Expert Systems With Applications 135 (2019) 194–200

Contents lists available at ScienceDirect

Expert Systems With Applications

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

# Randomized neural network based signature for dynamic texture classification

Jarbas Joaci de Mesquita Sá Junior[a,*], Lucas Correia Ribas[b], Odemir Martinez Bruno[c]

[a] Curso de Engenharia da Computação, Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Campus de Sobral, Universidade Federal do Ceará, Rua Coronel Estanislau Frota, 563, Centro, Sobral, Ceará, CEP: 62010-560, Brasil
[b] Institute of Mathematics and Computer Science, University of São Paulo, Avenida Trabalhador São-Carlense, 400, Centro, São Carlos 13566-590, SP, Brazil
[c] São Carlos Institute of Physics, University of São Paulo, PO Box 369, São Carlos 13560-970, SP, Brazil

ARTICLE INFO

*Article history:*
Received 18 December 2018
Revised 19 April 2019
Accepted 30 May 2019
Available online 30 May 2019

*Keywords:*
Dynamic textures
Randomized neural network
Dynamic texture analysis method

ABSTRACT

Dynamic texture analysis has been the focus of intensive research in recent years. Thus, this paper presents an innovative and highly discriminative dynamic texture analysis method, whose signature is composed of the weights of the output layer of a randomized neural network after a training procedure. This training is performed by using the pixels of slices of each orthogonal plane of the video (*XY*, *YT*, and *XT*) as input feature vectors and corresponding output labels. The obtained video signature provided an accuracy of 97.05%, 98.54%, 97.74% and 96.51% on the UCLA-50 classes, UCLA-9 classes, UCLA-8 classes and Dyntex++, respectively. These results, when compared to other dynamic texture analysis methods, demonstrate that our descriptors are very effective and that our proposed approach can contribute significantly to the field of dynamic texture analysis.



## 1. Introduction

Dynamic texture analysis is an important research area of computer vision responsible for extracting meaningful characteristics from dynamic texture videos. This field has gained much attention due to the range of applications, such as monitoring of traffic in highway (Chan & Vasconcelos, 2005; Derpanis & Wildes, 2011), human activity recognition (Kellokumpu, Zhao, & Pietikäinen, 2008), facial expression recognition (Zhao & Pietikainen, 2007), medical videos analysis (Brieu et al., 2010), crowd analysis and management (Chan, Morrow, & Vasconcelos, 2009), among others.

Although the understanding and perception of dynamic textures are easy to humans, their formal definition and description using computational methods are a hard task (Gonçalves & Bruno, 2013a). Unlike traditional texture images, dynamic textures are sequences of images with texture patterns that represent a dynamic object or process and present certain stationary properties in space and time (Doretto, Chiuso, Wu, & Soatto, 2003). Therefore, dynamic textures can be defined as an extension of traditional texture images to the spatial and temporal domain, which correspond to the appearance and motion characteristics, respectively (Gonçalves & Bruno, 2013b). Examples of dynamic textures are sea waves, boiling water, waterfall, metal corrosion process and fire.

The addition of the time domain causes new challenges in the characterization task, since it is necessary to combine appearance and motion information (e.g. some methods analyze textures based on motion only), and to process it with low computational complexity. To overcome this, many approaches have been proposed, each one investigating characteristics of the dynamic texture video in a different way. Most of the existing dynamic texture methods can be divided into six categories: based on motion (e.g. optical flow (Fazekas & Chetverikov, 2007; Péteri & Chetverikov, 2005; Polana & Nelson, 1997; Soygaonkar, Paygude, & Vyas, 2015)); based on models (e.g. linear dynamical systems (Chan & Vasconcelos, 2008), hidden Markov model (Qiao & Weng, 2015) and ensemble support vector machines (Yang, Xia, Liu, Zhang, & Huang, 2016)); based on filters (e.g. wavelet filters (Dubois, Péteri, & Ménard, 2009) and Gabor filter (Gonçalves, Machado, & Bruno, 2011)); based on geometric properties (e.g. spatiotemporal motion trajectory (Otsuka, Horikoshi, Suzuki, & Fujii, 1998)); based on discrimination (e.g. local binary patterns (Tiwari & Tyagi, 2016a; 2016b; 2017; Zhao & Pietikainen, 2007)); and based on agents (e.g. deterministic partially self-avoiding walks (Gonçalves & Bruno, 2013b; 2013c)).

In this paper, we propose a method based on randomized neural network to extract signatures from dynamic textures, aiming to provide a novel tool to, but not limited to, the range of applications

* Corresponding author.
*E-mail addresses:* jarbas_joaci@yahoo.com.br (J.J.d.M. Sá Junior), lucasribas@usp.br (L.C. Ribas), bruno@ifsc.usp.br (O.M. Bruno).

https://doi.org/10.1016/j.eswa.2019.05.055
0957-4174/

aforementioned. Our contributions are: (i) demonstrating that a powerful method for texture analysis can be adapted successfully for dynamic texture analysis, obtaining accuracies higher than many literature methods with competitive processing time, and (ii) opening a promising research field for dynamic texture analysis and recognition. To explain the method, the remainder of the paper is organized as follows. Randomized neural network algorithm is described in Section 2. Section 3 describes the proposed method for dynamic texture analysis. In Section 4, we describe the experimental setup and datasets. Results and discussion are presented in Section 5, which is followed by the conclusion in Section 6.

## 2. Randomized neural network

A randomized neural network has a unique hidden layer *feed-forward* with a very fast learning algorithm (Huang, Zhu, & Siew, 2006; Pao, Park, & Sobajic, 1994; Pao & Takefuji, 1992; Schmidt, Kraaijveld, & Duin, 1992). In this neural network, the weights of the hidden neurons are randomly generated and the weights of the output neurons can be determined according to the least-squares method. This solution makes the learning process faster and, therefore, allows the neural networks to deal with problems that require more processing speed. Thus, due to its simplicity, easy implementation, high predictive performance, among other characteristics, this type of neural network has attracted the interest of researchers in recent years (Bacciu, Colombo, Morelli, & Plans, 2018; Dudek, 2019; Pratama et al., 2017; Pratama, Angelov, Lughofer, & Er, 2018; Zhang & Suganthan, 2016; Zhang, Wu, Cai, Du, & Yu, 2019).

To describe the learning algorithm used in this work, let $X = [\vec{x_1}, \vec{x_2}, \ldots, \vec{x_N}]$ be a matrix of $N$ input feature vectors with $p$ attributes, and $D = [\vec{d_1}, \vec{d_2}, \ldots, \vec{d_N}]$ be a matrix with the corresponding label vectors. Initially, the weights of the hidden neurons, which can be generated using a uniform or Gaussian distribution, are arranged as a matrix $W$. In this matrix, each line represents the weights of a determined hidden neuron $q$ and the first column represents the bias weights.

A constant $-1$ is added to each feature vector $\vec{x_i}$ as first attribute in order to connect to the bias weights of the hidden neurons. Next, the output of each hidden neuron is computed using the activation function $\phi(WX)$, which may be, for instance, a sigmoid or hyperbolic tangent function. The output of $\phi(.)$ is used to compose a matrix $Z = [\vec{z_1}, \vec{z_2}, \ldots, \vec{z_N}]$ of feature vectors, which are used as input in the output layer. Again, $-1$ is added as first attribute to each vector $\vec{z_i}$ in order to connect to the bias weights of the output neurons.

Finally, the weights of the hidden neurons are organized as a matrix $M$, in which each line represents the output of an output neuron. These weights aim to satisfy the equation $D = MZ$. For this purpose, we can use the Moore–Penrose pseudo-inverse (Moore, 1920; Penrose, 1955), thus resulting in the following equation

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

Moreover, it is common in many problems that the matrix $ZZ^T$ becomes near singular, resulting in an inaccurate inverse. To solve this problem, it is possible to use the Tikhonov regularization (Calvetti, Morigi, Reichel, & Sgallari, 2000; Tikhonov, 1963), according to the following equation

$$M = DZ^T(ZZ^T + \lambda I)^{-1}, \tag{2}$$

where $0 < \lambda < 1$ and $I$ is an identity matrix.

## 3. Proposed method

In this section, we describe the proposed method for dynamic texture analysis, which is based on the static texture signature presented in Sá Junior and Backes (2016). Fig. 1 illustrates the main steps of the proposed method. This method first divides the dynamic texture video into three orthogonal planes, as can be seen in Fig. 1(a). For each orthogonal plane, matrices of input feature vectors $X$ and its respective labels $D$ are built from each slice. These input and output matrices are submitted to randomized neural network, and the weights of the output layer are used as signature of the slice (Fig. 1(b)). Next, the average of these signatures is the signature of the orthogonal plane. Lastly, the final signature is the concatenation of the three orthogonal plane signatures, as shown in Fig. 1(c). Additionally, Fig. 2 shows a flowchart of our proposed method.

It is important to mention that we chose scalar values as labels in order to simplify our method, since this procedure implies a neural network with only one output neuron, whose weights can be directly used as image descriptors. However, there is no technical reason that prevents using more neurons in the output layer. In this case, it is necessary to deal with the problem of how to combine weights from multiple output neurons in order to build a concise and discriminative signature.

### 3.1. Orthogonal planes

The three orthogonal planes strategy is a well-established and efficient way to analyze appearance and motion characteristics of dynamic texture videos (Gonçalves & Bruno, 2013b; Tiwari & Tyagi, 2016b; Zhao & Pietikainen, 2007). Basically, the video is divided into three orthogonal planes, denominated *XY, XT* and *YT*. Considering the video as a cube, the orthogonal planes are slices in vertical (*YT*), horizontal (*XT*) and time (*XY*) axes. In other words, in the case of the *XY* plane, the slices are the frames of the video. The idea is that each plane highlights different characteristics of the dynamic texture: the *XY* plane is responsible for the appearance characteristics, and the *XT* and *YT* planes describe the motion characteristics.

The planes can be formally described by means of the definition of neighboring pixels for *XY, XT* and *YT*. In the *XY* plane, the neighboring pixels of a pixel $i$ are defined by the neighborhood function $\nu^{XY}(i)$. This one defines a pixel $j$ as neighbor of $i$ if the Euclidean distance between them is lower than or equal to $R$, and the temporal coordinates $t_i$ and $t_j$ are equal, according to

$$\nu(i)^{XY} = \left\{ j \mid \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq R \text{ and } t_i = t_j \right\}, \tag{3}$$

where $x$, $y$ and $t$ are the Cartesian coordinates of the pixel.

Similarly, the neighborhood function for the *XT* and *YT* planes are defined as:

$$\nu(i)^{XT} = \left\{ j \mid \sqrt{(x_i - x_j)^2 + (t_i - t_j)^2} \leq R \text{ and } y_i = y_j \right\}, \tag{4}$$

$$\nu(i)^{YT} = \left\{ j \mid \sqrt{(t_i - t_j)^2 + (y_i - y_j)^2} \leq R \text{ and } x_i = x_j \right\}. \tag{5}$$

### 3.2. Randomized neural network based signature

In this paper, we propose to use the average of the randomized neural network output weights $M$ of the slices of each orthogonal plane to characterize a dynamic texture. For this, we first propose to divide each slice of a given orthogonal plane $\Omega$ into $L \times L$ (L is odd) joint windows. From these windows, we construct the input and output vectors: the gray level value of the central pixel $i$ is considered a desirable label $d_i$ and the neighboring border pixel values are the output feature vector $\vec{x_i}$. Fig. 3 illustrates this step for a window $5 \times 5$ from a slice of the *XY* plane. For each slice of an orthogonal plane, we construct a matrix of feature vector $X_{(\Omega)}$ concatenating the feature vectors $\vec{x_i}$ obtained from windows located on every possible slice pixel position, that is, position where

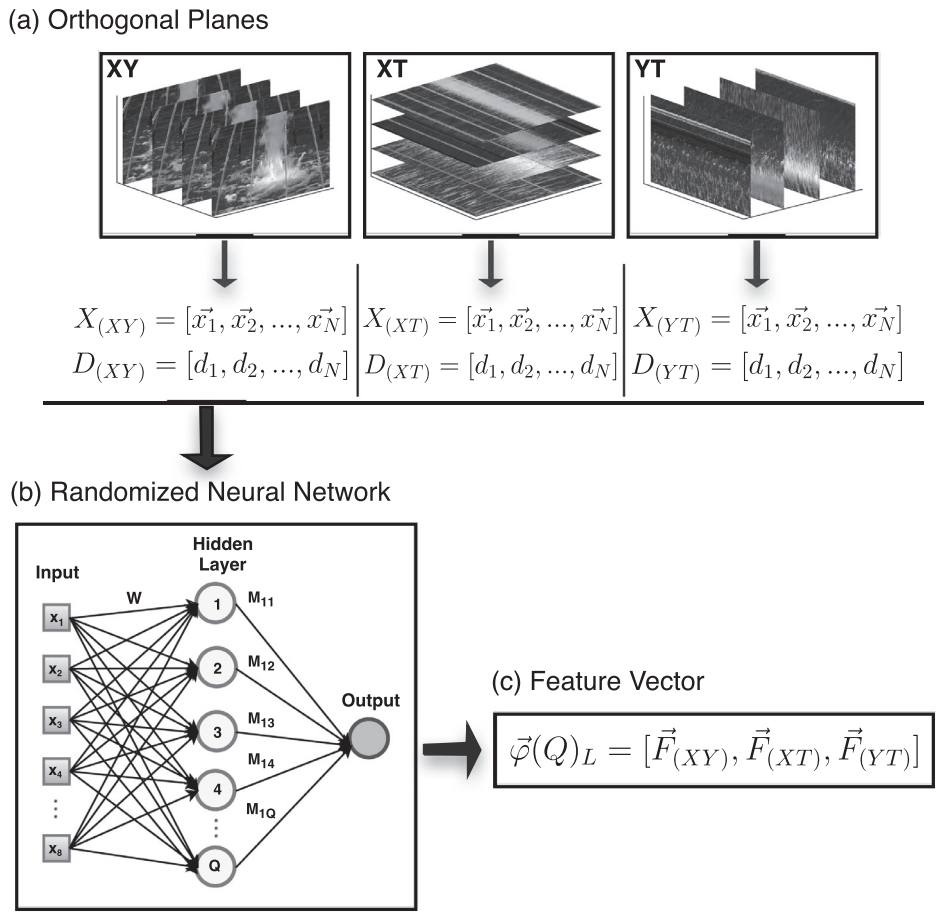


**Fig. 1.** Main steps of the proposed method: (a) set of matrices of input and output vectors for each orthogonal plane; (b) the randomized neural network training; (c) the average weights of the output layer for each orthogonal plane are concatenated.

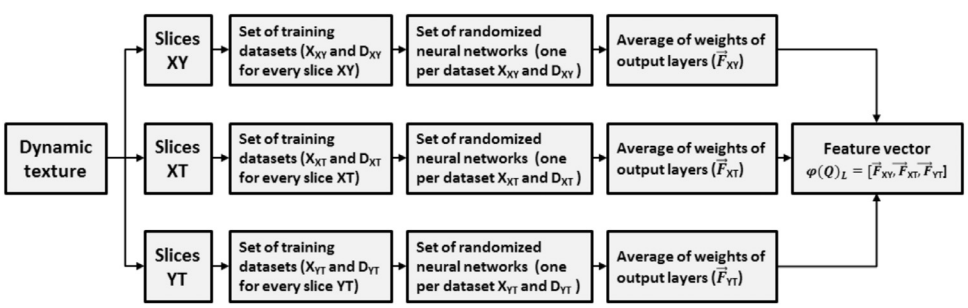


**Fig. 2.** Flowchart of the proposed method.

the window is completely within the slice (for instance, a slice $5 \times 5$ allows 9 different positions for a window $3 \times 3$). The matrix of labels $D_{(\Omega)}$ is composed of the labels $d_i$ associated to the pixels.

To try to analyze micro and macro textures, we propose windows of three different sizes $3 \times 3$, $5 \times 5$ and $7 \times 7$ with neighboring border pixels determined by a circle with radius less than $\sqrt{2}$, $\sqrt{8}$ and $\sqrt{13}$, respectively (same windows and radii proposed in Sá Junior & Backes, 2016). However, it is important to stress that the efficiency of this strategy depends on the resolution of the video. The order criterion for the elements $\{x_1, x_2, \ldots, x_p\}$ is established according to Fig. 3.

After that, we define $\phi(\cdot)$ (sigmoid function) and the values of the weight matrix $W$. Generally, these weights are generated in a random way and can be different in each new training step. However, in feature extraction methods, it is important that feature vector values be always the same for the same image. Therefore, it is necessary to use always the same weights values for an image. In this way, we use the classical linear congruent generator (LCG) (Lehmer, 1951; Park & Miller, 1988) to obtain the pseudorandom uniform numbers for the matrix $W$, according to

$$V(n+1) = (a * V(n) + b) \bmod c, \tag{6}$$

where $V$ is the random number sequence and the values of $a$, $b$ and $c$ are parameters. The sequence $V$ has length $E = Q(p+1)$, where $p$ and $Q$ are the number of attributes of the input vector $\vec{x_i}$ and the number of neurons of the hidden layer, respectively. It is started by $V(1) = E + 1$, and the values of the parameters are $a = E + 2$, $b = E + 3$ and $c = E^2$ (values used in Sá Junior & Backes, 2016). Then, the matrix $W$ is composed of the vector $V$ divided into $Q$ segments of $p + 1$ values. Finally, the values of the matrices $W$ and $X$ (each line) are normalized to zero mean and unit variance.

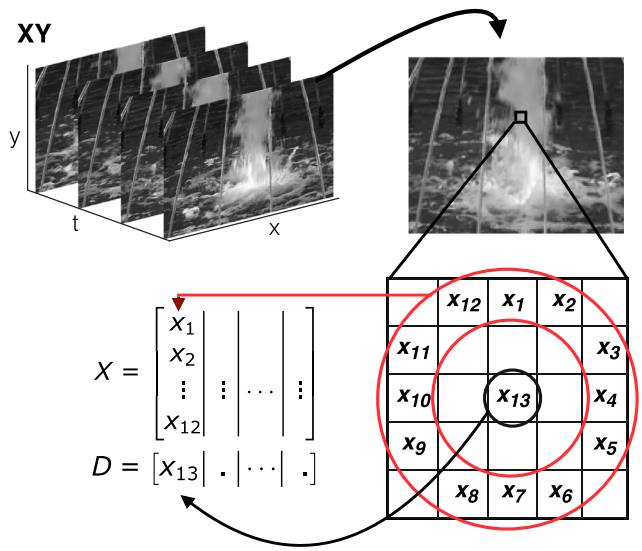


**Fig. 3.** Example of extraction of an input feature vector and its respective label from a window of a slice of the orthogonal plane *XY*.

The feature vector is constructed based on the matrix $M$, which becomes a vector $\vec{f} = DZ^T(ZZ^T + \lambda I)^{-1}$, where $\lambda = 10^{-3}$. Note that $\vec{f}$ has length $Q+1$ due to the bias value. Thus, we propose the feature vector of an orthogonal plane $\Omega$ as the average $\mu$ of the vectors $\vec{f}$ obtained from its slices. In this way, for the *XY* plane, the feature vector is given by

$$\vec{F}_{(XY)} = \mu\left(\begin{bmatrix}\vec{f}_1\\ \vec{f}_2\\ \vdots\\ \vec{f}_T\end{bmatrix}\right), \tag{7}$$

where $T$ is the number of slices of the *XY* plane (i.e. the number of frames of the video). At this point, it is important to stress that we averaged the vectors $\vec{f}$ in order to speed up our method, once learning the regression weights over all frames simultaneously would make matrices $X$ and $D$ extremely large and, therefore, computing the descriptors would be a very slow task.

Thus, to describe appearance and motion characteristics of the dynamic texture, a combined feature vector $\vec{F}(Q)$ considering the three orthogonal planes is given by the concatenation

$$\vec{\varphi}(Q)_L = [\vec{F}_{(XY)}, \vec{F}_{(XT)}, \vec{F}_{(YT)}]. \tag{8}$$

The second feature vector can be obtained by concatenating the previous features vectors using different values of window size $L$, according to

$$\lambda_Q = [\vec{\varphi}(Q)_{L_1}, \vec{\varphi}(Q)_{L_2}, \ldots, \vec{\varphi}(Q_m)_{L_n}]. \tag{9}$$

The feature vector $\lambda_Q$ depends on the value of $Q$, which can provide different characteristics for each value. Thus, we can create a final feature vector $\Upsilon_{Q_1,Q_2,\ldots,Q_n}$ by concatenating the vectors $\lambda_Q$ for different values of $Q$, according to

$$\Upsilon_{Q_1,Q_2,\ldots,Q_n} = [\lambda_{Q_1}, \lambda_{Q_2}, \ldots, \lambda_{Q_n}]. \tag{10}$$

### 3.3. Computational complexity

Taking into account a dynamic texture $N \times N \times N$, let $I$ be one of its slices $N \times N$ and $W_i$ be a window $L \times L$. Considering that the core of our method is to solve the equation $\vec{f} = DZ^T(ZZ^T)^{-1}$, we can divide its analysis into some fundamental steps. First, the method sweeps $T_1 = (N-L+1)^2$ pixel positions to construct the matrices $X$ and $D$ from a slice $I$ using $W_i$. Next, computing $Z = \phi(WX)$ and transposing $Z$ require approximately $T_2 = Q(p+1)(N-L+1)^2 + Q(N-L+1)^2$ and $T_3 = (Q+1)(N-L+1)^2$ operations, respectively. Multiplying $DZ^T$, computing $ZZ^T$ and obtaining its inverse require roughly $T_4 = (N-L+1)^2(Q+1)$, $T_5 = (Q+1)^2(N-L+1)^2$ and $T_6 = (Q+1)^3$ operations, respectively (there are methods more efficient to compute $T_6$). Finally, multiplying $DZ^T$ by $(ZZ^T)^{-1}$ requires approximately $T_7 = (Q+1)^2$ operations. Thus, we can consider the total time to compute $\vec{f}$ as $T = \sum_{i=1}^{7} T_i = O(N^2)$, once $Q$, $L$ and $p$ are usually smaller than $N$ and do not depend on the number of pixels in a slice. Also, considering that $\vec{f}$ is computed for three window sizes in our method and that each dimension of the dynamic texture has $N$ slices, we can establish that our proposed signature has time complexity $3NO(N^2)$, that is, it is $O(N^3)$. It is worthwhile to mention that we suppressed some operations (Tikhonov regularization, computation of the randomized weights $W$ and of the average of the feature vectors in each orthogonal plane etc.) in order to simplify our demonstration, once they do not change our method's time complexity.

## 4. Experiment

In order to validate the proposed method and compare its efficiency to other ones in recognition tasks, the signatures were classified using 1-nearest neighbor (1-NN) with Euclidean distance. We have adopted this classifier due to its simplicity, thereby emphasizing the features obtained by the methods. An experimental setup similar to Ravichandran, Chaudhry, and Vidal (2009), Ghanem and Ahuja (2010), Tiwari and Tyagi (2016a) (for the UCLA-50, UCLA-9 and UCLA-8 databases) and Gonçalves, Machado, and Bruno (2015) (for the Dyntex++ database) was adopted for the evaluation of the proposed method. For the Dyntex++ and UCLA-50 databases, we divided them into test and training sets using the $k$−fold cross-validation scheme with 10−fold and 4−fold, respectively. For the UCLA-8 and UCLA-9 databases, we used half of them as the training set and the remainder for testing. This experiment was repeated 20 times and the average accuracy (ACC) and standard deviation of all trials were reported.

The dynamic textures databases used as benchmark to evaluate the proposed method were:

- Dyntex++ (Ghanem & Ahuja, 2010): This database is a compiled version of the Dyntex database (Péteri, Fazekas, & Huiskes, 2010). The videos of this database were preprocessed from their original form to evidence their representative dynamics. Thus, a single dynamic texture is shown in each video. This database consists of 3600 videos divided into 36 classes.
- UCLA (Saisan, Doretto, Wu, & Soatto, 2001): this database is a benchmark in the area of dynamic texture classification. It contains 200 videos separated into 50 different dynamic texture classes. Each video has 75 frames of 48 × 48 pixels. In addition to the original database with 50 classes (UCLA-50), we have used two different variations proposed in Ravichandran et al. (2009) for comparison with other methods. The first reorganizes the UCLA database to combine the videos taken from different viewpoints. Thus, the database is reduced to 9 classes (UCLA-9): smoke (4), boiling water (8), fire (8), sea (12), water (12), flower (12), waterfall (16), fountains (20), and plants (108). The value in parentheses is the number of samples per class. The second variation discards the videos of plant class, because this class far outnumbered the other classes. Therefore, this variation contains 8 classes (UCLA-8).

**Table 1**
Classification results for the feature vector $\lambda_Q$ with different values of $Q$.

| $Q$ | No of descriptors | ACC (%) | |
|---|---|---|---|
| | | UCLA | Dyntex++ |
| {09} | 90 | 96.50 ($\pm$1.80) | 96.71 ($\pm$0.84) |
| {19} | 180 | 96.70 ($\pm$2.00) | 96.84 ($\pm$0.87) |
| {29} | 270 | 97.55 ($\pm$1.60) | 96.64 ($\pm$0.96) |
| {39} | 360 | 97.35 ($\pm$1.89) | 96.67 ($\pm$0.85) |
| {49} | 450 | 97.85 ($\pm$1.59) | 96.49 ($\pm$0.97) |
| {59} | 540 | 97.50 ($\pm$1.96) | 96.38 ($\pm$0.97) |

**Table 2**
Comparison of the proposed method with feature vector $\Upsilon$ combining different values of $Q$.

| $\{Q_1, Q_2\}$ | No of descriptors | ACC (%) | |
|---|---|---|---|
| | | UCLA | Dyntex++ |
| {09,19} | 270 | 97.05 ($\pm$1.87) | 97.22 ($\pm$0.84) |
| {09,29} | 360 | 97.25 ($\pm$1.85) | 96.96 ($\pm$0.88) |
| {09,39} | 450 | 97.10 ($\pm$1.81) | 96.81 ($\pm$0.85) |
| {09,49} | 540 | 97.70 ($\pm$1.54) | 96.74 ($\pm$0.87) |
| {09,59} | 630 | 97.60 ($\pm$1.65) | 96.80 ($\pm$0.86) |
| {19,29} | 450 | 97.55 ($\pm$1.66) | 96.98 ($\pm$0.87) |
| {19,39} | 540 | 97.35 ($\pm$1.89) | 96.75 ($\pm$0.88) |
| {19,49} | 630 | 97.90 ($\pm$1.63) | 96.58 ($\pm$0.89) |
| {19,59} | 720 | 97.80 ($\pm$1.74) | 96.73 ($\pm$0.92) |
| {29,39} | 630 | 97.50 ($\pm$1.85) | 96.96 ($\pm$0.86) |
| {29,49} | 720 | 97.95 ($\pm$1.60) | 96.70 ($\pm$0.92) |
| {29,59} | 810 | 97.40 ($\pm$1.93) | 96.69 ($\pm$0.90) |
| {39,49} | 810 | 97.80 ($\pm$1.56) | 96.69 ($\pm$0.86) |
| {39,59} | 900 | 97.75 ($\pm$1.71) | 96.67 ($\pm$0.88) |
| {49,59} | 990 | 98.00 ($\pm$1.63) | 96.51 ($\pm$0.94) |

**Table 3**
Comparison on the UCLA-50 database (4-fold cross validation). The compared results were obtained from Tiwari and Tyagi (2016a) and Tiwari and Tyagi (2017).

| Method | ACC (%) |
|---|---|
| KDT-MD (Chan & Vasconcelos, 2007) | 89.50 |
| DFS (Xu, Quan, Ling, & Ji, 2011) | 89.50 |
| 3D-OTF (Xu, Huang, Ji, & Fermüller, 2012) | 87.10 |
| CVLBP (Tiwari & Tyagi, 2016b) | 93.00 |
| HLBP (Tiwari & Tyagi, 2016a) | 95.00 |
| MEWLSP (Tiwari & Tyagi, 2017) | 96.50 |
| LBP-TOP (Zhao & Pietikainen, 2007) | 94.50 |
| Proposed method ($\Upsilon_{9,19}$) | 97.05 ($\pm$1.87) |

## 5. Results and discussion

To choose an image signature, we performed several experiments with different values of $Q$ and 1-NN classifier (Euclidean distance) on the UCLA-50 and Dyntex++ datasets using 4-fold and 10-fold cross-validation schemes, respectively. For this, we used a sparse interval of $Q \in \{9, 19, \ldots, 59\}$ in order to increase the chance of finding a suitable number of hidden neurons. Thus, in order to perform a fair comparison with other dynamic texture analysis methods, we considered the results shown in Tables 1 and 2 and adopted the best set of values $Q$ of one database to classify another database. In this way, for the UCLA-8, UCLA-9 and UCLA-50 databases, we used the best parameter values of the Dyntex++ database ({09, 19}); and for the Dyntex++ database, we used the best parameter values of the UCLA-50 database ({49, 59}).

Table 3 shows the comparison of our proposed method on the UCLA-50 database using 4-fold cross validation. The results demonstrate that our signature obtains the highest average accuracy (97.05%). Also, the standard deviation of our success rate confirms that our method is very discriminative in this dataset, since the lower bound accuracy (97.05%−1.87% = 95.18%) is superior to the results of all the compared methods, except for MEWLSP. Moreover, it is worth stressing the relatively reduced number of descriptors of our method (270 features) when compared to the MEWLSP signature (1536 features).

**Table 4**
Comparison of the proposed method with other dynamic texture methods on the UCLA-9 and UCLA-8 databases (half of the samples for training and the remainder for testing). The compared results were obtained from Tiwari and Tyagi (2016a) and Tiwari and Tyagi (2017).

| Method | ACC (%) | |
|---|---|---|
| | UCLA-9 | UCLA-8 |
| 3D-OTF (Xu et al., 2012) | 96.32 | 95.80 |
| CVLBP (Tiwari & Tyagi, 2016b) | 96.90 | 95.65 |
| HLBP (Tiwari & Tyagi, 2016a) | 98.35 | 97.50 |
| MEWLSP (Tiwari & Tyagi, 2017) | 98.55 | 98.04 |
| MBSIF (Arashloo & Kittler, 2014) | 98.75 | 97.80 |
| High level feature (Wang & Hu, 2015) | 92.67 | 85.65 |
| DNGP (Rivera & Chae, 2015) | 98.10 | 97.00 |
| WMFS (Ji, Yang, Ling, & Xu, 2013) | 96.95 | 97.18 |
| Chaotic vector (Wang & Hu, 2016) | 85.10 | 85.00 |
| LBP-TOP (Zhao & Pietikainen, 2007) | 96.00 | 93.67 |
| Proposed method ($\Upsilon_{9,19}$) | 98.54 ($\pm$1.56) | 97.74 ($\pm$2.99) |

**Table 5**
Comparison of the proposed method and others on the Dyntex++ database (10-fold cross validation).

| Method | ACC (%) |
|---|---|
| RI-VLBP (Zhao & Pietikäinen, 2007) | 96.14 ($\pm$0.77) |
| LBP-TOP (Zhao & Pietikainen, 2007) | 97.72 ($\pm$0.43) |
| DPSW (Gonçalves & Bruno, 2013b) | 91.39 ($\pm$1.29) |
| CNDT (Gonçalves et al., 2015) | 83.86 ($\pm$1.40) |
| Proposed method ($\Upsilon_{49,59}$) | 96.51 ($\pm$0.94) |

Tables 4 shows the comparison of our proposed approach with other dynamic texture analysis methods in the variants of 9 classes and 8 classes of the UCLA database. The results shown in both the experiments demonstrate that our signature is among the most discriminative methods. For instance, on the UCLA-9 experiment, the highest accuracy is 98.75% (MBSIF method), which is within the interval of standard deviation of our approach (average accuracy of 98.54%, with $\pm$1.56% of standard deviation). Similarly, on the UCLA-8 experiment, the highest accuracy is 98.04% (MEWLSP method), which is again within the interval of standard deviation of our approach (average accuracy of 97.74%, with $\pm$2.99% of standard deviation). Moreover, when we consider the number of descriptors, our signature is very reduced when compared to the signature lengths of the aforementioned compared methods. For instance, our method's signature is 95.61% and 82.42% smaller than MBSIF (6144 features) and MEWLSP (1536 features) signatures, respectively.

Table 5 shows the comparison of our proposed signature against other dynamic texture analysis methods on the Dyntex++ database. In this experiment, our signature ($\Upsilon_{49,59}$) reached the second highest accuracy (96.51%) and has 28.91% more descriptors than the LBP-TOP signature of 768 features adopted for the Dyntex++. However, it is important to emphasize that all the other signature configurations for the Dyntex++ in Table 2 present higher success rates and less descriptors than those of $\Upsilon_{49,59}$. For instance, the signature $\Upsilon_{9,19}$ provides 97.22% of accuracy and has 64.84% less descriptors than the LBP-TOP signature.

Also, we performed an additional comparison with the results presented in the paper Andrearczyk and Whelan (2018), which presented accuracies of 99.50%, 98.35%, 99.02%, 98.58% on the UCLA-50, UCLA-9, UCLA-8 and Dyntex++, respectively, using Convolutional Neural Networks. For this, we used again the signatures $\Upsilon_{9,19}$ for the three UCLA datasets and $\Upsilon_{49,59}$ for the Dyntex++.

**Table 6**
Comparison of our proposed method using SVM and LDA classifiers with the CNN based approach proposed by Andrearczyk and Whelan (2018). (*) - results obtained from Andrearczyk and Whelan (2018).

| Database | SVM | LDA | CNN |
|---|---|---|---|
| UCLA-50 | 98.15 ($\pm$1.46) | 99.65 ($\pm$0.33) | 99.50* |
| UCLA-9 | 99.36 ($\pm$0.68) | 99.17 ($\pm$1.55) | 98.35* |
| UCLA-8 | 98.61 ($\pm$1.46) | 98.85 ($\pm$3.14) | 99.02* |
| Dyntex++ | 92.82 ($\pm$0.69) | 84.61 ($\pm$0.86) | 98.58* |

Also, we used the same validation procedure adopted in the aforementioned paper. Thus, because these results are higher than the success rates we obtained using 1-NN (except for the UCLA-9), we decided to classify our signatures using other two classifiers: Linear Discriminant Analysis - LDA, and Support Vector Machines - SVM (we used a polynomial SVM from Weka (Holmes, Donkin, & Witten, 1994) using class SMO, which implements *S*equential *M*inimal *O*ptimization algorithm (Platt, 1999), with the default parameter values of this class). Table 6 shows our obtained accuracies, which are slightly higher than that of the paper (Andrearczyk & Whelan, 2018) on the UCLA-9 using SVM, and on the UCLA-50 and UCLA-9 using LDA. Also, it is important to emphasize that our results on the UCLA-8 using LDA and SVM, even though slightly smaller than 99.02%, have intervals of standard deviation that reach this accuracy. This allows us to conclude that our performance on UCLA-8 is equivalent to that of the paper (Andrearczyk & Whelan, 2018). On the Dyntex++ database, however, our results were inferior, indicating that our proposed approach needs to be improved to extract more discriminative signatures from this dataset.

The proposed method took, on average, 0.14 s and 0.19 s to compute a signature from a single dynamic texture from the Dyntex++ and UCLA-50 databases, respectively. Also, our approach took, on average, 0.004 s (UCLA-50) and 0.61 s (Dyntex++) to classify the feature vectors from the whole databases using the 1-NN with cross-validation. In these experiments, we used a 3.60 GHz Intel(R) Core i7, 64GB RAM and 64-bit Operating System. The results demonstrate that our proposed signature is built in a reasonable time, considering that dynamic textures from the UCLA-50 and Dyntex++ databases have $48 \times 48 \times 75$ and $50 \times 50 \times 50$ pixels, respectively. The time for classification is also efficient in the both datasets.

Finally, we would like to comment some aspects of our proposed method. First, it used a randomized neural network with offline learning, but we think that it does not limit it since there are works that extend this kind of neural network to online learning (for instance, Pratama et al., 2017; Pratama et al., 2018). This suggests that our proposed method could be applied in real-time applications. Second, we believe that our proposed approach has two advantages when compared to CNNs: 1 - CNNs require a large number of samples to be trained. In our approach, it is not a drawback, once each "image" is the source of the training set and even small images provide large training datasets (for instance, an image $50 \times 50$ provides a training set of $48 \times 48 = 2304$ (using a window $3 \times 3$) input feature vectors and respective labels); 2 - Our proposed descriptors can be used in many classifiers, which can be chosen based on several criteria (speediness, simplicity, robustness etc.).

Thus, in the light of the high accuracies obtained by our method, its feature vectors with reduced number of descriptors, and its relatively low time complexity, we can affirm that our proposed signature has good performance in these three important aspects and, therefore, is comparable to the most discriminative state-of-the-art methods present in the literature.

## 6. Conclusion

This paper presented a highly discriminative dynamic texture analysis signature based on the weights of the output layer of a randomized neural network after using the pixels of a video as input feature vectors and corresponding labels. The obtained results are very promising, since they are among the highest success rates obtained in four video benchmarks. Also, some of our results were obtained with a relatively small number of descriptors when compared to other methods evaluated in this work. Thus, based on this performance, we can conclude that our proposed approach provides a powerful tool to recognize dynamic textures and, therefore, opens a rich line of research in computer vision. As future works, we intend to exploit different architectures for the neural network and new ways of building datasets for training it. Also, an interesting line of research is to combine our proposed method with other approaches, such as complex networks, fractal dimension, local binary patterns and so on.

## Declaration of Competing Interest

There is no conflict of interest.

## Credit authorship contribution statement

**Jarbas Joaci de Mesquita Sá Junior:** Conceptualization, Formal analysis, Investigation, Methodology, Funding acquisition, Validation, Writing - original draft, Writing - review & editing. **Lucas Correia Ribas:** Conceptualization, Formal analysis, Investigation, Methodology, Funding acquisition, Validation, Writing - original draft, Writing - review & editing. **Odemir Martinez Bruno:** Funding acquisition, Project administration, Resources, Software, Supervision, Writing - original draft, Writing - review & editing.

## Acknowledgments

Jarbas Joaci de Mesquita Sá Junior thanks CNPq (National Council for Scientific and Technological Development, Brazil) (Grant: 302183/2017-5) for the financial support of this work. Lucas Correia Ribas gratefully acknowledges the financial support grant #2016/23763-8, São Paulo Research Foundation (FAPESP). Odemir Martinez Bruno gratefully acknowledges the financial support of CNPq (307797/2014-7 and 484312/2013-8) and FAPESP (14/08026-1 and 2016/18809-9).

## References

Andrearczyk, V., & Whelan, P. F. (2018). Convolutional neural network on three orthogonal planes for dynamic texture classification. *Pattern Recognition, 76*, 36–49.
Arashloo, S. R., & Kittler, J. (2014). Dynamic texture recognition using multiscale binarized statistical image features. *IEEE Transactions on Multimedia, 16*(8), 2099–2109.
Bacciu, D., Colombo, M., Morelli, D., & Plans, D. (2018). Randomized neural networks for preference learning with physiological data. *Neurocomputing, 298*, 9–20.
Brieu, N., Serbanovic-Canic, J., Cvejic, A., Stemple, D., Ouwehand, W., Navab, N., et al. (2010). Thrombus segmentation by texture dynamics from microscopic image sequences. *SPIE medical imaging*. International Society for Optics and Photonics. 76233Z–76233Z.
Calvetti, D., Morigi, S., Reichel, L., & Sgallari, F. (2000). Tikhonov regularization and the L-curve for large discrete ill-posed problems. *Journal of Computational and Applied Mathematics, 123*(1), 423–446.
Chan, A., & Vasconcelos, N. (2008). Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30*(5), 909–926.
Chan, A. B., Morrow, M., & Vasconcelos, N. (2009). Analysis of crowded scenes using holistic properties. In *Performance evaluation of tracking and surveillance workshop at CVPR* (pp. 101–108).
Chan, A. B., & Vasconcelos, N. (2005). Classification and retrieval of traffic video using auto-regressive stochastic processes. In *Proceedings. IEEE intelligent vehicles symposium* (pp. 771–776).

Chan, A. B., & Vasconcelos, N. (2007). Classifying video with kernel dynamic textures. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1–6).

Derpanis, K. G., & Wildes, R. P. (2011). Classification of traffic video based on a spatiotemporal orientation analysis. In *Ieee workshop on applications of computer vision (wacv)* (pp. 606–613).

Doretto, G., Chiuso, A., Wu, Y., & Soatto, S. (2003). Dynamic textures. *International Journal of Computer Vision, 51*(2), 91–109.

Dubois, S., Péteri, R., & Ménard, M. (2009). A comparison of wavelet based spatio-temporal decomposition methods for dynamic texture recognition. In H. Araujo, A. M. Mendonça, A. J. Pinho, & M. I. Torres (Eds.), *4th iberian conference on pattern recognition and image analysis (IBPRIA 2009)*. In *Lecture Notes in Computer Science (LNCS): 5524* (pp. 314–321). Springer.

Dudek, G. (2019). Generating random weights and biases in feedforward neural networks with random hidden nodes. *Information Sciences, 481*, 33–56.

Fazekas, S., & Chetverikov, D. (2007). Dynamic texture recognition using optical flow features and temporal periodicity. In *International workshop on content-based multimedia indexing (CBMI)* (pp. 25–32).

Ghanem, B., & Ahuja, N. (2010). Maximum margin distance learning for dynamic texture recognition. In *Proceedings of the 11th European conference on computer vision: Part II*. In *ECCV'10* (pp. 223–236). Berlin, Heidelberg: Springer-Verlag.

Gonçalves, W. N., & Bruno, O. M. (2013a). Combining fractal and deterministic walkers for texture analysis and classification. *Pattern Recognition, 46*(11), 2953–2968.

Gonçalves, W. N., & Bruno, O. M. (2013b). Dynamic texture analysis and segmentation using deterministic partially self-avoiding walks. *Expert Systems with Applications, 40*(11), 4283–4300.

Gonçalves, W. N., & Bruno, O. M. (2013c). Dynamic texture segmentation based on deterministic partially self-avoiding walks. *Computer Vision and Image Understanding, 117*(9), 1163–1174.

Gonçalves, W. N., Machado, B. B., & Bruno, O. M. (2011). Spatiotemporal Gabor filters: A new method for dynamic texture recognition. In *Proceedings of the VII workshop de visão computacional* (pp. 184–189). Curitiba, Brazil.

Gonçalves, W. N., Machado, B. B., & Bruno, O. M. (2015). A complex network approach for dynamic texture recognition. *Neurocomputing, 153*, 211–220.

Holmes, G., Donkin, A., & Witten, I. H. (1994). WEKA: A machine learning workbench. In *Proceedings of anziis '94 - Australian New Zealand intelligent information systems conference* (pp. 357–361).

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing, 70*(1), 489–501.

Ji, H., Yang, X., Ling, H., & Xu, Y. (2013). Wavelet domain multifractal analysis for static and dynamic texture classification. *IEEE Transactions on Image Processing, 22*(1), 286–299.

Kellokumpu, V., Zhao, G., & Pietikäinen, M. (2008). Human activity recognition using a dynamic texture based method. In *British machine vision conference (bmvc): 1* (pp. 88.1–88.10).

Lehmer, D. H. (1951). Mathematical methods in large scale computing units. *Annals of Computation Laboratory of Harvard University, 26*, 141–146.

Moore, E. H. (1920). On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society, 26*, 394–395.

Otsuka, K., Horikoshi, T., Suzuki, S., & Fujii, M. (1998). Feature extraction of temporal texture based on spatiotemporal motion trajectory. In *Proceedings. fourteenth international conference on pattern recognition: 2* (pp. 1047–1051). IEEE.

Pao, Y.-H., Park, G.-H., & Sobajic, D. J. (1994). Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing, 6*(2), 163–180.

Pao, Y.-H., & Takefuji, Y. (1992). Functional-link net computing: Theory, system architecture, and functionalities. *Computer, 25*(5), 76–79.

Park, S. K., & Miller, K. W. (1988). Random number generators: Good ones are hard to find. *Communications of the ACM, 31*(10), 1192–1201.

Penrose, R. (1955). A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society, 51*(3), 406–413.

Péteri, R., & Chetverikov, D. (2005). Dynamic texture recognition using normal flow and texture regularity. In *Pattern recognition and image analysis* (pp. 223–230). Springer.

Péteri, R., Fazekas, S., & Huiskes, M. J. (2010). Dyntex: A comprehensive database of dynamic textures. *Pattern Recognition Letters, 31*(12), 1627–1632.

Platt, J. C. (1999). *Advances in kernel methods* (pp. 185–208). Cambridge, MA, USA: MIT Press.

Polana, R., & Nelson, R. (1997). Temporal texture and activity recognition. In M. Shah, & R. Jain (Eds.), *Motion-based recognition*. In *Computational Imaging and Vision: 9* (pp. 87–124). Springer Netherlands.

Pratama, M., Angelov, P. P., Lu, J., Lughofer, E., Seera, M., & Lim, C. P. (2017). A randomized neural network for data streams. In *2017 international joint conference on neural networks (IJCNN)* (pp. 3423–3430).

Pratama, M., Angelov, P. P., Lughofer, E., & Er, M. J. (2018). Parsimonious random vector functional link network for data streams. *Information Sciences, 430–431*, 519–537.

Qiao, Y., & Weng, L. (2015). Hidden markov model based dynamic texture classification. *IEEE Signal Processing Letters, 22*(4), 509–512.

Ravichandran, A., Chaudhry, R., & Vidal, R. (2009). View-invariant dynamic texture recognition using a bag of dynamical systems. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 1651–1657).

Rivera, A. R., & Chae, O. (2015). Spatiotemporal directional number transitional graph for dynamic texture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 37*(10), 2146–2152.

Sá Junior, J. J. M., & Backes, A. R. (2016). ELM based signature for texture classification. *Pattern Recognition, 51*, 395–401.

Saisan, P., Doretto, G., Wu, Y. N., & Soatto, S. (2001). Dynamic texture recognition. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, vol.2: 2*.

Schmidt, W. F., Kraaijveld, M. A., & Duin, R. P. W. (1992). Feedforward neural networks with random weights. In *Proceedings., 11th IAPR international conference on pattern recognition. vol.II. Conference B: Pattern recognition methodology and systems* (pp. 1–4).

Soygaonkar, P., Paygude, S., & Vyas, V. (2015). *Proceedings of the 3rd international conference on frontiers of intelligent computing: Theory and applications (ficta) 2014: Volume 2* (pp. 281–288). Cham: Springer International Publishing.

Tikhonov, A. N. (1963). On the solution of ill-posed problems and the method of regularization. *Doklady Akademii Nauk USSR, 151*(3), 501–504.

Tiwari, D., & Tyagi, V. (2016a). A novel scheme based on local binary pattern for dynamic texture recognition. *Computer Vision and Image Understanding, 150*, 58–65.

Tiwari, D., & Tyagi, V. (2016b). Dynamic texture recognition based on completed volume local binary pattern. *Multidimensional Systems and Signal Processing, 27*(2), 563–575.

Tiwari, R. D., & Tyagi, V. (2017). Dynamic texture recognition using multiresolution edge-weighted local structure pattern. *Computers & Electrical Engineering, 62*, 485–498.

Wang, Y., & Hu, S. (2015). Exploiting high level feature for dynamic textures recognition. *Neurocomputing, 154*, 217–224.

Wang, Y., & Hu, S. (2016). Chaotic features for dynamic textures recognition. *Soft Computing, 20*(5), 1977–1989.

Xu, Y., Huang, S., Ji, H., & Fermüller, C. (2012). Scale-space texture description on SIFT-like textons. *Computer Vision and Image Understanding, 116*(9), 999–1013.

Xu, Y., Quan, Y., Ling, H., & Ji, H. (2011). Dynamic texture classification using dynamic fractal analysis. In *2011 international conference on computer vision* (pp. 1219–1226).

Yang, F., Xia, G.-S., Liu, G., Zhang, L., & Huang, X. (2016). Dynamic texture recognition by aggregating spatial and temporal features via ensemble SVMs. *Neurocomputing, 173, Part 3*, 1310–1321.

Zhang, L., & Suganthan, P. (2016). A comprehensive evaluation of random vector functional link networks. *Information Sciences, 367–368*, 1094–1105.

Zhang, Y., Wu, J., Cai, Z., Du, B., & Yu, P. S. (2019). An unsupervised parameter learning model for RVFL neural network. *Neural Networks*. doi:10.1016/j.neunet.2019.01.007.

Zhao, G., & Pietikainen, M. (2007). Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(6), 915–928.

Zhao, G., & Pietikäinen, M. (2007). *Dynamical vision: ICCV 2005 and ECCV 2006 workshops, WDV 2005 and WDV 2006, Beijing, China, October 21, 2005, Graz, Austria, May 13, 2006*. (pp. 165–177)). Berlin, Heidelberg: Springer Berlin Heidelberg. Revised papers.