Physica A 635 (2024) 129518

Contents lists available at ScienceDirect

Physica A

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

Color-texture classification based on spatio-spectral complex network representations

Lucas C. Ribas [a,*], Leonardo F.S. Scabini [b], Rayner H.M. Condori [c], Odemir M. Bruno [b]

[a] Institute of Biosciences, Humanities and Exact Sciences, São Paulo State University, Rua Cristóvão Colombo, 2265, 15054-000, São José do Rio Preto, SP, Brazil
[b] São Carlos Institute of Physics, University of São Paulo, São Carlos, SP, Brazil
[c] Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, SP, Brazil

ARTICLE INFO

Keywords:
Color-texture
Neural network
Complex network

ABSTRACT

This paper proposes a method for color-texture analysis by learning spatio-spectral representations from a complex network framework using the Randomized Neural Network (RNN). We model the color-texture image as a directed complex network based on the Spatio-Spectral Network (SSN) model, which considers within-channel connections in its topology to represent the spatial characteristics and spectral patterns covered by between-channel links. The insight behind the method is that complex topological features from the SSN can be embedded by a simple and fast neural network model for color-texture classification. Thus, we investigate how to effectively use the RNN to analyze and represent the spatial and spectral patterns from the SSN. We use the SSN vertex measurements to train the RNN to predict the dynamics of the complex network evolution and adopt the learned weights of the output layer as descriptors. Classification experiments in four datasets show the proposed method produces a very discriminative representation. The results demonstrate that our method obtains accuracies higher than several literature techniques, including deep convolutional neural networks. The proposed method also showed to be promising for plant species recognition, achieving high accuracies in this task. This performance indicates that the proposed approach can be employed successfully in computer vision applications.

1. Introduction

Texture is a key property present in nature, object surfaces, among others. It is an important feature used to distinguish a wide range of things such as animals, plants, materials, and objects. Thus, most computer vision and image processing techniques employ texture features in pattern recognition tasks. In the literature, there are a lot of texture analysis techniques that were developed for gray-level images. However, nowadays, color images are predominantly captured by most imaging devices and are present in many datasets and applications.

Different approaches for color-texture analysis have been introduced aiming to characterize the texture in colored images [1,2]. Most of them are extensions that employ the gray-level texture techniques in the image channels separately (integrative methods) or transform the color image to gray-level. Many gray-level texture analysis methods are based on local patterns with improvements for noise robustness and rotation invariance, such as LETRIST [3] and LGONBP [4]. On the other hand, pure color methods represent a different approach to texture analysis, focusing exclusively on the colors in an image without taking into account the pixel

* Corresponding author.
E-mail address: lucas.ribas@unesp.br (L.C. Ribas).

https://doi.org/10.1016/j.physa.2024.129518
Received 1 August 2023; Received in revised form 19 December 2023
Available online 12 January 2024

spatial relationship. Thus, a common strategy involves integrating both pure color and gray-level features in parallel methods. Moreover, recently, deep convolutional neural network models have been employed in the characterization of color images for different tasks [5]. However, many of these approaches do not fully address color information, such as spatial patterns between the multispectral pixels. Other recent methods inspired by local patterns employ spatially weighted order binary pattern (SWOBP) [6], aiming to encode color order information between channels and color order relationships in the spatial domain. Song et al. [7] propose a method called quaternionic extended local binary pattern with adaptive structural pyramid pooling, which is robust to image rotation and spatial changes.

In recent years, researchers have successfully applied complex network (CN) concepts to image analysis, especially texture analysis. The field of complex networks combines graph theory, physics, and statistics, aiming to analyze large graphs derived from complex systems and natural phenomena. One of the main advantages of CN-based methods is the flexibility and ability to represent the intrinsic patterns of the data. CN-based methods for texture analysis work in two steps: modeling the image into a complex network, followed by structural analysis of the resulting network. In the modeling step, several works studied how to represent and quantify texture patterns in undirected [8,9] and directed weighted graphs [10]. To characterize color textures, some studies have proposed modeling spatial and spectral information through multilayer networks [2] and spatial–spectral networks [11]. Despite the promising results, these techniques only employ statistical measures for structural analysis of the modeled CN and obtaining the image descriptors.

This paper proposes a method for analyzing color-texture images. It is based on a directed graph modeling from the Spatio-Spectral Network (SSN) framework as in [11] and a Spatio-Spectral Representation (SSR) learned using the Randomized Neural Network (RNN) architecture. The color-texture images are modeled as directed SSNs, which define the pixels for all channels as vertices and create connections between the same and different channels employing a radially symmetric neighborhood. The resulting SSN provides spatial information through the symmetric neighborhood, and the between-channel connections represent the spectral characteristics. The RNN is trained from the vertex metrics (degree and strength) to predict the network evolution, and we use the learned weights of the output layer as a feature vector, called Spatio-Spectral representation. The proposed approach takes into account a handcrafted modeling (SSN) of the image combined with a feature learning part from the RNN for characterization. Thus, our contribution is a discriminative characterization of the complex network from the representation (SSR) learned by the RNN, which performs an effective description of spatial and spectral properties of the color image. We perform classification experiments in various color-texture datasets and compare the results with several literature techniques. In addition, we apply our method in the challenging task of plant species recognition based on the texture of plant leaves.

This paper is divided as follows. In Sections 2 and 3, we describe the methodologies used by the proposed method. Section 4 introduces the proposed method for color-texture analysis, while Section 5 details the experiments, parameter evaluation, and discusses the results and comparisons. Finally, Section 6 concludes the work.

2. Randomized neural network

Randomized neural network [12–15] has been widely used in recent years in different pattern recognition tasks [10,16–18] due to key characteristics such as simplicity, high accuracy performance, and a fast learning algorithm based on the least-squares solution. This neural network architecture comprises a feed-forward hidden layer with weights given randomly based on a uniform or Gaussian distribution and output weights learned using the least-square solution. For description of this architecture, consider an input feature matrix $X = [\vec{x_1}, \vec{x_2}, \dots, \vec{x_N}]$ composed of N input vectors (with a constant -1 to connect to the bias weights) and its respective labels $D = [d_1, d_2, \dots, d_N]$. The weights of the hidden neurons are represented by a matrix W of $Q \times (p+1)$, where Q is the size of the hidden layer and $(p+1)$ are the attributes with the first column as the bias weights.

To get the output Z of the hidden layer, we calculate the activation function $Z = \phi(WX)$, which is generally a sigmoid or hyperbolic tangent function. The output Z is a matrix $Z = [\vec{z_1}, \vec{z_2}, \dots, \vec{z_N}]$ (a constant -1 is added in the vector $\vec{z_i}$ as bias weight) that is used to compute the output neuron weights. In the output layer, the weights are represented by a matrix M (each row refers to a neuron), which can be calculated by the Moore–Penrose pseudo-inverse [19],

$$M = DZ^T(ZZ^T)^{-1}. \tag{1}$$

We also can apply the Tikhonov regularization [20,21] to solve the problem of inaccurate inverse in Eq. (1). This problem occurs because, in many cases, the matrix ZZ^T becomes close to singular. Thus, the Tikhonov regularization is applied according to $M = DZ^T(ZZ^T + \lambda I)^{-1}$, where $\lambda > 0$ is the regularization value and I the identity matrix.

3. Modeling color-texture as Spatio-Spectral Network

We model the color-texture image as a Spatio-Spectral Network (SSN) inspired by the framework presented in our previous work [11]. This approach models the relationship of the intra and inter-channel pixels by means of a directed CN, called SSN. The network represents the relationship of pixels considering both their spatial organization and their variations in the spectral domain (across different color channels). This approach is inspired by the theory of opponent color processing. It was shown that SSNs can capture a wide range of patterns in color textures.

To build SSNs, consider a color image I of $(w * h)$ pixels and z color-channels with pixel values between $[0, L]$. This image is modeled as a CN $N = \{V_N, E_N\}$ formed by a set of vertices V_N and a set of edges E_N. Each CN vertex v_i represents an image pixel i located in a given channel, forming the set $V_N = \{v_1, \dots, v_{w*h*z}\}$. Each vertex has two coordinates (x, y) mapping the position of

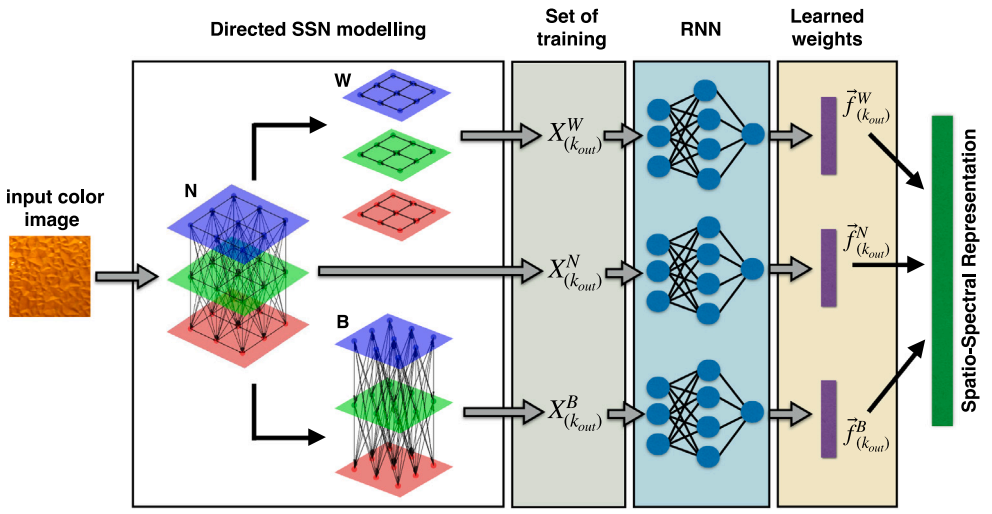


Fig. 1. The overall process of the proposed method (using the out-degree measure) for a color image and the parameters R and Q. Firstly, the color image is represented by a directed SSN modeling, resulting in three CNs: N, W and B. Note that in this step the radius of connection is ranged from $\{r_1, r_2, \ldots, r_R\}$. Then, we create the input feature matrices $X_{(k_{out})}$ (the process is the same for measures s_{out} and s_{in}) for the three CNs based on the set of radiuses and then we train the RNNs to predict the SSN dynamic. The output layer vectors $\vec{f}_{(k_{out})}$ with the learned weights compose the feature vector.

the pixel in the image and the value $p(v_i) \in [0, L]$ of the pixel intensity in its respective channel. Given these definitions, we have a multilayer complex network in which each layer represents a color channel.

To create connections, as in [11], we use a set of radius $\{r_1, \ldots, r_R\}$ that defines the vertex neighborhood using radially symmetrical connection criteria such as,

$$G^r_{v_i} = \{v_j \in V_N | r_{i-1} < d(v_i, v_j) \leq r_i\}, \tag{2}$$

where $d(v_i, v_j)$ is the 2-D Euclidean distance between the vertices based on its coordinates. That is, two vertices v_i and v_j are connected if the distance $d(v_i, v_j)$ is in an interval between the current r_i and the previous r_{i-1} radius, thus $\{G^{r_1}_{v_i}, \ldots, G^{r_{i-1}}_{v_i}\} \nsubseteq G^{r_i}_{v_i}$. For $G^1_{v_i}$ the pixels are connected to itself in different channels, i.e., when the distance is 0. Each edge has a weight computed using a definition that balances the color difference and distance between the pixels [2]: $a(v_i, v_j) = \frac{(|p(v_i)-p(v_j)|+1)(d(v_i,v_j)+1)-1}{(L+1)(r+1)-1}$.

Note that these definitions of neighborhood and weight perform the processing of opponent colors in a spatial manner [11]. The edges are directed through the gradient direction, i.e., the vertex with low color intensity points to the vertex with high color intensity. Thus, the unique modeling parameter is the set of radius $\{r_1, \ldots, r_R\}$ and the set of edges is defined by:

$$E_N = \left\{a(v_i, v_j) \in E \mid r_{i-1} < d(v_i, v_j) \leqslant r_i \wedge p(v_i) < p(v_j)\right\}. \tag{3}$$

The edge is bidirectional if the vertices have the same color intensity. The result of this modeling is a CN with topological properties that represent the color-texture patterns, which can be analyzed and embedded by simple vertex measures [11]. Also, the spatio-spectral properties from CN are extracted from the connection patterns between channels, which are highlighted through CNs derived from the original CN as in [2]. For this purpose, the two derived CNs are: $W^r(V_W, E_W)$ with only the edges between the same channel, such $\forall w(v_i, v_j) \in E_N$, $E_W = a(v_i, v_j) | p(v_i, z) = p(v_j, z)$, where $p(v_i, z)$ returns the channel; $B^r(V_B, E_B)$ that contains a set of edges connecting vertices in different channels, $\forall a(v_i, v_j) \in E_N$, $E_B = a(v_i, v_j) | p(v_i, z) \neq p(v_j, z)$. In this way, we have three CNs N, W and B whose sets of vertices ($V_N = V_W = V_B$) are the same, but with different sets of edges. Fig. 1 illustrates in the first step the modeling of a color-texture image in the three CNs.

4. Spatio-Spectral Representation learning

The proposed Spatio-Spectral Representations (SSRs) are formed by texture features learned by RNNs using the centrality measures computed from the modeled graph (SSN), which are the out-degree, out-strength, and in-strength. These are simple measures that can be quickly calculated in the graph modeling step since it uses only the neighboring vertices. We use only the out-degree because the in-degree and out-degree are a linear transformation in function of the maximum radius, that is, $\forall v_i \in V$, $k(v_i)_{in} + k(v_i)_{out} = k^r_{max}$, where k^r_{max} is the maximum number of connection for a radius r. On the other hand, as the vertex strength is calculated based on the distribution of the pixel intensities, there is no correlation between the in-strength and out-strength. Thus, the measures used for vertex representation are: out-degree, in-strength and out-strength. The out-degree $k(v_i)_{out}$ of the vertex v_i is the number of outgoing connections from v_i:

$$k(v_i)_{out} = \sum_{\forall v_j \in V} \begin{cases} 1, & a(v_i, v_j) \in E \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

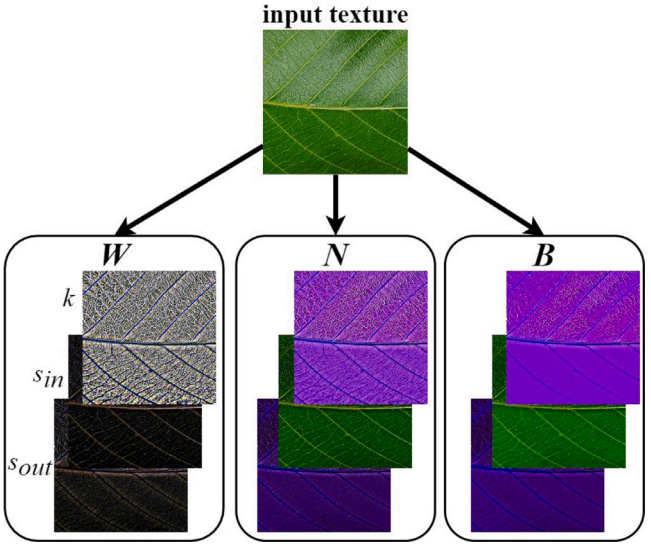


Fig. 2. Visual representation of SSN ($R = 8$); resulting images are obtained by computing CN measures for a vertex on each network layer, then normalizing it to 8-bit RGB pixels.

The out-strength $s(v_i)_{out}$ and in-strength $s(v_i)_{in}$ represent the sums of the weights of the output and input connections of v_i, respectively:

$$s(v_i)_{\text{out}} = \sum_{\forall v_j \in V} \begin{cases} a(v_i, v_j), & a(v_i, v_j) \in E \\ 0, & \text{otherwise} \end{cases} \quad s(v_i)_{\text{in}} = \sum_{\forall v_j \in V} \begin{cases} a(v_j, v_i), & a(v_j, v_i) \in E \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The CN measures from SSNs comprise important information regarding the texture it represents. In Fig. 2 we show the results of reconstructing the original RGB image from each CN measure (out-degree, in-strength and out-strength) for CNs N, W and B. It is possible to notice how each measure and CN highlights different visual properties of the input. When combined, they form an enriched representation that the RNN can learn for texture characterization.

To take advantage of the SSN information, we concatenate all vertex measures into feature matrices used to train the randomized neural network. These matrices are constructed using a strategy of graph evolution for different values of modeling parameter r, as illustrated in Fig. 3. In this sense, for a vertex v_i, the out-degrees for different values of radius $r = \{1, 2, \dots, (R-1)\}$ form the input vector $\vec{x}_{v_i} = [k(v_i)^1_{out}, k(v_i)^2_{out}, \dots, k(v_i)^{R-1}_{out}]$, which will be used to predict the out-degree of v_i for the maximum radius R, with its label $d_i = k(v_i)^R_{out}$. This process is performed for all vertices of the CN in the original version (called N) to obtain the input feature matrix $X^N_{(k_{out})}$ for the out-degree and the label vector D (see Fig. 3(b)). Thus, we can analyze the topological evolution of the vertices (representing pixels with a given intensity) as we increase their neighborhood, reaching larger regions and performing a multiscale analysis. Similarly, we create input feature matrices with the out-strength $X^N_{(s_{out})}$ and the in-strength $X^N_{(s_{in})}$. In addition, input feature matrices were constructed using measures computed from the CN of within-channel connections W, and the CN with between-channel connections B, resulting in the following input matrices: $X^W_{(k_{out})}$, $X^W_{(s_{out})}$, $X^W_{(s_{in})}$, $X^B_{(k_{out})}$, $X^B_{(s_{out})}$, $X^B_{(s_{in})}$. It is important to stress that these feature matrices are computed from a single texture image sample.

The next step to train the neural network is to define the weight matrix of the input layer. In this case, the values of the weight matrix directly influence the training result of the neural network and, consequently, on the feature vector to represent the image [10]. Thus, we need to employ the same values in the weight matrix to ensure that a unique feature vector will always be generated for the same image. For this purpose, the linear congruent generator (LCG) [22] is used to obtain the pseudo-random numbers of the matrix, according to $V(n+1) = (a * V(n)+b) \bmod c$, whose the sequence V has length $E = Q * (p+1)$ and $V(1) = E+1$. The parameter values a, b and c are defined as $a = E + 2$, $b = E + 3$ and $c = E^2$. The weight matrix is obtained by dividing the sequence V into Q segments of length $p + 1$. This matrix and the input matrices X are normalized with z-score.

In our proposal, the feature vectors that form the spatio-spectral representations to analyze the color-texture image are created using the matrix M, which is defined as the vector $\vec{f} = DZ^T(ZZ^T + \lambda I)^{-1}$, where $\lambda = 10^{-3}$ and its length is $Q + 1$ due to the bias value (Fig. 1(c)–(d)). Thus, nine randomized neural networks are trained with the following input feature matrices $X^N_{(k_{out})}$, $X^N_{(s_{out})}$, $X^N_{(s_{in})}$, $X^W_{(k_{out})}$, $X^W_{(s_{out})}$, $X^W_{(s_{in})}$, $X^B_{(k_{out})}$, $X^B_{(s_{out})}$, $X^B_{(s_{in})}$, producing the vectors $\vec{f}$ that are combined to obtain the first feature vector,

$$\vec{Y}(R)_Q = \left[\vec{f}^N_{(k_{out})}, \vec{f}^N_{(s_{out})}, \vec{f}^N_{(s_{in})}, \vec{f}^W_{(k_{out})}, \vec{f}^W_{(s_{out})}, \vec{f}^W_{(s_{in})}, \vec{f}^B_{(k_{out})}, \vec{f}^B_{(s_{out})}, \vec{f}^B_{(s_{in})}\right], \tag{6}$$

where R is the number of used radius. Note that each input feature matrix generates a vector $\vec{f}$. The vector $\vec{Y}(R)_Q$ is produced with a unique value of Q and R. These two parameters influence the learning process of the neural network and, consequently, the

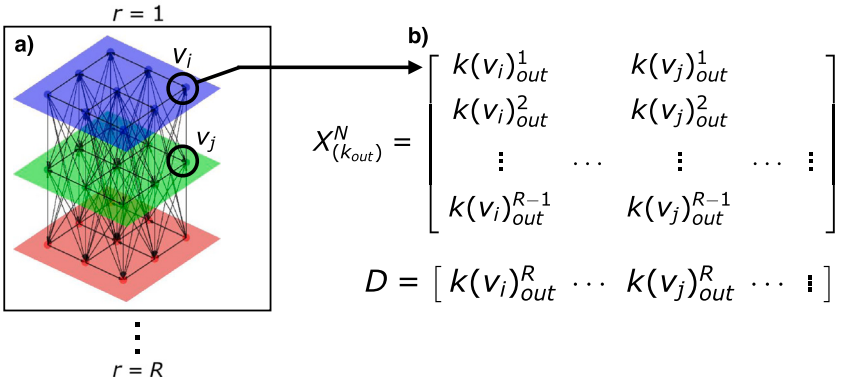


Fig. 3. Illustration to create the input feature matrix $X^N_{(k_{out})}$ with the out-degree from the original version N. The color-texture image is modeled in graphs with different values of r (the figure shows an example for $r = 1$). The out-degree $k(v_i)_{out}$ of the vertices for different values of r are used to create the input matrix $X^N_{(k_{out})}$ and their color intensities are the labels.

learned features. In this context, we propose the representation $\vec{\Theta}(Q)_{(R_1,R_2)}$ that combines the vectors $\vec{Y}(R)_Q$ for two values of R:

$$\vec{\Theta}(Q)_{(R_1,R_2)} = \left[\vec{Y}(R_1)_Q, \vec{Y}(R_2)_Q\right]. \tag{7}$$

In the same way, we propose the final spatio-spectral representation $\vec{\Psi}(R_1, R_2)_{Q_1,Q_2,\ldots,Q_m}$ combining the vectors $\vec{\Theta}(Q)_{(R_1,R_2)}$ with different values of Q,

$$\vec{\Psi}(R_1, R_2)_{Q_1,Q_2,\ldots,Q_m} = \left[\vec{\Theta}(Q_1)_{(R_1,R_2)}, \vec{\Theta}(Q_2)_{(R_1,R_2)}, \ldots, \vec{\Theta}(Q_m)_{(R_1,R_2)}\right]. \tag{8}$$

Fig. 1 presents the overall process of the proposed methodology for a color image and using a single parameter value of Q and R. In this illustration, we show the process using the out-degree matrix $X_{(k_{out})}$, however, the procedure is the same for the other topological measures.

5. Experiments and results

To evaluate the methods, the following datasets are used in the experiments:

- Outex_00013 [23]: this dataset contains 1360 samples divided into 68 classes with 20 images of 128 × 128 pixels per class.
- USPTex [24]: in this dataset, the samples are divided into 191 classes with 12 samples (128 × 128) each, totaling 2292 samples.
- Multi-Band Texture (MBT) [25]: the dataset contains colored images that combine effects of intraband and interband spatial variations. In general, this characteristic comes from astronomy and remote sensing fields. The dataset is formed by 2464 images separated into 154 classes, with 16 samples of size 160 × 160 each.
- CUReT [26]: this dataset is formed by colored images of materials (e.g., gypsum, concrete, artificial grass, leather) divided into 61 classes with 92 samples each. It exhibits a diverse variety of photometric and geometric characteristics, including variations in illumination, rotation, and differing viewpoints.

In this work, we employed the Linear Discriminant Analysis (LDA) [27] classifier to evaluate and compare the performance of our method and the others. In the parameter analysis, the 10-fold cross-validation scheme is adopted with 10 repetitions. We also use the leave-one-out cross-validation scheme in comparative analysis. In all experiments, the performance metric used is classification accuracy, defined as the number of images correctly classified.

5.1. Parameter evaluation

To define the parameter values, we employ the proposed representations to analyze the three color-texture datasets using different values of the parameters Q and R. The first learned representations analyzed are $\vec{Y}(R)_Q$ and $\vec{\Theta}(Q)_{(R_1,R_2)}$ for different values of $R = \{04, 06, 08, 10, 12\}$ and its combinations. To compute this representation, we used a fixed value $Q = 4$. The parameter R determines the number of radiuses used in the analysis of the dynamical evolution of the complex network and also defines the radiuses used to create the input feature matrices X to train the neural network. Fig. 4 shows the matrix with the average accuracies achieved in the three datasets for one value of R (main diagonal) and two combined values. Note that two combined values of R obtain the highest accuracies. Such behavior is expected: more combined radius values, larger the size of the Spatio-Spectral Representation (SSR). In this case, the representation $\vec{\Theta}(04)_{(04,10)}$ produces the higher accuracy. That is, the SSNs are analyzed with two sets of radiuses: $r = \{1, 2, 3, 4\}$ e $r = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. In terms of image, the smaller radiuses perform a local analysis

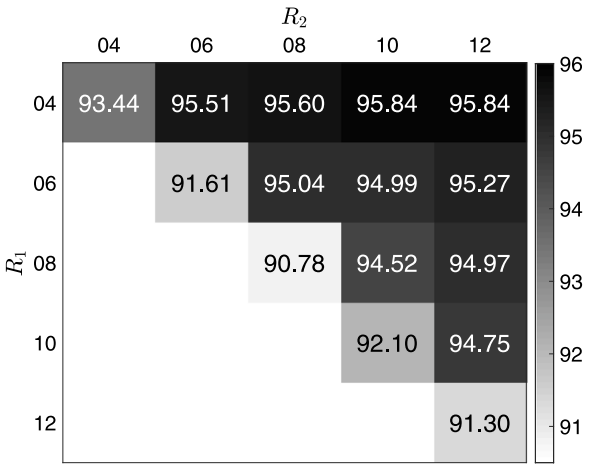


Fig. 4. Accuracies for different values of R (main diagonal) and its combinations with $Q = 04$.

Table 1
Accuracies for the feature vector combining $\{Q_1, Q_2, Q_3\}$ for the radius $R = \{4, 10\}$.

$\{Q_1, Q_2, Q_3\}$	Outex	MBT	USPTex
{04, 09, 14}	96.40 (± 1.49)	98.38 (± 0.77)	99.43 (± 0.36)
{04, 09, 19}	97.13 (± 0.88)	98.17 (± 0.70)	99.43 (± 0.36)
{04, 09, 24}	96.76 (± 1.31)	98.17 (± 0.64)	99.26 (± 0.80)
{04, 14, 19}	96.54 (± 0.85)	97.85 (± 0.66)	99.52 (± 0.38)
{04, 14, 24}	96.62 (± 1.44)	97.93 (± 0.56)	99.39 (± 0.51)
{04, 19, 24}	96.47 (± 1.19)	98.21 (± 0.77)	99.17 (± 0.66)
{09, 14, 19}	96.76 (± 1.05)	97.97 (± 0.51)	99.39 (± 0.37)
{09, 14, 24}	96.54 (± 0.98)	98.25 (± 0.81)	99.26 (± 0.36)
{09, 19, 24}	97.13 (± 1.27)	98.21 (± 0.64)	99.21 (± 0.61)
{14, 19, 24}	97.21 (± 1.33)	96.96 (± 1.13)	99.13 (± 0.41)

of the pixel neighborhood, in turn, as we increase the radius value, more distant regions are accessed and considered in the SSN topology, allowing a multiscale analysis in the image.

A second experiment evaluates the representation $\vec{\Psi}(R_1, R_2)_{Q_1,Q_2,\ldots,Q_m}$ using different values and combinations of Q. In this experiment, we test the following values of $Q = \{04, 09, 14, 19, 24\}$ with a fixed value $R = \{04, 10\}$ that was defined in the previous experiment. Fig. 5 shows the accuracy matrices computed for the three datasets. In this test, the results indicate that two combined values of Q are better than a single value (main diagonal) in all datasets. However, the results using a single value of Q are also very competitive because they produce smaller representations and better computational performance.

We also evaluated the performance of the combination of three values of Q in the three datasets, as can be seen in Table 1. For this case, the results indicate similar performances in various combined values of Q in the three datasets. When considering the trade-off between accuracy and representation dimension in all datasets, the best combination is the $Q = \{04, 09, 19\}$. In this way, the accuracies achieved were 97.13 (± 0.88), 98.17 (± 0.70), and 99.43 (± 0.36) on the Outex, MBT, USPTex datasets, respectively. In addition to this setting, we also can consider the combination $Q = \{09, 19, 24\}$, because it produces high accuracies in the datasets, although the learned representation is larger. Therefore, the proposed Spatio-Spectral Representations of our method are $\vec{\Psi}(04, 10)_{04,09,19}$ (SSR^1) and $\vec{\Psi}(04, 10)_{09,19,24}$ (SSR^2).

Finally, we also carried out a study to understand the impact of the different components of the proposed method on the final representation. Our final representation is formed by the combination of representations $\vec{f}$ learned using three measures (out-degree k_{out}, out-strength s_{out} and the in-strength s_{in}) from three different complex networks (original N, within-channel W and between-channel B). For this purpose, we adopted the two best parameter configurations: SSR^1 and SSR^2.

Table 2 presents the average accuracies obtained on the Outex, USPTex, and MBT datasets for the several components (representations) that form our final representations. When we analyze the components individually, it becomes evident that the representations derived from the complex network with within-channel connections, denoted as W, along with strength measures, provide a better characterization of texture images. This indicates that the spatial information and the variation in pixel colors are significant factors in discriminating texture images. For the combinations involving the three complex networks and three measurements, despite very close results in all combinations, the best performance is achieved with the combination $\left[\vec{f}^{N}_{(s_{in})}, \vec{f}^{B}_{(s_{in})}, \vec{f}^{W}_{(s_{in})}\right]$, which utilizes the in-strength as a topological measure for all three complex networks. These average accuracies are even lower than the final representations, which are composed by concatenation of all components: 98.24% (SSR^1) and 98.18% (SSR^2).

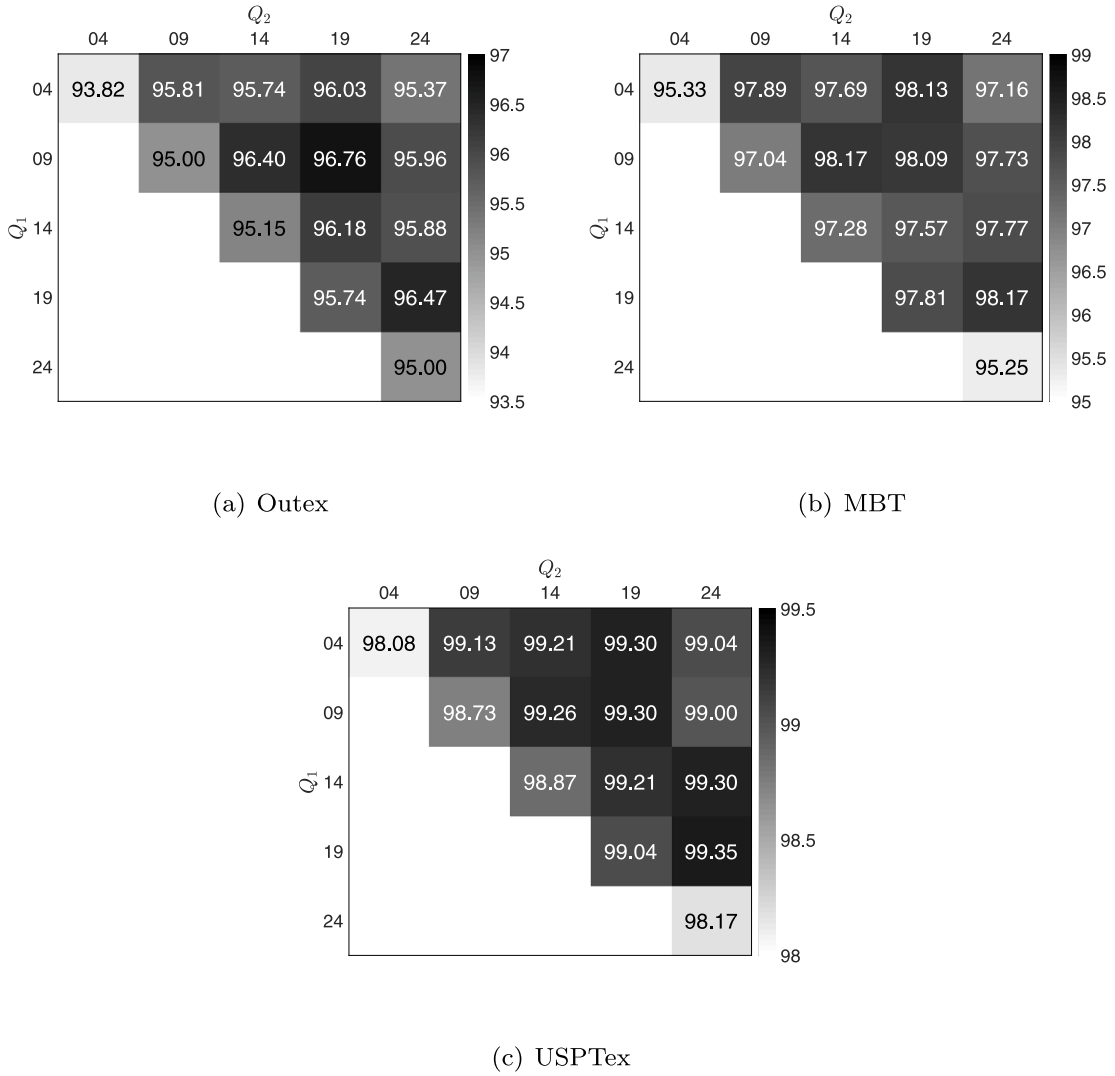


Fig. 5. Accuracies using different combinations of Q_1 and Q_2 for $R = \{04, 10\}$.

5.2. Analysis of noise tolerance and rotation robustness

In this section, we evaluate the ability of our method to deal with noisy and rotated images. The noise tolerance and rotation robustness are properties desirable in problems and real applications involving texture analysis techniques. In the noise tolerance test, we apply salt and pepper noise, which is a common visual noise present in images, in the Outex dataset. To assess our method under different conditions, we added noise to each image from the Outex dataset at densities $d = [0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3]$. The noise density, denoted by d, impacts roughly $d * n$ pixels in an image with $n = (w * h * z)$ pixels. As we can see in Fig. 6, as we increase the d value, more pixels have their value modified, making it difficult to distinguish the texture image. The classification accuracies produced by the proposed representation SSR^1, starting from the lowest to the highest density, were: 96.13% (0.26), 95.02% (0.33), 94.54% (0.27), 93.59% (0.28), 93.79% (0.32), 93.82% (0.38), 92.51% (0.3). On the other hand, the accuracies for the SSR^2 representation were: 95.65% (0.24), 94.79% (0.33), 94.32% (0.32), 93.3% (0.34), 93.01% (0.4), 93.5% (0.33), 92.48% (0.47). As the noise density increases, the accuracy decreases slightly when compared to the original dataset. This behavior is expected since texture image patterns are modified with high noise density rates. Therefore, the results indicated that our method is to be quite tolerant to noise, even at high rates.

In the experiment of rotation robustness, we adopted a rotated version of the Outex dataset according to the procedure in [8]. In this dataset, the original image is rotated in 11 different angles, resulting in 11 samples for each texture class. Examples of texture images rotated in 11 angles are shown in Fig. 7. The image representations obtained by our method were submitted to the LDA classifier using a leave-one-out cross-validation scheme. In this validation approach, an image rotated to a specific angle is

Table 2
Average accuracies (Outex, USPTex and MBT datasets) for the various components that form the two representations (SSR^1 and SSR^2) of the proposed method.

Component	SSR^1	SSR^2
$\vec{f}^{N}_{(k_{out})}$	93.17	92.29
$\vec{f}^{N}_{(s_{out})}$	92.30	91.15
$\vec{f}^{N}_{(s_{in})}$	92.71	91.85
$\vec{f}^{B}_{(k_{out})}$	90.64	89.84
$\vec{f}^{B}_{(s_{out})}$	91.36	90.89
$\vec{f}^{B}_{(s_{in})}$	92.32	91.60
$\vec{f}^{W}_{(k_{out})}$	91.11	90.45
$\vec{f}^{W}_{(s_{out})}$	93.61	93.00
$\vec{f}^{W}_{(s_{in})}$	94.03	92.96
$\left[\vec{f}^{N}_{(k_{out})}, \vec{f}^{N}_{(s_{out})}, \vec{f}^{N}_{(s_{in})}\right]$	96.11	95.51
$\left[\vec{f}^{B}_{(k_{out})}, \vec{f}^{B}_{(s_{out})}, \vec{f}^{B}_{(s_{in})}\right]$	95.51	95.02
$\left[\vec{f}^{W}_{(k_{out})}, \vec{f}^{W}_{(s_{out})}, \vec{f}^{W}_{(s_{in})}\right]$	96.05	95.66
$\left[\vec{f}^{N}_{(k_{out})}, \vec{f}^{B}_{(k_{out})}, \vec{f}^{W}_{(k_{out})}\right]$	96.15	95.89
$\left[\vec{f}^{N}_{(s_{out})}, \vec{f}^{B}_{(s_{out})}, \vec{f}^{W}_{(s_{out})}\right]$	96.59	96.27
$\left[\vec{f}^{N}_{(s_{in})}, \vec{f}^{B}_{(s_{in})}, \vec{f}^{W}_{(s_{in})}\right]$	97.16	96.82

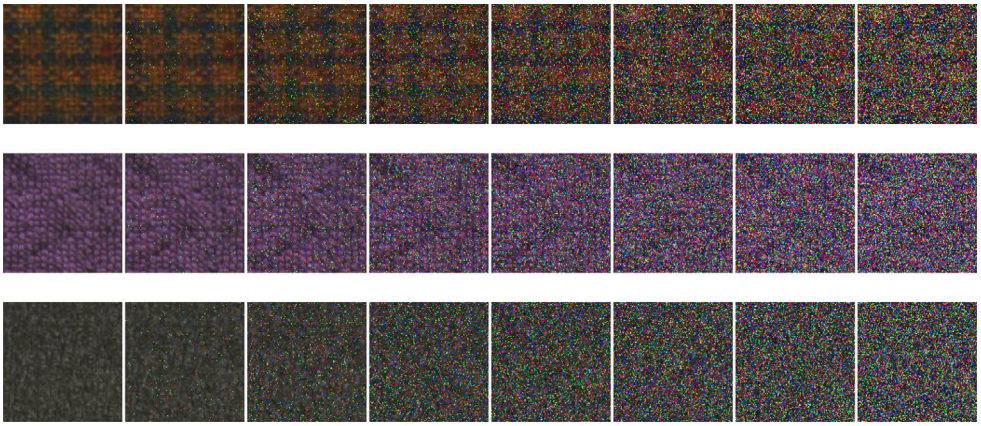


Fig. 6. Examples of noisy images. The first column is the original image, while columns 2 to 8 are transformed images with the following noise densities: 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3.

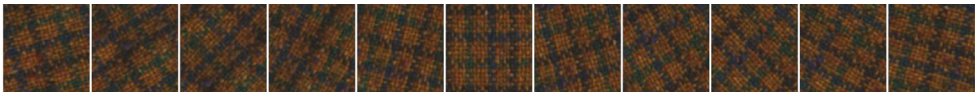


Fig. 7. Samples of a texture image from the Outex dataset transformed at 11 distinct rotation angles. From left to right: 15°, 30°, 45°, 60°, 75°, 90°, 105°, 120°, 135°, 150°, 165°.

used for testing, while the images corresponding to the remaining angles are designated to compose the training set. In this way, this experiment evaluates the ability of our method to tolerate rotation variations. For both proposed representations SSR^1 and SSR^2, the accuracy obtained was 99.87%, confirming that the proposed representations are very discriminative and robust to image transformations. This is because spatio-spectral modeling approach does not consider the order of pixels to calculate the topological measures that are used as input to the RNN.

5.3. Comparison with other methods

To improve the analysis of the proposed method, we compare our performance to the results obtained by other literature methods. In this comparison, the results of the proposed method were computed with the two representations SSR^1 and SSR^2, which obtained

the best performance in the previous analysis. For our method, we considered the LDA classifier using two cross-validation schemes: leave-one-out (loo) and 10-fold. Firstly, we compare our method with the following methods (first block of rows in Table 3): Opponent-Gabor (264) [28], Complete Local Binary Patterns (CLBP) (354) [29], LETRIST (413) [3], LGONBP (1404) [4], Local Phase Quantization (LPQ) (768) [30], Complex Networks Traditional descriptors (CNTD) (324) [8], Sá Junior et al. [31], Multilayer Complex Network Descriptors (MCND) [2], SSN (φ^6_{WB}) (648) [11] and RNN-RGB (180). We extracted the results from the original papers and in [11]. For the LETRIST (413) [3], SWOBP (2244) [6] and LGONBP (1404) [4] methods, we performed the experiments with the same protocol setup, except for using the nearest neighbor classifier with chi-square distance, as implemented in the original paper. The results of these methods were calculated using leave-one-out cross-validation, and the values in parentheses represent the size of the feature vector of the methods.

Table 3 summarizes the results obtained for each method, including the proposed representations SSR. Note that our proposed representations obtain a higher accuracy on the Outex dataset when compared with the considered methods in the first block of results. On the USPTex dataset, the proposed method outperformed all others, except the SSN method, which achieved a marginally higher accuracy (0.4%). For the MBT and CUReT datasets, the performance of the proposed method was also competitive, reaching an accuracy close to the other compared methods. However, it is important to stress that the proposed method achieves superior accuracies for plant species recognition, which is a real and complex problem as presented in Section 5.4.

We also compare our method with others developed for gray-level texture, applying the methods in images converted to gray-level. Note that, in comparison to the proposed method, the gray-level methods (CNTD, LETRIST, and LGONBP) achieved significantly lower accuracies on three datasets (USPTex, Outex and MBT) even when we applied the SSR method on gray-level images. The results also show a high disparity between gray-level and color-texture methods, which is mainly attributed to the necessity of analyzing color information to effectively discriminate the images in these datasets. On the other hand, on the CUReT dataset, color information is not crucial for texture characterization, as methods focusing on gray-level texture images achieve high accuracies.

Regarding learning-based methods, our results were compared to the following Deep Convolutional Neural Network (DCNN) architectures: AlexNet [32], VGG16 and VGG19 [33], ResNet50 and ResNet101 [34], and InceptionV3 [5,35]. The results were extracted from [11], which used a feature extraction approach with pre-trained layers in the ImageNet [36] dataset. As shown in [2], pre-trained DCNNs produce better results than models trained from scratch using the color-texture datasets. The images are resized to match the input layer size of the DCNNs, and Global Average Polling (GAP) is applied to the convolutional layers to generate the feature vectors, following methodologies previously employed in other studies [37,38]. The size of the feature vectors corresponds to the number of 2D activation maps of the last convolutional layer, however, limited to a size smaller than 800. In this case, the previous convolutional layers are utilized for computing the image features, resulting in feature vectors with size 768, 512, 256 e 512 for the InceptionV3, ResNet, AlexNet and VGG models, respectively. More details about the experimental setup of these CNNs can be found in [11].

When compared to DCNN architectures, the proposed method achieved higher accuracy on the Outex and MBT datasets. On the other hand, on the USPTex and CUReT datasets, the ResNet50 model achieved slightly higher accuracy than the SSR approach. On the Outex and MBT datasets, the proposed method significantly improves the accuracy compared to various DCNNs architectures, for example, from 91.5% (ResNet50) to 96.8% on the Outex and from 94.9% (ResNet50) to 98.20% on the MBT dataset. In particular, on the Outex dataset, which is the most challenging because it is composed of samples under different illuminations, it was observed that all DCNN-based methods fail to achieve high accuracy. These results suggest that the SSR method is capable of achieving competitive results with a simple and fast neural network model, in contrast to DCNNs which are generally larger and more computationally demanding.

Concerning the methods founded on complex network theory and graphs, we compare our approach with four different methods: CNDT integrative and in gray-level images (Backes et al. [8]); shortest paths on graphs (Sá Junior et al. [31]); MCND [2]; and SSN [11]. In this comparison, the SSR method outperformed the others on the Outex dataset, while the SSN obtained very similar accuracies on the USPTex, CUReT, and MBT datasets. Both methods yield representations of comparable sizes, with 648 attributes for the SSN method and 630 attributes in the case of the SSR[1]. This demonstrates that the proposal of our work of using a simple neural network to learn and describe the SSN topology can enhance the color-texture analysis in many cases compared to other traditional measures from complex networks and graph theory. When comparing the SSR method with the RNN applied directly to the RGB image (RNN-RGB method), the proposed method improves the accuracies in the two compared datasets (USPTex and Outex), evidencing the effectiveness of combining complex networks with RNNs for pattern recognition.

Finally, unlike methods that exhibit performance limitations in specific datasets, the proposed method, with its two representations, performs a good average accuracy when considering all scenarios. For instance, in the USPTex dataset, CNN-based approaches surpass the proposed method by up to 0.4%, while the proposed method considerably increases the accuracy on the Outex and MBT datasets compared to these approaches. Similarly, integrative descriptors also show varying results depending on the dataset. For example, on the Outex dataset, the accuracies are lower for these methods due to their lack of spectral analysis of the image.

5.4. Plant species recognition

We also applied the proposed representation in the task of plant species recognition, utilizing samples of leaf texture. Plant species recognition is a challenging problem due to its variety and intrinsic complexity in nature and taxonomic characteristics. This complexity makes the task of manual recognition by biologists and botanists a time-consuming process and subject to human

Table 3
Comparison results for the proposed method and literature methods on 4 color-texture datasets. The results of the deep convolutional networks were obtained from [11]. The results of some methods are not available in the original paper (empty cells).

Methods	USPTex	Outex	MBT	CUReT
Opponent-Gabor [28]	99.1	93.5	97.6	95.8
LPQ integrative [30]	90.4	80.1	95.7	91.7
CLBP integrative [29]	97.4	89.6	98.2	91.8
CNTD integrative [8]	97.9	92.3	98.5	91.9
CNTD (gray-level) [8]	92.3	86.8	83.7	84.2
LETRIST[a] (gray-level) [3]	92.4	82.8	79.1	99.4
LGONBP[a] (gray-level) [4]	83.3	83.5	74.2	99.2
Sá Junior et al. [31]	96.9	91.5	–	–
SWOBP[a] [6]	97.00	79.3	88.3	98.2
RNN-RGB [39]	98.4	94.8	–	–
MCND [2]	99.0	95.4	97.0	97.1
SSN (φ^{6}_{WB}) [11]	99.7	96.6	98.6	98.9
AlexNet [32]	99.6	91.4	97.8	98.2
VGG16 [33]	99.5	91.1	97.2	98.5
VGG19 [33]	99.5	90.7	96.3	98.6
InceptionV3 [5]	99.5	89.5	94.4	97.3
ResNet50 [34]	99.7	91.5	94.9	98.7
ResNet101 [34]	99.5	91.3	94.6	98.8
Proposed method				
SSR^{1}[b]	99.43 (± 0.36)	97.13 (± 0.88)	98.17 (± 0.70)	98.56 (± 0.51)
SSR^{2}[b]	99.21 (± 0.61)	97.13 (± 1.27)	98.21 (± 0.64)	98.62 (± 0.51)
SSR^{1}	99.3	96.7	98.2	98.6
SSR^{2}	99.0	96.8	98.0	98.7
SSR^{1} (gray-level)	95.8	90.8	90.1	96.3
SSR^{2} (gray-level)	96.3	91.6	91.00	96.4

[a] Methods marked used the nearest neighbor classifier with the chi-square distance.
[b] Results using 10-fold cross-validation.

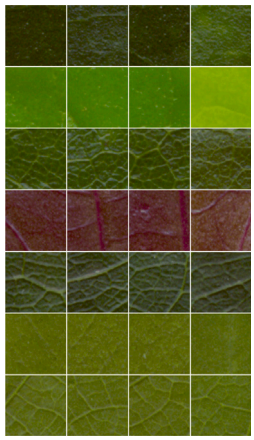


Fig. 8. Examples of 4 samples (columns) of 7 classes (rows) of the plant leaf dataset [40].

error. Furthermore, the variation in textural and color patterns within the same species, or even on a single leaf, along with changes resulting from climatic factors (e.g. rain, weather, and sun), add layers of complexity to this texture classification problem [40].

In this experiment, we used a leaf dataset composed of samples divided into 20 species of the Brazilian flora, each with 20 specimens per species, as described in [40]. The plant specimens were individually captured by scanning their surfaces using a flatbed scanner at a resolution of 1200 dpi. The images may exhibit imperfections due to dirt on the leaf or scanner glass, as well as issues from the scanning process, which can cause shadows or overlaps [40]. To analyze the micro-texture of the leaves, three non-overlapping windows of 128 × 128 pixels were manually extracted from each image, resulting in a dataset of 1200 images. This procedure increases the classification challenge due to the texture variability present on the surface of the same plant. Fig. 8 shows examples of these windows from different species in the dataset.

In the tests, the texture analysis methods were applied in images converted to gray-level, except for our method and the SSN method that use the RGB images. For classification, we use the LDA classifier in a leave-one-out cross-validation scheme. In this scheme, the three windows of the same specimen were used for testing, while the remaining windows served as training data. This

Table 4
Accuracies for different methods of texture analysis applied to the task of the plant leaf recognition (leaf and windows) using the LDA classifier.

Methods	Leaf (%)	Windows (%)
AHP [41]	88.75	79.17
GLDM [42]	88.00	79.92
LCP [43]	88.50	76.58
LFD [44]	85.50	74.67
LPQ [30]	83.75	73.00
Fourier [42]	75.00	65.75
Fractal [45]	80.50	70.75
Gabor [46,47]	86.75	77.25
CNTD [8]	89.50	83.33
SSN [11]	95.50	91.83
Proposed method		
SSR^1	96.25	94.08
SSR^2	95.50	93.67

procedure was repeated for all leaves, ensuring that windows from the same specimen were not used simultaneously in both the training and testing sets.

To measure the characterization ability of the texture analysis methods, we adopted two distinct metrics. The first metric is accuracy, calculated as the percentage of leaves correctly classified based on their three cropped windows. For this purpose, the LDA computes the posterior probability of a window belonging to a given class, generating a probability vector equal to the number of classes or species (20) in the dataset. Thus, for the three windows J_i^1, J_i^2, J_i^3 extracted from the same leaf i, we calculated the average of the three class probability vectors given by the LDA classifier. Then, the class with the highest average probability is assigned to the leaf i from which these windows originated. This strategy, proposed in [40], aims to minimize errors in species identification of a leaf, addressing the high variability in textural and morphological patterns found on the surface of the same leaf. The second metric employed in our study focuses on the individual classification of windows, with accuracy calculated as the percentage of correctly classified windows. Note that, in this scenario, windows from the same leaf may be classified into different species, making the problem more challenging and testing the robustness of the texture analysis methods.

Table 4 presents the results for various texture analysis methods in terms of both window and whole leaf classification. Regarding the leaf classification, the proposed method SSR^1, with an accuracy of 96.25%, outperforms all compared methods, including the SSN method that also uses RGB images. It is also possible to note that color information plays an important role in this task, as recognition performance consistently improves when all three color channels are considered. On the other hand, from the results, it is evident that the classification of individual windows is more challenging. In this task, the performance of several methods significantly decreased compared to the classification of leaves, for example, from 88.75% to 79.15% with AHP and from 88.50% to 76.58% using the LCP. This behavior is expected due to the high variation of texture patterns in leaves of the same species.

In this dataset, both the proposed method and SSN achieved the highest performance, with our method surpassing the SSN in both tasks (leaf and windows). To more effectively illustrate the performance difference between the two methods, Fig. 9 presents the confusion matrices for the window classification task. The confusion matrices show that the SSN method has higher misclassifications in certain classes, whereas the proposed representations exhibit a superior ability to differentiate between the classes. For instance, in the SSN method (Fig. 9(c)), eight samples from class 12 are incorrectly classified as class 16. In contrast, the proposed method misclassifies only two samples from class 12 as class 16. Note in Fig. 10(a) that there is a certain similarity between the leaf texture patterns of the two species (classes), which makes plant species recognition a challenging task.

To visualize the discriminative capability of the representations computed by the two methods, Fig. 10(b–d) presents the first 100 features of each representation corresponding to the texture images shown in Fig. 10(a). As can be seen, the proposed method yields representations (learned weights) that effectively separate the two classes while maintaining similarity within each class. On the other hand, the SSN method generates representations that are very similar inter-class, struggling to differentiate the leaf texture patterns. This indicates that the RNN can effectively learn to synthesize spatio-spectral topological measures into color-texture information, thereby enabling its learned weights to represent and distinguish different leaf species. Therefore, the results demonstrate the characterization capability and robustness of the proposed method, even when applied to real and complex problems. This makes it an interesting tool for developing automated systems that can identify plant species efficiently and with low computational cost.

5.5. Computational complexity analysis

To evaluate the complexity of the methods, consider a color image with ($N_p = w * h * z$) pixels. The computational complexity of our method can be divided into two fundamental steps for analysis First, the cost of transforming a color-texture image into a SSN is directly proportional to the number of pixels n_p and the largest radius used R. This implies that to construct the radially symmetric neighborhood of the pixels, all pixels within a distance less than R must be visited. Thus, calculating the radially symmetric neighborhood of the pixels necessitates $G^R = (\frac{(2R+1)^2-1}{2}z) + z$ operations. It is important to note that the complex networks N, W, and B are built simultaneously. As a result, the computational complexity of this modeling is $O(N_p G^R)$.

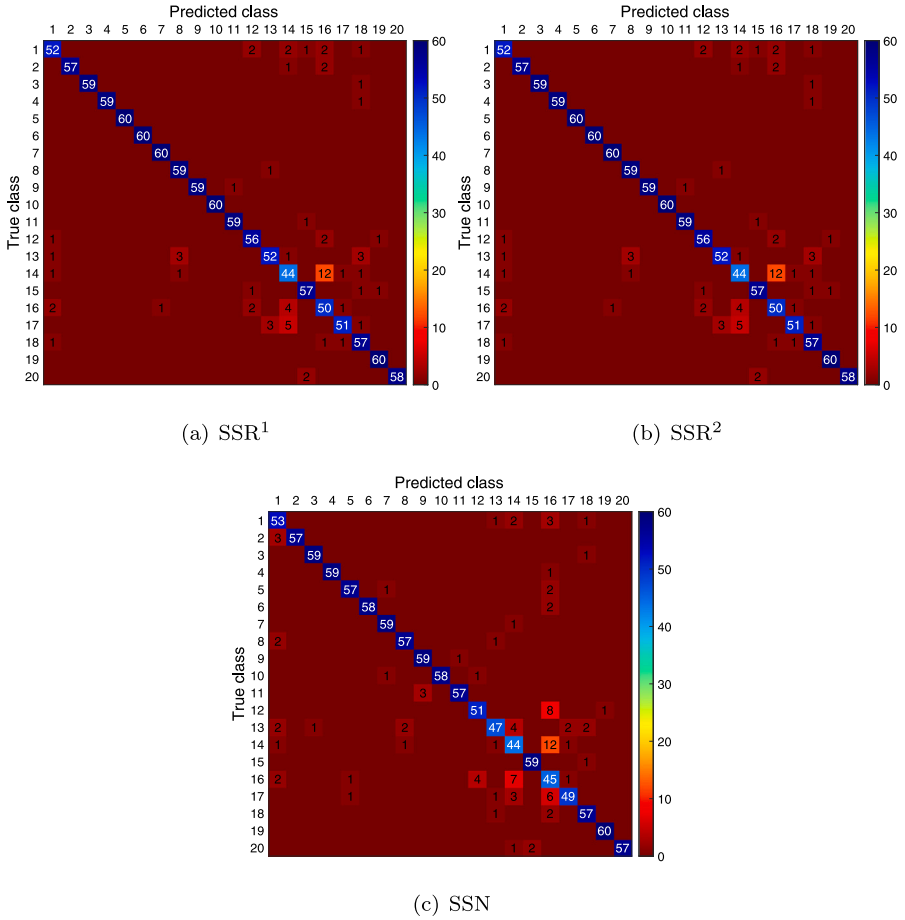


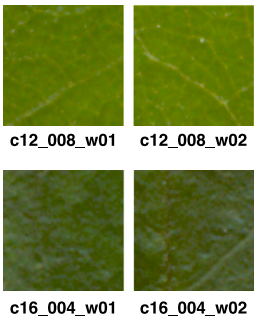
Fig. 9. Confusion matrices for proposed representations and SSN method in the plant species recognition task.

The second step of our method involves using RNNs to learn representations from the modeled complex networks (SSN). To assess the computational cost of the RNN, consider a set of $N_{\vec{x}}$ input vectors $\vec{x}$, each of size R. The most significant cost of this neural network comes from solving the equation $\vec{f} = DZ^T(ZZ^T)^{-1}$. Firstly, calculating the output of the hidden layer $Z = \phi(WX)$ and transposing Z takes about $T_1 = Q(R+1)n_{\vec{x}}$ and $T_2 = (Q+1)N_{\vec{x}}$ operations, respectively. Then, to perform DZ^T is necessary $T_3 = N_{\vec{x}}(Q+1)$ operations, while calculating ZZ^T and its inverse, requires $T_4 = (Q+1)^2 N_{\vec{x}}$ and $T_5 = (Q+1)^3$ operations, respectively. Finally, multiplying the two terms DZ^T and $(ZZ^T)^{-1}$ requires approximately $T_6 = (Q+1)^2$ operations. Therefore, the total computational complexity of the RNN can be considered as $O(N_{\vec{x}})$, since Q and R are generally much smaller than $N_{\vec{x}}$ and are independent of the number of input vectors. For example, in some experimental cases, $N_{\vec{x}} = 200 \times 200$, $R = 10$ and $Q = 24$.

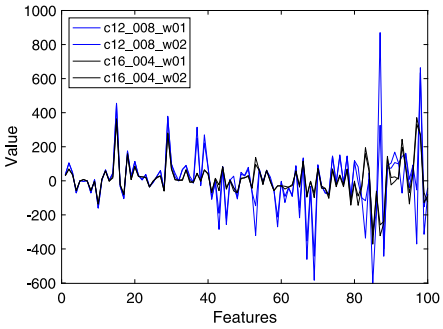
Regarding methods similar to ours, those based on complex network theory (such as SSN, MCND, and CNDT) typically require a computational cost similar to the first step (modeling) of our method. In other words, the number of operations is directly related to the number of pixels in the image and the size of the neighborhood radius. On the other hand, the RNN-RGB method exhibits a cost close to the second step of our method. It also uses a number of input vectors equivalent to the number of pixels in the image, but with vector sizes slightly larger than those used in our proposed method, varying between 9 and 49.

6. Conclusion

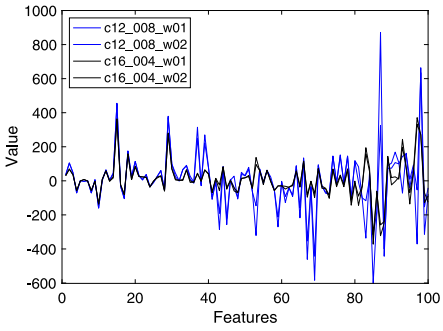
This paper introduces a new approach to color-texture characterization, which uses RNNs to learn Spatio-Spectral Representations from a complex network handcraft modeling. We model the color image as a Spatio-Spectral Network (SSN) and use its vertex measurements to train the RNNs. The learned weights of the output layer are adopted as representation. The spatial and spectral patterns were characterized by two SSN derived from the original network: W with only within-channel links and B that contains only between-channel connections. Thus, different RNNs were trained to learn the topological features from the three networks: original N, within-channel W and between-channel B. We proposed a final representation formed by combining the learned weights from the three complex networks, which we called Spatio-Spectral Representation (SSR). This scheme produces rich color-texture representations that contain a diverse variety of texture patterns associated to spatio-spectral properties. The results outperformed



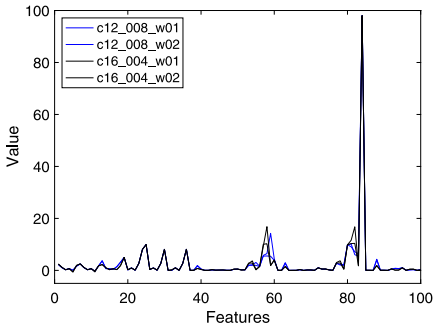
(a) Misclassified images



(b) SSR^1



(c) SSR^2



(d) SSN

Fig. 10. Examples of misclassified images by the SSN method from two plant species (classes 12 and 16), along with their representations (first 100 features) computed using the proposed method and the SSN method.

almost all the compared methods in the four datasets, demonstrating that the proposed method produces a highly discriminative feature vector. Our representations also proved to be very competitive compared to deep CNN architectures. The experiments demonstrated that the proposed method is robust against rotation and noise and is competitive in the challenging task of plant species recognition. These findings motivate the use of our method in many computer vision problems.

A limitation that requires further exploration is the physical interpretation of the representations learned by the proposed method. For future work, in the SSN modeling step, other topological measures and edge weight functions can be studied. We also think that several enhancements can be investigated in the feature learning process: (i) employing a deeper RNN for more comprehensive feature learning; (ii) adopting a RNN classification model rather than a regression model to train the neural network, (iii) and exploring alternative neural network architectures, such as D-RVFL and convolutional networks. Finally, due to its flexibility, we believe that our method can be easily extended to other domains and problems such as dynamic texture analysis.

CRediT authorship contribution statement

Lucas C. Ribas: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Leonardo F.S. Scabini:** Conceptualization, Formal analysis, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing. **Rayner H.M. Condori:** Investigation, Methodology, Writing – original draft. **Odemir M. Bruno:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known conflict of interests that could have inappropriately influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

Lucas C. Ribas gratefully acknowledges the financial support grant from São Paulo Research Foundation (FAPESP), Brazil (Grant #2023/04583-2). Leonardo Scabini acknowledges funding from FAPESP (Grants #2019/07811-0 and #2021/09163-6) and the National Council for Scientific and Technological Development (CNPq), Brazil (Grant #142438/2018-9). Odemir M. Bruno thanks the financial support of CNPq (Grant # 307897/2018-4) and FAPESP (Grant 2018/22214-6).

References

[1] T. Mäenpää, M. Pietikäinen, Classification with color and texture: jointly or separately? Pattern Recognit. 37 (8) (2004) 1629–1640.
[2] L.F. Scabini, R.H. Condori, W.N. Gonçalves, O.M. Bruno, Multilayer complex network descriptors for color–texture characterization, Inform. Sci. 491 (2019) 30–47.
[3] T. Song, H. Li, F. Meng, Q. Wu, J. Cai, LETRIST: Locally encoded transform feature histogram for rotation-invariant texture classification, IEEE Trans. Circuits Syst. Video Technol. 28 (7) (2017) 1565–1579.
[4] T. Song, J. Feng, L. Luo, C. Gao, H. Li, Robust texture description using local grouped order pattern and non-local binary pattern, IEEE Trans. Circuits Syst. Video Technol. 31 (1) (2020) 189–202.
[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: The IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 2818–2826.
[6] T. Song, J. Feng, S. Wang, Y. Xie, Spatially weighted order binary pattern for color texture classification, Expert Syst. Appl. 147 (2020) 113167.
[7] T. Song, L. Xin, C. Gao, T. Zhang, Y. Huang, Quaternionic extended local binary pattern with adaptive structural pyramid pooling for color image representation, Pattern Recognit. 115 (2021) 107891.
[8] A.R. Backes, D. Casanova, O.M. Bruno, Texture analysis and classification: A complex network-based approach, Inform. Sci. 219 (2013) 168–180.
[9] L.C. Ribas, R. Riad, R. Jennane, O.M. Bruno, A complex network based approach for knee Osteoarthritis detection: Data from the Osteoarthritis initiative, Biomed. Signal Process. Control 71 (2022) 103133.
[10] L.C. Ribas, J.J.M. Sá Junior, L.F. Scabini, O.M. Bruno, Fusion of complex networks and randomized neural networks for texture analysis, Pattern Recognit. 103 (2020) 107189.
[11] L.F. Scabini, L.C. Ribas, O.M. Bruno, Spatio-spectral networks for color-texture analysis, Inform. Sci. 515 (2020) 64–79.
[12] W.F. Schmidt, M.A. Kraaijveld, R.P.W. Duin, Feedforward neural networks with random weights, in: Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems, 1992, pp. 1–4.
[13] Y.-H. Pao, Y. Takefuji, Functional-link net computing: theory, system architecture, and functionalities, Computer 25 (5) (1992) 76–79.
[14] Y.-H. Pao, G.-H. Park, D.J. Sobajic, Learning and generalization characteristics of the random vector functional-link net, Neurocomputing 6 (2) (1994) 163–180.
[15] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006) 489–501.
[16] J.J.M. Sá Junior, L.C. Ribas, O.M. Bruno, Randomized neural network based signature for dynamic texture classification, Expert Syst. Appl. 135 (2019) 194–200.
[17] L.C. Ribas, J.J. de Mesquita Sá Junior, A. Manzanera, O.M. Bruno, Learning graph representation with Randomized Neural Network for dynamic texture classification, Appl. Soft Comput. 114 (2022) 108035.
[18] L. Scabini, K.M. Zielinski, L.C. Ribas, W.N. Gonçalves, B. De Baets, O.M. Bruno, RADAM: Texture recognition through randomized aggregated encoding of deep activation maps, Pattern Recognit. 143 (2023) 109802.
[19] R. Penrose, A generalized inverse for matrices, Math. Proc. Camb. Phil. Soc. 51 (3) (1955) 406—413.
[20] D. Calvetti, S. Morigi, L. Reichel, F. Sgallari, Tikhonov regularization and the L-curve for large discrete ill-posed problems, J. Comput. Appl. Math. 123 (1) (2000) 423–446.
[21] A.N. Tikhonov, On the solution of ill-posed problems and the method of regularization, Dokl. Akad. Nauk SSSR 151 (3) (1963) 501—504.
[22] S.K. Park, K.W. Miller, Random number generators: good ones are hard to find, Commun. ACM 31 (10) (1988) 1192–1201.
[23] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, S. Huovinen, Outex-new framework for empirical evaluation of texture analysis algorithms, in: Pattern Recognition, 2002. Proceedings. 16th International Conference on, Vol. 1, IEEE, 2002, pp. 701–706.
[24] A.R. Backes, D. Casanova, O.M. Bruno, Color texture analysis based on fractal descriptors, Pattern Recognit. 45 (5) (2012) 1984–1992.
[25] S. Abdelmounaime, H. Dong-Chen, New Brodatz-based image databases for grayscale color and multiband texture analysis, ISRN Mach. Vis. 2013 (2013).
[26] K.J. Dana, B. Van Ginneken, S.K. Nayar, J.J. Koenderink, Reflectance and texture of real-world surfaces, ACM Trans. Graph. 18 (1) (1999) 1–34.
[27] B. Everitt, G. Dunn, Applied Multivariate Data Analysis, A Hodder Arnold Publication, Wiley, 2001.
[28] A. Jain, G. Healey, A multiscale representation including opponent color features for texture recognition, IEEE Trans. Image Process. 7 (1) (1998) 124–128.
[29] Z. Guo, L. Zhang, D. Zhang, A completed modeling of local binary pattern operator for texture classification, IEEE Trans. Image Process. 19 (6) (2010) 1657–1663.
[30] V. Ojansivu, J. Heikkilä, Blur insensitive texture classification using local phase quantization, in: International Conference on Image and Signal Processing, Springer, 2008, pp. 236–243.
[31] J.J.d.M.S. Junior, P.C. Cortez, A.R. Backes, Color texture classification using shortest paths in graphs, IEEE Trans. Image Process. 23 (9) (2014) 3751–3761.
[32] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[33] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.
[34] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: The IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, pp. 1–9.
[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, Ieee, 2009, pp. 248–255.
[37] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3606–3613.
[38] M. Cimpoi, S. Maji, I. Kokkinos, A. Vedaldi, Deep filter banks for texture recognition, description, and segmentation, Int. J. Comput. Vis. 118 (1) (2016) 65–94.
[39] J.J.d.M. Sá Junior, A.R. Backes, O.M. Bruno, Randomized neural network based signature for color texture classification, Multidimens. Syst. Signal Process. 30 (3) (2019) 1171–1186.
[40] D. Casanova, J.J. de Mesquita Sá Junior, O.M. Bruno, Plant leaf identification using Gabor wavelets, Int. J. Imaging Syst. Technol. 19 (3) (2009) 236–243, http://dx.doi.org/10.1002/ima.20201.

[41] Z. Zhu, X. You, C.P. Chen, D. Tao, W. Ou, X. Jiang, J. Zou, An adaptive hybrid pattern for noise-robust texture analysis, Pattern Recognit. 48 (8) (2015) 2592–2608.
[42] J.S. Weszka, C.R. Dyer, A. Rosenfeld, A comparative study of texture measures for terrain classification, IEEE Trans. Syst., Man, Cybern. (4) (1976) 269–285.
[43] Y. Guo, G. Zhao, M. Pietikäinen, Texture classification using a linear configuration model based descriptor, in: BMVC, Citeseer, 2011, pp. 1–10.
[44] R. Maani, S. Kalra, Y.-H. Yang, Noise robust rotation invariant features for texture classification, Pattern Recognit. 46 (8) (2013) 2103–2116.
[45] A.R. Backes, D. Casanova, O.M. Bruno, Plant leaf identification based on volumetric fractal dimension, Int. J. Pattern Recognit. Artif. Intell. 23 (06) (2009) 1145–1160.
[46] B.S. Manjunath, W.-Y. Ma, Texture features for browsing and retrieval of image data, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 837–842.
[47] J. Daugman, C. Downing, Gabor wavelets for statistical pattern recognition, in: The Handbook of Brain Theory and Neural Networks, MIT Press, 1995, pp. 414–419.