-Azar & Hamey, 2017).

For image classification through texture descriptors, the literature presents the improvement of local binary patterns in texture analysis (Kwak, Xu, & Wood, 2015). In are evaluated feature from images using descriptors (Oliva, Lee, Spolaôr, Coy, & Wu, 2016), proposes a method based on the use of scale invariant features transform (Abdolshah, Teimouri, & Rahmani, 2017) for classifications containers x-ray images. In the table is presented the main contributions in this area.

[1] four first-level feature maps, followed by eight subsampling maps connected in pairs to each first-layer feature maps, then 16 feature maps, followed by 16 subsampling maps, followed by a fully-connected layer with 120 units, followed by the output layer (ten units).

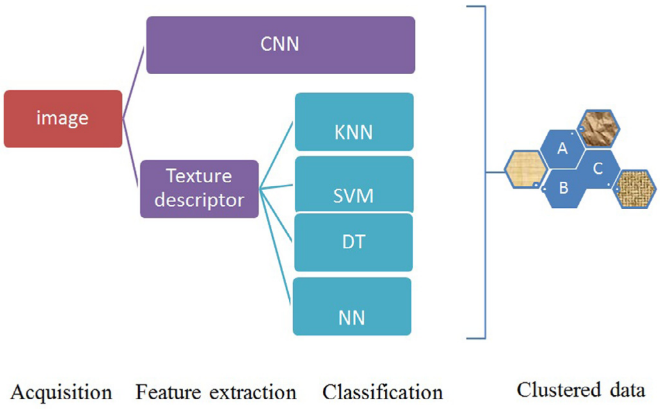

**Fig. 2.** Classification techniques.

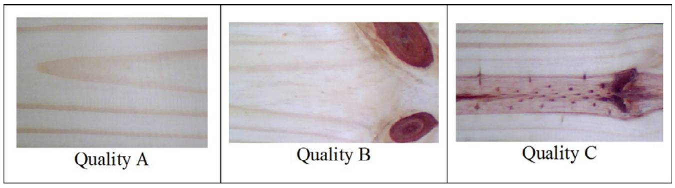

**Fig. 3.** Classification of the wood samples. Source: Own Authors.

## 4. Experiments

In this section, it is presented the experiments carried out to evaluate the image classification performance for CNN and compare them to other ML techniques for the classification of wood board quality. The main difference of CNN is that it is able to extract features from the raw image pixels, while more traditional classification algorithms expected a set of features describing an image, as input. Therefore, in this paper it was used Haralick's texture descriptors to pre-process an image before training a model by different ML techniques, i.e., KNN), ANN, DT and SVM. This strategy is shown on Fig. 2.

As discussed earlier, the hypothesis is that some or all these traditional ML algorithms will generate more accurate models than CNN due to the characteristics of this problem, particularly small number of instances and well-defined domain.

### 4.1. Data

The images used in these experiments were acquired within a Brazilian sawmill company, which commercializes wood as raw material to other industries.

In this company, wood boards are classified by experts as A, B or C, according to their quality. Wood boards classified as A do not present defects and do not need any further process. Wood boards classified as B need extra processes before they are sold. The last class indicates wood with major defects and just small parts will be used. Fig. 3 shows examples of the wood images classified according to this company's experts in A, B and C quality. An amount of 374 images, 640 × 480 pixels each, have been captured from the three different qualities of pine wood boards through a 1.3 Megabyte pixel WebCam.

The resolution of these images was changed to 96 × 27 pixels. From the amount of 374, 144 images were labeled as A (38,5%), 177 were classified as B (47,3%) and 53 as C (14,2%). Therefore, the last one is a class with imbalanced classification problem. The image processing system operates as follow. The features are extracted from color images by treating each channel of color image (Red-Green-Blue) as a monochrome image and transforming their shape information into pixel surfaces.

In the second step, a matrix is created with the numeric values of color intensity corresponding to each pixel (between 0 and 255).

Later, it performs normalization into the numerical matrix, where it is assigned a unit value for pixel maximum intensity and zero to minimal intensity. These features will be the input for the CNN algorithm.

Each example $e = (\mathbf{x}, y_b)$ consists of a tuple of $p$ attributes values $\mathbf{x} = (x_1, \dots, x_p)$, where $p$ is the number of pixels of the image, and a scalar label $y_b \in \{-1, 0, 1\}$, corresponding respectively to qualities A, B and C. The set of all 374 examples constitutes a data set named here as A-B-C.

Another data set is generated from the same features, but, instead of considering classes A, B and C, the classes B and C are aggregated. Therefore, this data set has only two classes and it is named A-BC.

This is done because some companies or experts may be interested in classifying wood on approved boards (class A) and not approved boards (classes B and C).

### 4.2. Feature extraction

On the opposite to CNN, the features extracted directly from the images are not appropriate for traditional ML algorithms, as discussed earlier.

To describe these images properly for these algorithms, texture descriptors can be used.

In this paper, the images are described using Haralick's texture features since they have been widely and successfully employed for this purpose (Leng & Huang, 2012; Nanni, Brahnam, & Lumini, 2012).

For such, a Spatial Gray Level Dependence Matrix (SGLD) from the image and Haralick's texture features are computed from the SGLD.

The inputs to the classification techniques are:

(a) Image: the input gray scale image
(b) Input bits: the gray level resolution of the input image, e.g. for a 256 gray level image input bits=8
(c) Output bits: the gray level resolution of the SGLD matrix, e.g. for output bits=0, the SGLD matrix will be at a resolution of input bits, e.g. for output bits=1, the SGLD matrix will be at a resolution of input bits=-1, etc.
(d) the distance between pixels for calculating the SGLD matrix; it can take values between 0 and the minimum size of image.
(e) Theta: the angle, at which the SGLD matrix is calculated; it can take on values of $0^o$ or $45^o$ or $90^o$ or $135^o$.

Note: a pair of pixels is counted for both the forward and the backward directions for calculating the SGLD matrix.

The outputs are:

(a) SGLD matrix
(b) energy, entropy, maximum probability, correlation, Inverse Difference Moment (IDM)

Therefore, each example using texture descriptor, $e = (\mathbf{x}, y_b)$, consists of the five Haralick's texture features extracted from the images, $\mathbf{x}$={entropy, energy, maximum intensity, correlation, IDM} and the target, $y_b \in \{A, B, C\}$. Just as for CNN, another data set is created using the same features, $\mathbf{x}$, but considering only two classes: A and BC, where BC is the aggregation of classes B and C.

As it can be verified at Table 2, Haralick's descriptor had a good performance to extract features from the image data set and to cluster them on approved parts (class A) and not approved parts

**Table 1**
Relative works summary.

| Author | Application / material | Technique |
|---|---|---|
| LeCun, Bengio, and Haffner (1998) | handwritten-lenet4 | CNN-SVM,KNN |
| Neubauer (1998) | handwritten and face | neocog-KNN,MLP |
| Makantasis, Protopapadakis, Doulamis, Doulamis, and Loupos (2015) | visual tunnel inspection | CNN-SVM,KNN,DT |
| Norlander, Grahn, and Maki (2015) | wooden knot detection | CNN |
| Park, Kwon, Park, and Kang (2016) | material inspection | CNN |
| Oliva, Lee, Spolaôr, Coy, and Wu (2016) | medical images | Haralick |
| Chong, Han, and Park (2017) | market analysis | Deep Learning |
| Yousefi-Azar and Hamey (2017) | email messages | CNN |
| Kwak, Xu, and Wood (2015) | high dimensional space | LBP |
| Gibert, Patel, and Chellappa (2017) | Railway Inspection | LBP and Gabor |
| Ferreira and Giraldi (2017) | Granite Classification | GLCM, HOG and LBP |
| Abdolshah, Teimouri, and Rahmani (2017) | X-rays containers | SIFT |

**Table 2**
Haralick descriptor for test and training data set.

| Class | entropy | energy | intensity | correlation | IDM |
|---|---|---|---|---|---|
| A | 0.51 ± 0.11 | 1.31 ± 0.27 | 0.63 ± 0.12 | 2.30 ± 0.16 | 4.43 ± 1.37 |
| B | 0.48 ± 0.07 | 1.51 ± 0.23 | 0.61 ± 0.09 | 3.19 ± 0.64 | 34.58 ± 28.96 |
| C | 0.38 ± 0.10 | 1.87 ± 0.38 | 0.50 ± 0.12 | 3.22 ± 0.66 | 26.60 ± 21.08 |

**Table 3**
Convolutional neural network parameters and performance.

| Topology | Accuracy | $\kappa$ |
|---|---|---|
| 10-3-3-2-2, 12-3-3-2-2 | 0.6470 ± 0.0979 | 0.5837 ± 0.1175 |
| 10-5-5-2-2, 15-5-5-2-2 | 0.5881 ± 0.1443 | 0.5484 ± 0.1094 |
| 20-3-3-2-2, 25-3-3-2-2 | 0.6339 ± 0.1405 | 0.5665 ± 0.1276 |
| 20-5-5-2-2, 25-5-5-2-2 | 0.5714 ± 0.1344 | 0.5380 ± 0.1079 |
| 20-10-10-4-4, 25-10-10-4-4 | **0.7769** ± 0.0711 | **0.6115** ± 0.1239 |

**Table 4**
CNN Confusion matrix: Classes A, B and C.

| real \ predicted | A | B | C |
|---|---|---|---|
| A | **117** | 27 | 0 |
| B | 17 | **158** | 2 |
| C | 15 | 28 | **10** |

**Table 5**
CNN Confusion matrix: Classes A and BC.

| real \ predicted | A | BC |
|---|---|---|
| A | **99** | 45 |
| BC | 33 | **197** |

(class B and C). This table shows the average value for each characteristic extracted and the standard deviation (STD).

Considering the IDM Texture descriptor, for instance, it is possible to verify that image type-A forms a cluster. However, its descriptor is not enough to classify the images type B and C.

### 4.3. Classification techniques

All the classification techniques employed in this paper were used from Weka (Waikato Environment for knowledge Analysis) software, which is a collection of machine learning algorithms for data mining tasks (Witten & Frank, 2000). Particularly, CNN was performed using a plugin for Weka written by Johannes Amtn.[2] The predictive performance of each algorithm is assessed by the accuracy and Cohen Kappa measures. The former is easier to interpret but does not consider the class imbalance while the latter is the opposite.

#### 4.3.1. Convolutional neural network

Some computational experiments were carried out in order to optimize the CNN topology as shown on Table 3 for the A-B-C data set. The topology is defined by the number of feature maps, patch width, patch height, pool width and pool height, respectively, separated by a hyphen. Besides this parameter, the number of maximum training iterations (epochs) was set to 100. For each topology, the CNN algorithm was performed 10 times, due to the tuning of learning rate process and cost parameters.

Among all parameters evaluated, the best results were found using the topology 20-10-10-4-4, 25-10-10-4-4, which achieved accuracy of 77.69% and $\kappa = 0.6115$. Since CNN was performed 10 times, it produced different confusion matrices. A typical case for this topology is presented on Table 4, which generated an accuracy of 76.20%. It is possible to see that most of the error was mainly due to the misclassified examples between classes A-B and between classes B-C. Thus, differentiating between A-C is easier than the other classes. This result was expected since class A corresponds to non-defective wood boards while class C represent wood boards with major problems.

It is important on classification problems to compare the class tendencies errors, such as presented in Table 4.

Although CNN has been designed to extract features from the raw image pixels, we also tested CNN with the texture descriptor features to support the claim that CNN is not appropriate for this scenario. This experiment resulted in an $accuracy = 47.33$ (± 1.05) and $\kappa = 0.00$ (± 0.00) because CNN generated a random model which predicted all examples as class B. Moreover, we also run CNN using the same parameter values over the binary data set, i.e., considering only classes A and BC, as described in Section 4.1. The confusion matrix for this data set is presented on Table 5, and in this case, the predictive performance was accuracy = 79.07 (± 9.95) and $\kappa = 0.5493$ (± 0.2156). %

#### 4.3.2. Artificial neural network - ANN

The MLP parameter values were set on computational experiments as shown on Table 6. The interval tested and other values were based on expert knowledge of this technique for this domain. The configuration of the number of layers and neurons is accomplished by choosing the architecture that has the lowest permissi-

[2] https://github.com/amten/NeuralNetwork

**Table 6**
MLP parameters and predictive performance.

| $\eta$ | momentum | # nodes | accuracy | $\kappa$ |
|---|---|---|---|---|
| 0.3 | 0.2 | 4 | 80.50 ± 4.27 | 0.6686 ± 0.0695 |
| 0.3 | 0.2 | 8 | **81.26** ± 4.90 | **0.6807** ± 0.0805 |
| 0.3 | 0.4 | 4 | 79.94 ± 3.67 | 0.6612 ± 0.0597 |
| 0.3 | 0.4 | 8 | 81.01 ± 2.85 | 0.6789 ± 0.0512 |
| 0.6 | 0.2 | 4 | 80.20 ± 4.73 | 0.6671 ± 0.0780 |
| 0.6 | 0.2 | 8 | 79.91 ± 3.83 | 0.6626 ± 0.0618 |
| 0.6 | 0.4 | 4 | 81.00 ± 3.56 | 0.6791 ± 0.0547 |
| 0.6 | 0.4 | 8 | 78.85 ± 4.46 | 0.6455 ± 0.0713 |

**Table 7**
MLP confusion matrix: Classes A, B and C.

| real \ predicted | A | B | C |
|---|---|---|---|
| A | **140** | 4 | 0 |
| B | 17 | **150** | 10 |
| C | 5 | 34 | **14** |

**Table 8**
MLP confusion matrix: Classes A and BC.

| real \ predicted | A | BC |
|---|---|---|
| A | **38** | 0 |
| BC | 0 | **56** |

**Table 9**
KNN confusion matrix: Classes A, B and C.

| real \ predicted | A | B | C |
|---|---|---|---|
| A | **139** | 4 | 1 |
| B | 15 | **144** | 18 |
| C | 5 | 24 | **24** |

**Table 10**
KNN confusion matrix: Classes A and BC - accuracy 93,4%.

| real \ predicted | A | BC |
|---|---|---|
| A | **138** | 6 |
| BC | 22 | **208** |

ble error or running time. The generated model achieved an accuracy of 81.26% ( ± 4.90) and $\kappa = 0.6807$ ( ± 0.0805)

It is imperative to compare the class tendencies errors based on confusion matrix, for the case of ANN, as presented on Table 7 for the best results.

The confusion matrix for the 2 class configuration A-BC is presented on Table 8.

The value found for this case was accuracy = 93.56% (± 3.86) and $\kappa$ = 0.8673 ($pm$ 0.0796).

### 4.3.3. K-Nearest Neighbors - KNN

KNN is among the simplest machine learning algorithms. It is used to classify an example based on the classes of the $k$ closest training examples in data set feature space, known as nearest neighbors. Many different measures can be used to compute the distance among the examples. In this paper, we employed the KNN algorithm using the Euclidean distance and $k = 3$. In this case, the value found was; $\kappa$ = 0.703 (± 0.108) and accuracy = 82.11% (± 6.499) according to Table 9. The confusion matrix for the 2 class configuration A-BC is presented on Table 10; the value found was accuracy = 92.50% (± 4.15) and $\kappa$ = 0.8456 (± 0.0846).

**Table 11**
SVM confusion matrix: Classes A, B and C.

| real \ predicted | A | B | C |
|---|---|---|---|
| A | **142** | 2 | 0 |
| B | 18 | **159** | 0 |
| C | 7 | 45 | **1** |

**Table 12**
SVM confusion matrix: Classes A and BC.

| real \ predicted | A | BC |
|---|---|---|
| A | **38** | 0 |
| BC | 1 | **55** |

**Table 13**
DT confusion matrix: Classes A, B and C.

| real \ predicted | A | B | C |
|---|---|---|---|
| A | **131** | 10 | 3 |
| B | 13 | **148** | 16 |
| C | 4 | 24 | **25** |

### 4.3.4. Support vector machines - SVM

SVM is a learning technique based on Statistical Learning Theory (Vapnik, 1995). This technique is robust to high-dimensional data and has been shown high generalization ability in different domains (Statnikov, Wang, & Aliferis, 2008). The principle of SVMs in classification problems is to find an optimum hyperplane that satisfactory split the input data. The optimum hyperplane is defined such that the separation margin among classes is maximized (Haykin, 1999). The support vectors used by SVMs are examples that fit the decision surface. Therefore, these examples are more difficult to classify and directly influence the decision boundary (Haykin, 1999). After a running time of 0.01 second, the SVM model mapped the example space and returns a classification, which achieved $\kappa = 0.660$ (± 0.051) and accuracy = 80.73% (± 3.03), according to Table 11. The confusion matrix for the 2 class configuration A-BC is presented on Table 12, where accuracy = 92.49% (± 3.18) and $\kappa$ = 0.8455 (± 0.0653).

### 4.3.5. Decision tree induction algorithm - DT

A DT is a classifier expressed as a recursive partition of the example space and has been the most widely used approach to represent classification models, mainly due to its comprehensible nature. The structure of a decision tree consists of nodes that have exactly one incoming edge, except the root node, which is at the top of the tree. A node with outgoing edges is called an internal or test node. Each internal node splits the instance space into two or more sub-spaces according to an attribute. All other nodes are called leaves. Each leaf represents the most appropriate class for a subset of examples. New examples are classified starting at the root and go down to a leaf node, according to the outcome of the tests along the path (Quinlan, 1986; Rokach & Maimon, 2005). The J48 algorithm available in Weka was employed in this paper to generate the tree model. This is a version of the C4.5 decision tree algorithm proposed by (Quinlan, 1993). This model achieved $\kappa$ = 0.688 (± 0.093) and accuracy = 81.28% (± 5.506), according to Table 13.With only 2 classes, J48 constructs a model, as showed on Table 14, where accuracy = 93.31% (± 2.75) and $\kappa$ = 0.8598 (± 0.0555).

## 5. Results

So far, we have compared two strategies, i.e., one using texture descriptor as feature extractor, associated with KNN, NN, SVM and

**Table 14**
DT confusion matrix: Classes A and BC.

| real \ predicted | A | BC |
|---|---|---|
| A | **36** | 2 |
| BC | 0 | **56** |

**Table 15**
data set: A-B-C average accuracy by classification process.

| model | Accuracy (%) | $\kappa$ |
|---|---|---|
| CNN | 77.69 ( ± 7.11) | 0.611 ( ± 0.124) |
| MLP | 81.26 ( ± 4.90) | 0.681 ( ± 0.081) |
| KNN | **82.11** ( ± 6.17) | **0.703** ( ± 0.102) |
| SVM | 80.73 ( ± 3.03) | 0.660 ( ± 0.051) |
| DT | 81.28 ( ± 5.51) | 0.688 ( ± 0.093) |

**Table 16**
data set: A-BC average accuracy by classification process.

| model | $\kappa$ | accuracy % |
|---|---|---|
| CNN | 79.07 ( ± 9.95) | 0.549 ( ± 0.216) |
| MLP | **93.56** ( ± 3.86) | 0.867 ( ± 0.080) |
| KNN | 92.50 ( ± 4.15) | 0.846 ( ± 0.085) |
| SVM | 92.49 ( ± 3.18) | 0.845 ( ± 0.065) |
| DT | 93.31 ( ± 2.75) | 0.860 ( ± 0.056) |

DT as classification techniques; and other directly with CNN without taking any pre-processing. In order to provide a more comprehensive comparison between these strategies, Tables 15 and 16 exhibits our main results. Each box summarizes the accuracy values from each classification technique for the test set.

Considering the data summarized at Tables 15 and 16, it is possible to realize that the accuracy presented by CNN is the lowest among other ML techniques.

## 6. Conclusion

In this paper, we extended and performed a comprehensive evaluation of a Convolutional Neural Network (CNN) compared to texture descriptors to classify wood board samples. To evaluate the predictive performance of this technique, experiments were carried out using a data set taken from a real-world problem. The industries are not always interested in the most accurate technique for a given problem, but the most appropriate for the expected results. In other words, there must be a balance between accuracy and computational cost, beneficial to the process efficiency.

Empirical results for the image data set have shown that the texture descriptor method proposed, regardless of the strategy employed, is very competitive when compared with CNN for all performed experiments.

These results point out that ML techniques, associated with texture descriptors, were able to improve the general performance of the classification systems for the problems under analysis.

The best performance of the Texture Descriptor (TD) method could be caused by the nature of the image data set, once it was captured from two-dimensional well behaved images, such as wood board samples. In cases of higher dimensionality, as face or handwritten recognition system, there are plenty examples on literature that point out CNN as the best performance solution.

Variations of DL techniques that gives more attention to the extraction of texture-based features is another promising direction for this niche of image classification tasks.

Once the proposed method achieve good performance when applied to wood board images, the same procedure also should be expanded to other image data sets.

Many important issues that are critical to practical applications of image classifications on industrial environment remain to be explored, for example, the task of labeling data for learning process is daunting and tedious, requiring sometimes millions of labels to achieve the reasonable results. Active learning can often significantly decrease the effort of humans operators by carefully selecting which instances from the unlabeled dataset should be labeled.

In this context we propose for future works to creating industrial visual inspection methodology for materials with appearance change between batches, and having variations that for an human are hard to classify in a consistent manner.

## Acknowledgement

The authors would like to thank the financial support of UNESP and CeMEAI FAPESP, Proc. 13/07375-0 and 16/23410-8.

## References

Abdolshah, M., Teimouri, M., & Rahmani, R. (2017). Classification of x-ray images of shipping containers. *Expert Systems with Applications, 77*, 57–65. doi:10.1016/j.eswa.2017.01.030.

Affonso, C., Sassi, R. J., & Barreiros, R. M. (2015). Biological image classification using rough-fuzzy artificial neural network. *Expert Systems with Applications, 42*(24), 9482–9488. doi:10.1016/j.eswa.2015.07.075.

Al-Allaf, O. N. A., Abdalkader, S. A., & Tamimi, A. A. (2013). Pattern recognition neural network for improving the performance of iris recognition system. *Journal of Scientific and Engineering Research*, 661–667.

Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and Trendsë in Machine Learning, 2*(1), 1–127. doi:10.1561/2200000006.

Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., et al. (2007). Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems, 19*, 153.

Chen, X., Xiang, S., Liu, C.-L., & Pan, C.-H. (2014). Vehicle detection in satellite images by hybrid deep convolutional neural networks. *Geoscience and Remote Sensing Letters, IEEE, 11*(10), 1797–1801. doi:10.1109/LGRS.2014.2309695.

Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications, 83*, 187–205. doi:10.1016/j.eswa.2017.04.030.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (cvpr'05): 1* (pp. 886–893 vol. 1). doi:10.1109/CVPR.2005.177.

Erhan, D., Manzagol, P.-A., Bengio, Y., Bengio, S., & Vincent, P. (2009). The difficulty of training deep architectures and the effect of unsupervised pre-training. In *Twelfth international conference on artificial intelligence and statistics (aistats)* (pp. 153–160).

Ferreira, A., & Giraldi, G. (2017). Convolutional neural network approaches to granite tiles classification. *Expert Systems with Applications In press*. doi:10.1016/j.eswa.2017.04.053.

Gibert, X., Patel, V. M., & Chellappa, R. (2017). Deep multitask learning for railway track inspection. *IEEE Transactions on Intelligent Transportation Systems, 18*(1), 153–164. doi:10.1109/TITS.2016.2568758.

Gonzaga, A., de Franca, C. A., & Frere, A. F. (1999). *Wood texture classification by fuzzy neural networks*. doi:10.1117/12.341113.

Haralick, R., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on, SMC-3*(6), 610–621. doi:10.1109/TSMC.1973.4309314.

Haykin, S. (1999). *Neural networks: A Comprehensive foundation*. Prentice Hall.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., … Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE, 29*(6), 82–97.

Hossain, S., & Serikawa, S. (2013). Texture databases: A comprehensive survey. *Pattern Recognition Letters, 34*(15), 2007–2022. Smart Approaches for Human Action Recognition.

Hubel, D., & Wiesel, T. (1962). Recptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology, 160*, 255–258.

Kim, S., Choi, Y., & Lee, M. (2015). Deep learning with support vector data description. *Neurocomputing, 165*, 111–117. doi:10.1016/j.neucom.2014.09.086.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Kwak, J. T., Xu, S., & Wood, B. J. (2015). Efficient data mining for local binary pattern in texture image analysis. *Expert Systems with Applications, 42*(9), 4529–4539. doi:10.1016/j.eswa.2015.01.055.

Kwon, B.-K., Won, J.-S., & Kang, D.-J. (2015). Fast defect detection for various types of surfaces using random forest with vov features. *International Jour-*

*nal of Precision Engineering and Manufacturing, 16*(5), 965–970. doi:10.1007/s12541-015-0125-y.

LeCun, Y., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *PIEEE*, 2278–2324.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howar, R., Hubbard, W., & Jackel, L. (1990). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems, 160*, 396–404.

Leng, J., & Huang, Z. (2012). On analysis of circle moments and texture features for cartridge images recognition. *Expert Systems with Applications, 39*(2), 2092–2101. doi:10.1016/j.eswa.2011.08.003.

Makantasis, K., Protopapadakis, E., Doulamis, A., Doulamis, N., & Loupos, C. (2015). Deep convolutional neural networks for efficient vision based tunnel inspection. In *2015 IEEE international conference on intelligent computer communication and processing (ICCP)* (pp. 335–342). doi:10.1109/ICCP.2015.7312681.

Mehrotra, R., Namuduri, K., & Ranganathan, N. (1992). Gabor filter-based edge detection. *Pattern Recognition, 25*(12), 1479–1494. doi:10.1016/0031-3203(92)90121-X.

Muja, M., & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *International conference on computer vision theory and application VISSAPP'09)* (pp. 331–340). INSTICC Press.

Nanni, L., Brahnam, S., & Lumini, A. (2012). Random interest regions for object recognition based on texture descriptors and bag of features. *Expert Systems with Applications, 39*(1), 973–977. doi:10.1016/j.eswa.2011.07.097.

Neubauer, C. (1998). Evaluation of convolutional neural networks for visual recognition. *Transaction on Neural Networks, 9*, 685–695.

Norlander, R., Grahn, J., & Maki, A. (2015). Wooden knot detection using convnet transfer learning. In R. R. Paulsen, & K. S. Pedersen (Eds.), *Image analysis: 19th scandinavian conference, SCIA. proceedings* (pp. 263–274). Springer International Publishing. doi:10.1007/978-3-319-19665-7_22.

Oliva, J. T., Lee, H. D., Spolaôr, N., Coy, C. S. R., & Wu, F. C. (2016). Prototype system for feature extraction, classification and study of medical images. *Expert Systems with Applications, 63*, 267–283. doi:10.1016/j.eswa.2016.07.008.

Park, J.-K., Kwon, B.-K., Park, J.-H., & Kang, D.-J. (2016). Machine learning-based imaging system for surface defect inspection. *International Journal of Precision Engineering and Manufacturing-Green Technology, 3*(3), 303–310. doi:10.1007/s40684-016-0039-x.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106. doi:10.1023/A:1022643204877.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Rokach, L., & Maimon, O. (2005). Top-down induction of decision trees classifiers - a survey. *IEEE Transactions on Systems, Man, and Cybernetics Part C, 35*(4), 476–487. doi:10.1109/TSMCC.2004.843247.

Rossi, A. L. D., & Carvalho, A. C. P. L. F. (2008). Bio-inspired optimization techniques for svm parameter tuning. In *Proceedings of 10th brazilian symposium on neural networks* (pp. 435–440). IEEE Computer Society.

Statnikov, A., Wang, L., & Aliferis, C. (2008). A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinformatics, 9*(1), 1–10. doi:10.1186/1471-2105-9-319.

Unser, M. (1995). Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing, 4*(11), 1549–1560. doi:10.1109/83.469936.

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer-Verlag.

Witten, I., & Frank, E. (2000). *Data mining: Practical machine learning tools and techniques with java implementations*. San Francisco: Morgan Kaufmann.

Yousefi-Azar, M., & Hamey, L. (2017). Text summarization using unsupervised deep learning. *Expert Systems with Applications, 68*, 93–105. doi:10.1016/j.eswa.2016.10.017.