Improving the descriptors extracted from the co-occurrence matrix using preprocessing approaches

Loris Nanni\textsuperscript{a,}\textsuperscript{*}, Sheryl Brahnam\textsuperscript{b}, Stefano Ghidoni\textsuperscript{a}, Emanuele Menegatti\textsuperscript{a}

\textsuperscript{a} Department of Information Engineering, University of Padua, Via Gradenigo, 6, 35131 Padova, Italy
\textsuperscript{b} Department of Computer Information Systems, Missouri State University, 901 S. National, Springfield, MO 65804, USA

\section*{ARTICLE INFO}
\textbf{Keywords:}
Co-occurrence matrix
Texture descriptors
Preprocessing
Local approaches
Multi-scale

\section*{ABSTRACT}
In this paper, we investigate the effects that different preprocessing techniques have on the performance of features extracted from Haralick’s co-occurrence matrix, one of the best known methods for analyzing image texture. In addition, we compare and combine different strategies for extracting descriptors from the co-occurrence matrix. We propose an ensemble of different preprocessing methods, where, for each descriptor, a given Support Vector Machine (SVM) classifier is trained. The set of classifiers is then combined by weighted sum rule. The best result is obtained by combining the extracted descriptors using the following preprocessing methods: wavelet decomposition, local phase quantization, orientation, and the Weber law descriptor. Texture descriptors are extracted from the entire co-occurrence matrix, as well as from sub-windows, and evaluated at multiple scales. We validate our approach on eleven image datasets representing different image classification problems using the Wilcoxon signed rank test. Results show that our approach improves the performance of standard methods. All source code for the approaches tested in this paper will be available at: https://www.dei.unipd.it/node/2357

© 2015 Elsevier Ltd. All rights reserved.

\section*{1. Introduction}
Medical imaging is experiencing rapid technological developments and, as a result, is producing unprecedented amounts of data that are now being warehoused in specialized medical research databases around the world\textsuperscript{1}. It is generally agreed that these databases contain information that, if appropriately harnessed, could potentially revolutionize scientific knowledge in medicine. What is needed to make this dream a reality is the development of more sophisticated and robust machine vision algorithms.

Texture analysis is essential for many image classification tasks. Some of the best-performing methods rely on features that encapsulate specific information about the elementary characteristics inherent in a given input image. A feature extractor algorithm obtains a vector of numbers, called a “descriptor”, from an image that provides a specific description of that image. Features that are widely used in the computer vision and robotics communities include the Scale-Invariant Feature Transform (SIFT) (Lowe, 2004), Speeded Up Robust Feature (SURF) (Bay, Tuytelaars, & Gool, 2006), Histogram of Oriented Gradients (HOG) (Dalal & Triggs, 2005), Gradient Location and Orientation Histogram (GLOH) (Mikolajczyk & Schmid, 2005), Region Covariance Matrix (RCM) (Tuzel, Porikli, & Meer, 2006), Edgelet (Wu & Nevatia, 2005), Gray Level Co-occurrence Matrix (GLCM) (Haralick, Shanmugam, & Dinstein, 1973), Local Binary Patterns (LBP) (Ojala, Pietikäinen, & Harwood, 1996), non-binary encodings (Paci, Nanni, Lathi, Aalto-Setälä, & Hyttinen, 2013), Color CorreloGram (CCG) (Huang, Kumar, Mitra, Zhu, & Zabih, 1997), Color Coherence Vectors (CCV) (Pass, Zabih, & Miller, 1996), color indexing (Swain & Ballard, 1991), steerable filters (Freeman & Adelson, 1991), and Gabor filters (Jain & Farrokhnia, 1991).

An especially powerful feature extractor is the Local Binary Pattern (LBP) operator (Ojala, Pietikäinen, & Maenpää, 2002), which stands out because of its simplicity, effectiveness, and robustness. It has proven to be a powerful discriminator in a number of image classification problems, e.g., in discriminating endoscopic images of pediatric celiac diseases (Vécsei, Aman, Hegenbart, Liedlgruber, & Uhl, 2011) and in distinguishing real masses from normal parenchyma in mammographic images (Oliver, Lladó, Freixenet, & Martí, 2007). LBP is also useful in data mining, where it has been combined with other descriptors, for instance, to classify brain magnetic resonance data (Unay & Ekin, 2008) and different cell phenotypes (Nanni & Lumini, 2008).

http://dx.doi.org/10.1016/j.eswa.2015.07.055
This paper is a continuation of our research group’s previous explorations (Nanni, Brahnam, Ghidoni, & Menegatti, 2013d; Nanni, Brahnam, Ghidoni, Menegatti, & Barrier, 2013b; Nanni, Ghidoni, & Menegatti, 2013c; Nanni, Paci, Brahnam, Ghidoni, & Menegatti, 2013a) for improving one of the first methods for analyzing texture, the GLCM, which was first proposed by Haralick (1979) for analyzing satellite images. The GLCM is one of the most studied and most extensively used texture analysis techniques: its application ranges from visual inspection for industrial production (Zhao et al., 2014) to analysis of microscopy imagery (Pantic et al., 2014). An alternative version of the co-occurrence matrix capable of dealing with color is proposed in Subrahmanym, Wu, Maheshwari, and Balasubramanian (2013); the formulation they offer takes into account the inter-correlation between the different color channels of a RGB (Red-Green-Blue) color image.

The GLCM is usually analyzed by means of a set of descriptors, commonly referred to as Haralick descriptors, that are evaluated by means of a process that provides a single value for each single descriptor starting from the whole matrix (as it happens when mean and variance are evaluated from a probability distribution). Within the last few years, the GLCM has become the focus of several research groups whose aim is to discover novel methods for extracting yet more information from the GLCM. Some important work in this area includes that of Gelzinis, Verikas, & Bacauskiene (2007), where different values of the distance parameter influencing the GLCM are considered; (Walker, Jackway, & Longstaff, 2003), where features are extracted from areas presenting high discrimination using the weighted summation of GLCM elements; (Chen, Chengdong, Chen, & Tan, 2009), where descriptors are obtained from the gray level-gradient co-occurrence matrix by calculating the gradient value of each pixel in the neighborhood of interest points; and (Mitrea, Nedevschi, Badea, & Lupson, 2012), where the GLCM is combined with the edge orientation co-occurrence matrix of superior order (see, Akono, Tankam, & Tonye, 2005) by taking into consideration the gray levels of the image pixels and some local features such as edges. In addition, multi-scale analysis has been performed using the GLCM, where descriptors are extracted from each rescaling operation. For instance, in Rakwatin, Longepe, Isoguchi, Shimada, and Uryu (2010) the image is rescaled to different sizes, and in Hu, (2009) and Pacifici and Chini (2008) multiple scales are investigated by changing the window size used to extract GLCM descriptors. In some very recent work, LBP has been combined with the GLCM. In Sastry, Kumar, Rao, Mallika, and Lakshminarayana (2012), for instance, the GLCM is constructed after applying LBP to the image, and features are extracted from second-order statistical parameters from the resulting matrix. In Sun, Wang, Chen, She, and Kong (2012), GLCM is constructed from an LBP image produced after the application of a Gaussian filtering pyramid preprocessing step.

A major difficulty encountered when analyzing texture is the strong dependency some texture analysis methods have on image resolution and scale. Several researchers have recently reported the effectiveness of combining more than one multi-scale approach along with LBP descriptors to overcome this problem (see, for instance, Ren, Wang, & Zhao, 2012). In Nanni et al. (2013c) and Nanni et al. (2013d), our group applied this idea to the co-occurrence matrix, demonstrating that results can be enhanced when different descriptors extracted from the co-occurrence matrix are combined with a multi-scale approach.

In addition to combine the GLCM with texture descriptors and the multi-scale approach, our main intention in this paper is to compare and to combine (for the first time) the application of different preprocessing techniques to each image before calculating the co-occurrence matrix. An extensive study comparing and combining preprocessing methods with the GLCM is important in order to clarify which combinations of methods offer the best performance gain using a system based on the co-occurrence matrix. The different preprocessing methods explored in this paper are detailed in Section 3. Our best result is obtained by combining extracted GLCM descriptors using the following preprocessing methods: wavelet decomposition, local phase quantization, orientation, and the Weber law descriptor.

As in our previous work (Nanni et al., 2013b) (which did not include preprocessing methods), we utilize the following methods for extracting features:

- The standard approach proposed by Haralick (1979);
- A set of 3D features that consider the co-occurrence matrix as a 3D shape (Nanni et al., 2013c);
- Gray-level run-length features (Tang, 1998), that consider groups of consecutive pixels with the same gray level, and measure their lengths. In this paper, the gray-level run-length features (GL) are evaluated on the GLCM, and it is important to distinguish GL from the GLCM. As noted in Section 2.1, the GLCMs are calculated at different values of $d$ (the distance between the two pixels that are compared) and $\theta$ (the direction they are aligned). As described in more detail in Section 2.4, the GL approach has its own orientation (labelled $\theta_{GL}$). For each GLCM, the GL descriptor is extracted for all values of $\theta_{GL}$.

We also examine for the first time a novel approach for representing the co-occurrence matrix using Spatial Diversity (SD) features (Geraldo Junior, da Rocha, Gattass, Silva, & de Paiva, 2013). SD works poorly when applied to the whole co-occurrence matrix, but performs quite well when applied to a set of sub-windows extracted from the co-occurrence matrix, i.e., the GLCM considered only on a subset of its $256 \times 256$ domain. This is an important point, as it demonstrates that multiple descriptors, if appropriately applied, can be useful in describing the co-occurrence matrix: this, in turn, demonstrates our idea that the co-occurrence matrix can be considered as a texture image. Hence, it is possible to apply standard texture descriptors (e.g., SD, GL, …) to the GLCM to extract information.

In this study we also improve the performance of the feature extraction algorithms by extracting features from the co-occurrence matrix applied to the original image as well as to the image after a gaussian filtering at multiple scales. The co-occurrence matrix is also divided into different sub-windows, from which feature sets are also extracted. Separate SVM classifiers are trained, one for each sub-window. The overall classification outcome is obtained by combining the separate SVMs decisions using either a sum rule or a weighted sum rule.

In Nanni et al. (2013b) only the standard approach proposed by Haralick (1979) and a set of 3D features considering the co-occurrence matrix as a 3D shape are used for describing the GLCM. In this work we test four different approaches for describing the co-occurrence matrix, adding as well the innovation of combining each approach with a set of preprocessing methods. Moreover, in our previous work, our approach was validated only on five datasets, whereas in this study we employ a set of eleven datasets.

The remainder of this paper is organized as follows. The feature extractors used in our experiments are described in Section 2, and the preprocessing methods (and resulting processed images) are detailed in Section 3. The eleven datasets used to evaluate our approach and our rationale for using them are presented in Section 4, and the experimental results are reported and discussed in Section 5. Finally, we conclude in Section 6 with some remarks on the limitations of this study; we also highlight the importance of studying the GLCM and offer some suggestions for future investigations.

2. Feature extractors

As mentioned in the introduction, in Nanni et al. (2013c, 2013d), we showed how it is possible to enhance the performance of Haralick’s features by applying the multi-scale approach. We also demonstrated the effectiveness of combining features extracted from
different sub-windows of the co-occurrence matrix. In brief, the multi-scale approach generates images by applying a 2D symmetric Gaussian lowpass filter of kernel size $k$ (we use $k = 3$ and $k = 5$ in this work) with standard deviation $1$.

Described in this section are each of the feature extraction approaches used in the experiments reported in Section 5. In all experiments, SVM (Vapnik & Chervonenkis, 1964; Vapnik & Lerner, 1963), using both linear and radial basis function kernels, is the base classifier. The best kernel and the best set of parameters for each dataset are chosen using a 5-fold cross-validation approach on the training data. The parameters of the fusions are chosen based on criteria that will be detailed below and that were empirically discovered to maximize the performance obtained by the descriptors extracted from the whole image and from the sub-windows. It should be noted that in our experiments the same weights are always used without any ad-hoc dataset tuning.

2.1. The gray level dependency matrix (GLCM)

The Gray Level Co-occurrence Matrix (GLCM) (Haralick, 1979) is a matrix that is obtained as a histogram on a 2D domain of dimension $M \times M$, where $M$ is the number of gray levels in the image (in our case equal to 256). The GLCM counts the number of pixel transitions between two pixel values. In other words, the bin of the histogram is incremented whose indices are equal to the values of the two pixels. For example, a pixel transition between two adjacent pixels from 20 to 30 will increment the GLCM matrix at indices (20, 30). Since the pixels of the input image take values in the range (0, 255), the gray level dependency matrix has size 256×256; each bin at indices (also referred to as coordinates) $(i, j)$ represents the number of pixel couples showing the transition from gray level $i$ to gray level $j$.

The way pixel couples are created depends on two parameters: $d$ (the distance between the two pixels) and $\theta$ (the direction they are aligned). A value of $d = 1$ and $\theta = 0$, for instance, would result in pixel couples that are adjacent to each other on the same row. In our experiments, we consider four directions: the horizontal (H), the vertical (V), the diagonal top-left-bottom-right, or right-down (RD), and the top-right-bottom left, or left-down (LD).

In this study the GLCM is processed both by considering the whole matrix, as well as by restricting to some subsets of its domain (called sub-window): this means that only a subset of the pixel transitions are considered at a time.

2.2. The standard Haralick statistics (HAR)

Haralick (1979) was the first to consider the idea of using a set of features to extract information from the co-occurrence matrix. The following Haralick features are considered in our tests:

1. Energy
2. Correlation
3. Inertia
4. Entropy
5. Inverse difference moment
6. Sum average
7. Sum variance
8. Sum entropy
9. Difference average
10. Difference variance
11. Difference entropy
12. Information measure of correlation 1
13. Information measure of correlation 2

In HAR a set of 13 features is calculated from each co-occurrence matrix evaluated at $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ and $d = \{1, 3\}$. A global descriptor is obtained by concatenating the descriptors obtained for each distance and orientation value.

We also evaluated the performance of the following HAR variants:

- **HR**, in which features are extracted from the entire image only;
- **HRsub**, in which a feature set is extracted from the entire co-occurrence matrix plus from four sub-windows defined using the coordinates (0, 0) to (127, 127); (128, 128) to (255, 255), (0, 128) to (255, 128); and (128, 0) to (255, 255); these are the coordinates that provided the best experimental results in Nanni et al. (2013d); each set of extracted features is then used to train a separate SVM, with the results combined by weighted sum rule. The weights are 1 for each SVM running on the feature evaluated on a sub-window, and 4 for the SVM analyzing the data extracted from the whole image: this choice reflects the fact that each sub-window has an area that is $\frac{1}{4}$ the area of the entire co-occurrence matrix;
- **HRsca**, in which features are extracted from the original image and two filtered images. In other words, the feature sets are extracted from the entire co-occurrence matrix and from the four sub-windows defined in HRsub as well as from the original image and from the two filtered images. In this way 15 sets are obtained, and the 15 SVMs are combined by weighted sum rule.

2.3. Shape

This approach explores the shape of the co-occurrence matrix by considering it as a 3D function. SHAPE has been described in detail in Ghidoni, Cierniak, and Menegatti (2012); Nanni et al. (2013d). The main idea behind SHAPE is to intersect the GLCM with a set of horizontal planes defined by some given height values (see Fig. 1) to obtain a set of level curves (which are curves that describe the shape of the 3D GLCM at given height values). These level curves are used to derive a set of features.

Each level curve can be composed of more than one blob. The shape shown in Fig. 1 is the ideal configuration, but spikes can occur towards the edge of the histogram. To filter these spikes out so that only the main component that is representative of the overall texture of the image is considered, the main blob (the one with the largest area) is selected for the feature extraction process. To ease the analysis, it is approximated by fitting it to an ellipse (with only a small loss of information).

Starting at the base of the co-occurrence matrix with height 1 and going up by a distance of 2, level curves are progressively considered until height 19. The maximum height value of 19 is based on experimental observation of the effects of the different level curves. Level curves that have the most information lie at a relatively low height because the lower regions of the co-occurrence matrix are
very stable, unlike the upper regions, which are less stable due to image noise. To understand the impact of noise in the upper regions, consider input images of similar texture content, say two consecutive frames of a video in which no perceivable changes occur; the level curves evaluated on the two images, or video frames, would show more significant difference when in fact there is very little difference in the two images because of the noise in the higher level curves. This is an unwanted influence of the noise affecting the image pixel. The exact size chosen for the height values also depends on the image size. The values mentioned above were chosen for images of size $640 \times 480$, but they are suitable also for smaller and larger images, as shown in the experimental section. Finally, it should be mentioned that the co-occurrence matrix is not normalized because normalization to the highest bin would introduce instabilities. Certainly, it is the case that other types of normalization could be performed with respect to the total volume of the co-occurrence matrix; however, this volume is actually the total number of pixel couples in the image and depends on the image dimensions, which is constant for a given video flow. Normalizing using a constant would just scale the values, making the normalization irrelevant.

A set of features extracted from the ellipses obtained as described above is evaluated at each level. The features describing all levels are then analyzed together, and a set of nine features are extracted that describe the evolution of the level curves (see Ghidoni et al., 2012, and Nanni et al., 2013d, for details). These features provide a characterization of the input image that can be directly used as the input to a classifier.

The features in SHAPE, as defined in this paper, are evaluated on the entire co-occurrence matrix as well as on 12 sub-windows ($1-12$) of the GLCM, which are defined by the following index ranges: $\#1$: (0, 0) to (127, 127), $\#2$: (128, 128) to (255, 255), $\#3$: (0, 0) to (191, 191), $\#4$: (64, 64) to (255, 255), $\#5$: (0, 0) to (55, 55), $\#6$: (31, 31) to (95, 95), $\#7$: (63, 63) to (127, 127), $\#8$: (55, 55) to (159, 159), $\#9$: (127, 127) to (191, 191), $\#10$: (159, 159) to (223, 223), $\#11$: (191, 191) to (255, 255), and $\#12$: (63, 63) to (191, 191). In SHAPE we extract more sub-windows than the four previously used because, differently from that case, the performance of SHAPE improves when smaller sub-windows are used, as described in (Nanni et al., 2013c). The division of sub-windows follows the same process carried out in Nanni et al. (2013c), which proved to be the most effective solution when dealing with image classification based on Haralick's features.

For each of these 13 windows (counting the whole GLCM and the 12 sub-windows), a different feature vector is extracted and used as the input for a separate SVM. Results of the 13 SVMs are combined by weighted sum rule. A weight of 1 is given to the first five descriptors and a weight of 0.5 is assigned to the remainder.

Each set of 13 descriptors is derived from the co-occurrence matrices evaluated at $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ and $d = \{1, 3\}$ (in the case where multiple values are used for the distance). The feature vector is the concatenation of the features extracted for each value of the distance. In the experimental section, $SH$ denotes the case where features are extracted only from the entire co-occurrence matrix. $SH_{sub}$ denotes the method that extracts features from all 13 windows. $SH_{sub}^a$ is the approach that combines $SH_{sub}$ with the multi-scale approach.

### 2.4. Gray-level run-length features (GL)

GL (Tang, 1998) derives features from a run-length matrix that is calculated using the characteristics of gray level runs (a set of consecutive pixels with the same value). The run length is determined by the number of pixels in the set. Accordingly, the run-length matrix $P$ contains in each location $P(i,j)$ the number of runs of length $j$ of a given gray level $i$. The run length can be evaluated following four different orientations (horizontal, vertical and diagonal), described by means of the angle $\theta_{GL} = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$.

In this work, the run-length matrix is used in a slightly different way: instead of measuring the pixel gray levels of a given image, it is employed to measure the height of the bins of the GLCM; in other words, the GLCM is considered by the run-length evaluation algorithm like an image of size $256 \times 256$ whose “pixels” (namely the bins of the histogram) can have a value above 256.

The run-length matrix is a tool capable of evaluating the uniformity of the images, that is, the areas showing the same gray level. Such tool is employed in this study to measure this aspect of the GLCM histograms: the run-level matrix is a complementary aspect with respect to the SHAPE features described in Section 2.3: while the former is focused on the uniform regions, the latter is focused on the variations of the GLCM peaks.

It is possible to evaluate several features from the run-length matrix; in our experiments, we consider the following ones, described in (Tang, 1998):

- **Short Run Emphasis (SRE)**, evaluated as:

  $$SRE = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{P(i,j)}{j^2}$$  \hspace{1cm} (1)

  - where $n_r$ is the total number of runs, $M$ the number of gray levels, and $N$ the maximum run length;

- **Long Run Emphasis (LRE)**:

  $$LRE = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} P(i,j) \cdot j^2$$  \hspace{1cm} (2)

- **Gray Level Nonuniformity (GLN)**:

  $$GLN = \frac{1}{n_r} \sum_{i=1}^{M} \left( \sum_{j=1}^{N} P(i,j) \right)^2$$  \hspace{1cm} (3)

- **Run Length Nonuniformity (RLN)**:

  $$RLN = \frac{1}{n_r} \sum_{i=1}^{M} \left( \sum_{j=1}^{N} P(i,j) \right)^2$$  \hspace{1cm} (4)

- **Run Percentage (RP)**:

  $$RP = \frac{n_p}{n_r}$$  \hspace{1cm} (5)

where $n_p$ is the total number of pixels in the image;

- **Low Grey-Level Run Emphasis (LGRE)**:

  $$LGRE = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{P(i,j)}{i^2}$$  \hspace{1cm} (6)

- **High Grey-Level Run Emphasis (HGRE)**:

  $$HGRE = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{P(i,j)}{j^2}$$  \hspace{1cm} (7)

- **Short Run Low Grey Level Emphasis (SRLGE)**:

  $$SRLGE = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{P(i,j)}{i^2}$$  \hspace{1cm} (8)

- **Short Run High Grey Level Emphasis (SRHGE)**:

  $$SRHGE = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{P(i,j)}{j^2}$$  \hspace{1cm} (9)

- **Long Run Low Grey Level Emphasis (LRLGE)**:

  $$LRLGE = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{P(i,j)}{i^2}$$  \hspace{1cm} (10)
• Long Run High Grey Level Emphasis (LRHGE):

\[
LRHGE = \frac{1}{N^2} \sum_{i=1}^{M} \sum_{j=1}^{N} P(i, j) \cdot i^2 \cdot j^2 \tag{11}
\]

All the above features are evaluated for all the four orientations of \(\theta_{GL}\) previously described.

In this study, the run-length matrix is not used to analyze the input image, as it is often the case, instead it is used to analyze the GLCM (which is a matrix, and can therefore be analyzed using image processing techniques). In particular, we exploit the descriptors defined in Eqs. (1)–(11) from a run-length matrix that is evaluated on the GLCM. As noted in Section 2.1, several GLCMs are calculated at different values of \(d = \{1,3\}\) (the distance between the two pixels) and \(\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}\) (the direction they are aligned). To take into account all the combinations of such parameters, a single GL descriptor concatenates all the described features (1)–(11). A set of GLCMs is then evaluated using the four values of \(\theta\) and the two values of \(d\), listed in Section 2.2, for a total of eight GLCMs. For each of the eight GLCMs a given GL feature set is extracted. The GL approach has in turn its own set of orientations: \(\theta_{GL} = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}\), which includes all the values originally considered in Tang (1998). In the experimental section, we also explore the performance of the following GL variants:

- \(GR\), as in HR but using the GL features;
- \(GR_{sub}\), as in HRsub but using the GL features;
- \(GR_{sca}\), as in HRsca but using the GL features.

2.5. Spatial diversity (SD)

The concept of diversity comes from the field of ecology and is one of its central themes. Diversity represents a measure of how species are present in the environment and, more specifically, in different habitats and communities. The analysis of diversity concepts considers concepts like population, diffusion, and diversification and is performed by means of indices that analyze the presence of a species within a given environment.

To apply diversity analysis to pattern recognition, it is first necessary to express the data to be analyzed (in this case, images) in terms of population, species, and related concepts. In Geraldo Junior et al. (2013), the population \(P\) is defined as the set of all \(N_p\) pixels in the image, while the gray level values constitute the different species \(s(i)\); the total number of species is \(S\), which can be smaller than 256 if some gray level values are not present in the image. The number of individuals of a species \(s(i)\) is the number \(n(i)\) of pixels in the image with a given gray level. Finally, the relative frequency for a species is defined as: \(p(i) = n(i)/N_p\). Given this model it is possible to exploit the indices commonly used to measure population diversity and apply them to pattern recognition.

The richness of a population is defined as the Shannon–Wiener index, which is defined as:

\[
H = - \sum_{i=0}^{S-1} p(i) \ln p(i) \tag{12}
\]

This value increases with higher heterogeneity levels.

Another measure of diversity is provided by the McIntosh index, which is defined as:

\[
Mc = \frac{N_p - \sqrt{\sum_{i=0}^{S-1} n(i)^2}}{N_p - \sqrt{N_p}} \tag{13}
\]

The Brillouin index, yet another measure of diversity, is best suited when the randomness of the population is not guaranteed:

\[
Hb = \frac{1}{N_p} \left( \ln (N_p) - \sum_{i=0}^{S-1} \ln (n(i)!) \right) \tag{14}
\]

The total diversity index measures species variation, and is defined as:

\[
Td = \sum_{i=0}^{S-1} \frac{1}{n(i)} (p(i)(1 - p(i))) \tag{15}
\]

The Simpson index is a second order statistics, and is defined as:

\[
Ds = \frac{\sum_{i=0}^{S-1} n(i)(n(i) - 1)}{N_p(N_p - 1)} \tag{16}
\]

The Berger–Parker index is the relative frequency of the most frequent species:

\[
Bp = \max p(i). \tag{17}
\]

Additional indices can be defined based on those above. The \(f\) index, for example, compares the value of the \(H\) index for the current population to the same index provided by a maximally diverse population: \(H^*: f = H/H^*\), with \(H^* = \ln S\). The same concept applied to the Simpson index leads to the Ed index: \(d = Ds/Ds^*\).

The Hill index is defined as:

\[
Hil^l = \frac{1}{e^{H - 1}} \tag{18}
\]

The Buzas–Gibson index measures the uniformity of the Shannon index:

\[
Bg = \frac{e^{H}}{S} \tag{19}
\]

Finally, the last index considered in this work is the Camargo index, which compares the relative frequencies of the different species as follows:

\[
E = 1 - \left( \sum_{i=0}^{S-1} \sum_{j=i+1}^{S-1} \frac{p(i) - p(j)}{S} \right) \tag{20}
\]

We also explore the performance of the following SD variants:

- \(SD\), as in HR but using the SD features;
- \(SD_{sub}\), as in HRsub but using the SD features;
- \(SD_{sca}\), as in HRsca but using the SD features.

3. Preprocessing methods

The primary aim of this work is to improve performance using different methods for preprocessing the images before extracting features from the co-occurrence matrix. Before calculating the co-occurrence matrix, however, we normalize the preprocessed image \(I\) to \([0,255]\), such that each element becomes \(I' = [I - \min] / (\max - \min)\times 255\), where \(\max\) and \(\min\) represent the maximum and minimum values found in the images used for training, respectively; moreover, \(\times\) and \(/\) represent the per-element multiplication and division, respectively.

In Fig. 2,\(^4\) we provide some example images derived by applying the preprocessing techniques mentioned above. After the set of preprocessed images is available, a set of descriptors is extracted from each one: this set is then fed into an SVM. The outputs of the SVMs (one for each preprocessing technique) are finally combined by sum rule. Below we briefly describe all the preprocessing methods explored in this work.

\(^4\) In order to better visualize the processed images (PI), the PI are resized to the size of the original image.
3.1. Wavelet (WAV)

A common operation used in many vision problems is the Wavelet transform. WAV (Mallat, 1989) performs a single-level 2-D wavelet decomposition, which requires a 2-D scaling function, \( \zeta(x, y) \), and three 2-D wavelets, \( \psi(x, y) \), where the index \( i = \{ H, V, D \} \) is a superscript representing intensity variations along columns, or horizontal edges (\( \psi^H \)), along rows, or vertical edges (\( \psi^V \)), and along diagonals (\( \psi^D \)). Given the separable scaling function and the three wavelet functions, we first define the scaled and translated basis functions as follows:

\[
\begin{align*}
\zeta_{j,1,2}(x, y) &= 2^j \zeta(2^j x - s1, 2^j y - s2), \\
\psi_{i,1,2}(x, y) &= 2^j/2 \psi(2^j x - s1, 2^j y - s2), \quad i = \{ H, V, D \}
\end{align*}
\]

The discrete wavelet transform of function \( f(x, y) \) of size \( S1 \times S2 \) is then:

\[
\begin{align*}
W_{\zeta}(j_0, s1, s2) &= \frac{1}{2^{S1+S2}} \sum_{x=0}^{S1-1} \sum_{y=0}^{S2-1} f(x, y) \zeta_{j_0,1,2}(x, y), \\
W_{\psi}(j, s1, s2) &= \frac{1}{2^{S1+S2}} \sum_{x=0}^{S1-1} \sum_{y=0}^{S2-1} f(x, y) \psi_{i,1,2}(x, y) \quad i = \{ H, V, D \},
\end{align*}
\]

where \( j_0 \) is an arbitrary starting scale and the \( W_{\zeta}(j_0, s1, s2) \) coefficients define the approximation of \( f(x, y) \) at scale \( j_0 \). The \( W_{\psi}(j, s1, s2) \) coefficients add horizontal, vertical, and diagonal details for scales \( j \geq j_0 \).

In our experiments, we use the Daubechies wavelet with four vanishing moments, single scale decomposition (i.e., \( j = 1 \) and \( j_0 = 1 \)). The four preprocessing images are the approximation coefficient matrix and the three detail coefficient matrices (horizontal, vertical, and diagonal). The preprocessed images are thus \( W_{\zeta} \) and \( W_{\psi}, i = \{ H, V, D \} \). Notice that the processed images have a size that differs from that of the original image. Let us define \( sx \times sy \) the size of the original image and \( lf \) the length of the wavelet filter (here \( lf = 7 \)) then the size of the processed image is \( floor((sx + lf - 1)/2) \times floor((sy + lf - 1)/2) \).

3.2. Local binary patterns (LBP)

The canonical LBP operator (Ojala et al., 2002) is computed at each pixel location of an image by considering the values of a small circular neighborhood (with radius \( R \) pixels) around the value of a central pixel \( I_c \):  \[
LBP(N, R) = \sum_{n=0}^{N-1} s(I_n - I_c)2^n
\]

where \( N \) is the number of pixels in the circular neighborhood, \( R \) is the radius, \( I_n \) are the pixels in the current neighborhood (the varying element in the sum) and \( s(x) = 1 \) if \( x \geq 0 \), otherwise \( s(x) = 0 \).

LBP has proven to be highly discriminative and invariant to monotonic gray level changes. Unfortunately, it is not very robust to the noise present in images when the gray-level changes result from noise that is not monotonic (Tan & Triggs, 2007). Therefore, we use a well-known variant based on ternary coding (Paci et al., 2013) that makes it more robust to noise. The ternary coding is achieved by introducing the threshold \( \tau \) (in this work \( \tau = 3 \)) so that the \( s(x) \) function becomes:

\[
s(x) = \begin{cases} 
1, x \geq \tau \\
0, |x| \leq \tau \\
-1, x \leq -\tau
\end{cases}
\]

The ternary code obtained by the \( s(x) \) function is split into two binary codes by considering its positive and negative components according to the following binary function \( b_v(s(x)) \):

\[
b_v(s(x)) = \begin{cases} 
1 & s(x) = v \\
0 & \text{otherwise}
\end{cases} \quad v \in \{-1, 1\}
\]

Thus, the LTP0 operator for \( v \in \{-1, 1\} \) is:

\[
LTP0_v(N, R) = \sum_{n=0}^{N-1} b_v(s(I_n - I_c))2^n.
\]

In this work we use \( (R = 1; N = 8) \) and \( (R = 2; N = 16) \). Four codes are produced for each pixel (the two sets of parameters and, for both, a positive and a negative LBP code), resulting in four images (one for each code). The size of the processed image is \( (sx - R) \times (sy - R) \).

3.3. Gradients (GRA)

Gradient is a common image operation typically used to detect edges. A gradient magnitude image is defined as an image in which each pixel measures the change in intensity in a given direction of a corresponding point in the original image. GRA calculates the magnitude of the gradients of each pixel in the x (horizontal) and y (vertical) directions based on its neighbors.

For a function \( f(x, y) \) the gradient of \( f \) at coordinates \( (x, y) \) is defined as the 2D column vector:

\[
\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}
\]

The processed image used for extracting the co-occurrence matrix at \( (x, y) \) is given by the magnitude of the gradient vector at the same coordinates and is defined as:

\[
\nabla f = \text{mag}(\nabla f) = \left[ G_x^2 + G_y^2 \right]^{1/2} = \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2}.
\]

The size of the processed image is the same of the original image.
3.4. Orientation (OR)

In OR the image gradient is soft quantized (Vu, 2013) using three orientations, which result in three processed images. This method is used to reduce the noise and other forms of degradation in an image.

Given the gray level of a pixel \( p(i) \) at \((x, y)\) in the gradient image evaluated for the orientation of \( p(i) \):

\[
\theta(x, y) = \text{atan}2(G_y, G_x),
\]

where \( \theta(x, y) = \theta(x, y) + \pi \) in the case where \( \theta(x, y) \geq 0 \).

OR is the magnitude of \( p(i) \) distributed over \( n \) discretized orientations as detailed in Vu, (2013).

The size of the processed image is the same of the original image.

3.5. Weber law descriptor (WLD)

WLD is based on Weber’s law which states that a change of stimulus that is just noticeable for human beings is a constant ratio of the original stimulus; if the change is less than this constant ratio then it is considered background noise. WLD has two components: differential excitation and orientation (Chen, Shan, He, Zhao, & Pietikainen, 2010). For a given pixel, the differential excitation component is computed based on the ratio between (1) the relative intensity differences of a current pixel compared to its neighbors and (2) the intensity of the current pixel. The neighborhood is defined as a square of size \( l_{\text{NE}} \) (we use \( l_{\text{NE}} = 3 \)). From the differential excitation component, both the local salient patterns in the input image and the gradient orientation of the current pixel is computed. We use the differential excitation image for extracting the co-occurrence matrix, and adopt the same notation proposed in (Chen et al., 2010).

To compute a differential excitation \( \xi(l) \) of a current pixel \( l \), the differences between its neighbors must be calculated:

\[
\xi^0 l = \sum_{i=0}^{l_{\text{NE}}^2 - 1} (l_i - l),
\]

where \( l_i(0 \leq i \leq l_{\text{NE}}^2 - 1) \) defines the\( i \)th neighbors of \( l \) and \( l_{\text{NE}}^2 \) is the number of neighbors. The ratio of the differences to the intensity of the current point of both the original image \( \xi^0 l \) and the differential excitation image \( \xi^0 l \) is calculated by:

\[
G_{\text{ratio}}(l) = \frac{\xi^0 l}{\xi^0 l}.
\]

The arctangent function is then employed on \( G_{\text{ratio}}(\ast) \) to calculate the differential excitation \( \xi(l) \):

\[
\xi(l) = \arctan \left[ \frac{\xi^0 l}{\xi^0 l} \right] = \arctan \left[ \sum_{i=0}^{l_{\text{NE}}^2 - 1} \frac{l_i - l}{l} \right] .
\]

When \( \xi(l) \) is positive, it simulates surroundings that are lighter than the current pixel; when \( \xi(l) \) is negative, it simulates surroundings that are darker than the current pixel.

The size of the processed image is \((sx - 1) \times (sy - 1)\).

3.6. Local phase quantization (LPQ)

LPQ (Ojansivu & Heikkilä, 2008) is based on the blur invariance of the Fourier Transform Phase. The spatially invariant blurring of an image \( f(x) \) results in the blurred image, \( g(x) \), which can be expressed as \( g(x) = f(x)^{\ast} h(x) \), where \( x = [x, y]^T \) is the spatial coordinate vector and \( h(x) \) is the point spread function. What this means in the Fourier space can be expressed as \( G(u, v) = F(u)^{\ast} H(u) \), where \( G(u, v) \) and \( H(u) \) are the Discrete Fourier Transforms (DFT) of the blurred \( g(x) \) and \( h(x) \), respectively, and \( u = [u, v] \) is the frequency coordinate vector. The magnitude and phase aspects can be separated resulting in \( |G(u)| = |F(u)| \cdot |H(u)| \) and \( \angle G(u) = \angle F(u) + \angle H(u) \).

The Fourier transform is always real-valued if \( h(x) \) is centrally symmetric, and its phase is a two-valued function given by

\[
\angle H(u) = \begin{cases} 0, & H(u) \geq 0 \\ \pi, & H(u) < 0 \end{cases}
\]

Eq. (32) means that \( \angle G(u) = \angle F(u) \) for all \( H(u) \geq 0 \); in other words, the phase of the observed image \( \angle G(u) \) is invariant to centrally symmetric blur at the frequencies where \( |H(u)| \geq 0 \). LPQ is blur invariant because it uses the local phase information obtained from the Short Time Fourier Transform (STFT) computed over a rectangular neighborhood \( N_s \) of size \( l_{\text{NE}} \times l_{\text{NE}} \) at each pixel position \( x \) of the image \( f(x) \):

\[
F(u, v) = \sum_{y=0}^{N_s-1} f(x - y) e^{-j\pi \omega y} = w_{u,v} f_x
\]

where \( y \) is a spatial coordinate vector; \( w_{u,v} \) is the basis vector of the 2-D DFT at frequency \( u \), and \( f_x \) is a vector containing all the image samples from \( N_s \).

LPQ considers four frequency vectors: \( u_1 = [a,0]^T, u_2 = [0, a]^T, u_3 = [a, a]^T, \) and \( u_4 = [0, a]^T \) where \( a \) is small enough to reside below the first zero crossing of \( H(u) \) that satisfies \( \angle H(u) = \angle F(u) \) for all \( H(u) \geq 0 \).

If we let \( F_x = [F(u_1, x), F(u_2, x), F(u_3, x), F(u_4, x)] \) and \( F_x = [\Re(F_x), \Im(F_x)]^T \), then the corresponding transform matrix is:

\[
W = [\Re\{w_{u_1}, w_{u_2}, w_{u_3}, w_{u_4}\}, \Im\{w_{u_1}, w_{u_2}, w_{u_3}, w_{u_4}\}]^T
\]

Thus, \( F_x = W f_x \).

The coefficients need to be decorrelated before quantization to maximally preserve the information; after decorrelation the samples to be quantized are statistically independent. For details, see Rahtu, Heikkilä, Ojansivu, and Ahonen (2012). Assuming a Gaussian distribution, a whitening transform can achieve independence, such that \( G_x = V^T F_x \), where \( V \) is an orthonormal matrix derived from the singular value decomposition of the covariance matrix of the transform coefficient vector \( F_x \). After computing \( G_x \) for all the image positions, the resulting vectors are quantized using the following scalar quantizer:

\[
q_j = \begin{cases} 1, & G_j \geq 0 \\ 0, & G_j < 0 \end{cases}
\]

where \( G_j \) represent the\( j \)th component of \( G_x \). These quantized coefficients are represented as integers between 0 and 255 using the binary coding

\[
b = \sum_{j=1}^{8} q_j 2^{j-1}
\]

The image that contains the integers obtained with Eq. (36) is the preprocessed image. In this work, two sizes for the local window are used (\( l_{\text{NE}} = 3 \) and \( l_{\text{NE}} = 5 \)), both with Gaussian derivative quadrature filter pairs for local frequency estimation. In this way, two preprocessed images are obtained.

The size of the processed image is \((sx + 1 - l_{\text{NE}}) \times (sy + 1 - l_{\text{NE}}) \).

3.7. Minimum paths by Djikstra’s algorithm (Dj)

Dj (De Mesquita Sá Junior, Backes, & Cortez, 2013) models images using a graph where each pixel corresponds to a node that is connected to its 4-neighbors. The cost, \( w(x, y) \), of a transition between two connected nodes at locations \((x, y)\) and \((x', y')\) is determined by their gray levels \( l(x, y) \) and \( l(x', y') \), such that:

\[
w(x, y) = |l(x, y) - l(x', y')| + \frac{|l(x', y') - l(x, y)|}{2}
\]

Given this model, the image texture is measured by evaluating the minimum cost for connecting four point pairs: (i) the central points
of the first and last column \( p_{1,1} \); (ii) the lower left and upper right corner \( p_{43} \); (iii) the central points of the upper and lower row \( p_{289} \); and (iv) the lower right and upper left corner \( p_{135} \). To find the minimum paths, Dijkstra’s algorithm (Dijkstra, 1959) is exploited since link costs cannot be negative.

The cost \( w \) depends on two elements: the difference in the gray levels of the two pixels and their average value. The cost of the four paths (i–iv described above) will then be higher if a large number of strong transitions is present in the paths; this links the path costs to image texture.

In our experimental section, we divide each image into patches of 4 × 4 pixels. For each patch the four minimum path costs are calculated. For each of the four paths (i–iv), a different image is built where the cost path of patch \( i,j \) of the original image becomes the \( i,j \) element of the new processed image. Since we have four different cost paths, we process the image in four different ways. Notice that, since we use patches of 4 × 4 pixels, the size of the processed image is 25% of the size of the original image (one pixel for each patch).

4. Datasets

To assess our system, we validate it across eleven image datasets, representing different computer vision problems. Table 1 provides a short descriptive summary of each of the following datasets:

<table>
<thead>
<tr>
<th>Name</th>
<th>Abbreviation</th>
<th>#Classes</th>
<th>#Samples</th>
<th>Sample size</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>Histopathology</td>
<td>HI</td>
<td>4</td>
<td>2828</td>
<td>Various</td>
<td><a href="http://www.informed.unal.edu.co/histologyDS">http://www.informed.unal.edu.co/histologyDS</a></td>
</tr>
<tr>
<td>Pap smear</td>
<td>PAP</td>
<td>2</td>
<td>917</td>
<td>Various</td>
<td><a href="http://labs.fme.aegean.gr/decision/downloads">http://labs.fme.aegean.gr/decision/downloads</a></td>
</tr>
<tr>
<td>Viruses classification</td>
<td>VIR</td>
<td>15</td>
<td>1500</td>
<td>41 × 41</td>
<td><a href="http://www.cb.uw.se/~gastal/virustexture">http://www.cb.uw.se/~gastal/virustexture</a></td>
</tr>
<tr>
<td>Breast cancer</td>
<td>BC</td>
<td>2</td>
<td>584</td>
<td>various</td>
<td>upon request to Gerald Braz Junior [<a href="mailto:ge.braz@gmail.com">ge.braz@gmail.com</a>]</td>
</tr>
<tr>
<td>Protein classification</td>
<td>PR</td>
<td>2</td>
<td>349</td>
<td>various</td>
<td>upon request to Loris Nanni [<a href="mailto:nanni@dei.unipd.it">nanni@dei.unipd.it</a>]</td>
</tr>
<tr>
<td>Brodatz</td>
<td>BR</td>
<td>13</td>
<td>2080</td>
<td>205 × 205</td>
<td><a href="http://dismac.dii.unipg.it/rico/data.html">http://dismac.dii.unipg.it/rico/data.html</a></td>
</tr>
<tr>
<td>Pictures</td>
<td>PI</td>
<td>13</td>
<td>503</td>
<td>Various</td>
<td><a href="http://imagelab.ing.unimo.it/files/bible_dataset.zip">http://imagelab.ing.unimo.it/files/bible_dataset.zip</a></td>
</tr>
<tr>
<td>Painting</td>
<td>PA</td>
<td>13</td>
<td>2338</td>
<td>Various</td>
<td><a href="http://www.cat.uab.cat/~joost/painting91.html">http://www.cat.uab.cat/~joost/painting91.html</a></td>
</tr>
</tbody>
</table>

- **PAP**: the pap smear dataset in Jantzen, Norup, Dounias, and Bjerregaard (2005), containing images representing cells used in cervical cancer diagnosis.
- **VIR**: the virus dataset in Kylberg, Uppström, and Sintorn (2011), containing images of viruses extracted by negative stain transmission electron microscopy. We use the same 10-fold validation division of images, which we obtained from the authors of Kylberg et al. (2011). However, rather than use the mask provided in VIR for background subtraction, we use the entire image for extracting features since we found this method performs better.
- **HI**: the histopathology dataset in Cruz-Roa, Caicedo, and González (2011), containing images taken from different organs that represent the four fundamental tissues.
- **BC**: the breast cancer dataset in Junior, Cardoso, de Paiva, Silva, and Muniz de Oliveira (2009), containing 273 malignant and 311 benign breast cancer images.
- **PR**: the protein dataset in Nanni, Shi, Brahnam, and Lumini (2010), containing 118 DNA-binding Proteins and 231 Non-DNA-binding proteins, with texture descriptors extracted from the 2D distance matrix that represents each protein. The 2D matrix is obtained from the 3D tertiary structure of a given protein by considering only atoms that belong to the protein backbone (see Nanni et al., 2010, for details).
- **CHO**: the Chinese hamster ovary cell dataset in Boland, Markey, and Murphy (1998), containing 327 fluorescent microscopy images of ovary cells belonging to five different classes. Images are 16 bit grayscale of size 512 × 382 pixels.
- **HE**: the 2D HeLa dataset (Boland & Murphy, 2001), containing 862 single cell images divided into 10 staining classes taken from fluorescence microscope acquisitions on HeLa cells.
- **LO**: the LOCATE ENDOGENOUS mouse sub-cellular organelles dataset (Fink et al., 2006), containing 502 images unevenly distributed among 10 classes of endogenous proteins or features of specific organelles.
- **BR**: the exact same set of textures used in Gonzalez, Fernandez, and Bianconi (2014), composed of 13 texture classes (2080 images with different rotations) taken from the Brodatz’s album (Brodatz, 1966). The image with rotation \( 0^\circ \) are used in the training set, and the rotated images belong to the testing set.
- **PI**: the dataset in Borghesani, Grana, and Cucchiara (2014), containing pictures extracted from digitized pages of the Holy Bible of Borso d’Este, duke of Ferrara (Italy) from 1450 A.D. to 1471 A.D. The dataset PI is composed of 13 classes that are characterized by clear semantic meaning and significant search relevance.
- **PA**: the dataset in Khan, Beigpour, Weijer, and Felsberg (2014), containing 2338 paintings by 50 painters, representative of 13 different painting styles: abstract expressionism, baroque, constructivism, cubism, impressionism, neo-classical, pop art, post impressionism, realism, renaissance, romanticism, surrealism, and symbolism. All experiments on PA use the same training and testing sets that were used in Khan et al. (2014) (these sets were provided by the authors).

An important goal in our research efforts is to produce general-purpose systems that perform well across very different image datasets and problems. For this reason we test our algorithms on datasets, such as those listed above, that are intentionally very general and different from each other. The presence of too many datasets of the same kind would have polarized the overall results. Moreover, for a system to be truly general-purpose, it is essential that all parameters and weights remain constant across all datasets in our experimental section. This is crucial in order to avoid overfitting on single datasets.

The weights of the different weighted sum rules proposed in this paper are obtained using a grid search (where each weight between 0 and 5 is tested in increments of 0.5) for maximizing the performance in the three datasets used in Nanni et al. (2013b); Nanni et al. (2013c). Also the parameters of the preprocessing methods are obtained in a similar way. The results on these datasets are not reported in this paper since these datasets were used for parameter selection only. The three datasets used to determine the weighted sum rules are the Smoke datasets (Feinui, 2011), the Locate Transfected dataset (Hamilton, Pantelic, Hanson, & Teasdale, 2007), and RNAI, a dataset of fluorescence microscopy images of fly cells (Zhang & Pham, 2011).

5. Experimental results

We used the 5-fold cross-validation protocol to test each texture descriptor, except for the VIR, PR, and PA datasets, which require the
original testing protocols that were detailed in Section 4. The area under the ROC curve (AUC)\(^5\) is the performance indication since it provides a better overview of classification results. AUC is a scalar measure that can be interpreted as the probability that the classifier will assign a higher score to a randomly picked positive sample than to a randomly picked negative sample (Fawcett, 2004). In the multi-class problem, AUC is calculated using the one-versus-all approach (i.e., a given class is considered “positive” while all the other classes are considered “negative”) and the average AUC is reported.

The aim of the first experiment, reported in Table 2, is to determine the usefulness of the different preprocessing techniques applied to the image before creating the co-occurrence matrix. In this experiment all three feature extraction approaches (see Section 2) are tested. In addition, the following fusions are reported:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>WAV + WLD;</td>
</tr>
<tr>
<td>F2</td>
<td>WAV + WLD + LPQ;</td>
</tr>
<tr>
<td>F3</td>
<td>WAV + WLD+LPQ + OR;</td>
</tr>
<tr>
<td>F4</td>
<td>WAV + WLD+LPQ + OR + DJ;</td>
</tr>
</tbody>
</table>

A method represented above as \(A + B\) is the sum rule between approaches \(A\) and \(B\). Before fusion, the scores of \(A\) and \(B\) are normalized to mean 0 and standard deviation 1.

For feature extractors that we tested are HRsca, SHsca, GRsca and SDsca. When a given preprocessing method produces more than one image (e.g., wavelet produces four different images), a SVM classifier is applied separately to each processed image. For instance, with wavelet we train one SVM using the approximation coefficient matrix and three SVMs using the three detail coefficient matrices (horizontal, vertical, and diagonal). This set of SVMs is combined by sum rule. Notice that we report only the best ensemble (average in the four descriptors) built by 2, 3, 4, and 5 preprocessing approaches. We want

\(^5\) AUC is implemented as in dd_tools 0.95 davidt@ph.tn.tudelft.nl.
to stress that it is well known in the literature that a good ensemble is obtained by considering a tradeoff between the performance and the diversity of the individual classifiers that make up the ensemble (Kuncheva, 2014). For this reason, LBP is not used in any ensemble due to its lower diversity with respect to WLD and LPQ.

Examining Table 2, we see that the performance of F3 and F4 are similar. In F3, the DJ preprocessing method, which is computationally expensive, is not used. In the tests that follow, therefore, we use F3 to reduce computation time.

The goal of the second experiment, reported in Table 3, is to show the performance improvements obtainable when the features extracted from the original image are combined with the preprocessed images. The following methods are compared in Table 3:

- \{HR, SH, GR, SD\} is the performance of the descriptors \{HR, SH, GR, SD\} extracted from the whole original image only;
- \{HRsca, HSsca, GRsca, SDsca\} is the performance of the descriptors \{HRsca, HSsca, GRsca, SDsca\};
- \{HRp, SHp, GRp, SDp\} is the best fusion reported in Table 2 by varying the types of features extracted from the processed images;
- \{HRsca + 2, SHsca + 2\} extracted from the whole original image only;
- \{HRsca + 2\} extracted from the processed images.

The aim of the third experiment, reported in Table 4, is to examine the performance gained by fusing different descriptors extracted from the co-occurrence matrix. The methods compared in this experiment are the following:

- NanniA: the best ensemble proposed in Nanni et al. (2013b);
- NanniB: the best ensemble proposed in Nanni et al. (2013c);
- NanniC: the best ensemble proposed in Nanni et al. (2013d);
- SUMdiv: proposed in this paper as the sum rule fusion of HRsca + 2 × HRp, GRsca + 2 × GRp and SDsca + 2 × SDp;
- SUM: proposed in this paper as the sum rule fusion of HRsca + 2 × HRp, GRsca + 2 × GRp, and SHsca + 2 × SHp;
- WEI: proposed in this paper as the sum rule fusion of HRsca + 2 × HRp, GRsca + 2 × GRp, and SHsca + 2 × SHp (weight 1).

Examining Table 4, we find WEI outperforming all our previous ensembles, NanniA, NanniB and NanniC, with a p-value of 0.05 using Wilcoxon signed rank test. SUMdiv obtains a performance similar to SUM. In our experiments, we tried to improve WEI by adding SDsca + 2 × SDp to the ensemble but obtained no improvement in performance.

Finally, in Table 5, we report the results of some of the best performing texture descriptors reported in the literature:

- Local binary patterns (LBP)\(^6\) (Ojala et al., 2002);
- Local ternary patterns (LTP) (Tan & Triggs, 2007);
- Multi-threshold local quinary coding (MT) (Paci et al., 2013);
- Local phase quantization (LPQ) (Rahlu et al., 2012);
- Completed Local binary patterns (CLBP) (Guo, Zhang, & Zhang, 2010).
- W1: an ensemble of descriptors proposed in Nanni et al. (2013d).

---

\(^6\) For LBP, LTP, CLBP, and MT, we use uniform bins with \(P = 8, R = 1\) and \(P = 16, R = 2\); for LPQ the size of the local window was \((3, 3)\).
Moreover we report the following fusion:

- ENS: sum rule between MT and WEI.

Examining Table 5, we see that the proposed system ENS outperforms the other approaches.

6. Conclusion

In this study we improved the performance of the co-occurrence matrix by examining different preprocessing techniques applied to images before extracting features from the co-occurrence matrix. We also compared and combined strategies for extending the texture features extracted from the co-occurrence matrix after transforming the original image with different preprocessing methods. These methods are then improved by combining them with a multi-scale approach that is based on Gaussian filtering and on extracting features from both the entire co-occurrence matrix and a set of sub-windows. For all experiments, the SVM was used as the base classifier.

Our proposed system was validated across several image classification problems, with very different images, thereby demonstrating the generalizability of our approach. We also compared our results with some state-of-the-art descriptors. In our opinion, the results reported in this paper are very useful for practitioners; rather than use Haralick's descriptors, they can now use our more powerful set of features extracted from the GLCM. The code of the proposed ensemble is available at https://www.dei.unipd.it/node/2397. Moreover, our study shows the value of exploring novel methods for deriving more powerful descriptors from the co-occurrence matrix.

Because one of our goals was to create a general-purpose system that is capable of working with many image problems and databases, a limitation of this work could be the number of tested datasets. In the future we plan on expanding tests so that our systems are compared on at least 15 datasets representing different image classification problems. This expansion would provide a more reliable comparison. Another limitation of our proposed system has to do with computation time: the standard approach based on the Haralick's features is considerably less expensive, computationally, than our proposed system since it requires the training of several SVMs. It should be pointed out, however, that each SVM in our proposed ensemble is trained independently from the others, so the proposed approach is well suited for modern multi-core architectures.

As a follow-up to this work, we plan on exploring recent methods that boost the performance of texture descriptors, e.g., approaches where each image is split into different subimages (Vu, Nguyen, & Garcia, 2014). We will also examine new methods for choosing the most powerful set of state-of-the-art texture descriptors (such as, LTP+LQP) that can be extracted from the co-occurrence matrix.

References


         tion & Collaboration, 8(1), 1–18.

         extracting information from the co-occurrence matrix. PLoS ONE, 8(12), e83554.
         doi:10.1371/journal.pone.0083554.

         sures with classification based on featured distribution. Pattern Recognition Letters,

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rota-
         tion invariant texture classification with local binary patterns. IEEE Transactions on
         Pattern Analysis and Machine Intelligence, 24, 971–987.

         phase quantization. In International Conference on Image and Signal Processing
         (pp. 236–243).

Oliveira, A., Llado, X., Freixenet, J., & Martí, J. (2007). False positive reduction in mammo-
         graphic mass detection using local binary patterns. Medical Image Computing and
         Computer-Assisted Intervention, 4791, 286–293.

         for texture descriptors in sub-cellular and stem cell image classification. Current
         Bioinformatics, 8, 208–219.

         ings of the IEEE international geoscience and remote sensing symposium (pp. 342–
         345).

grey level co-occurrence matrix analysis in evaluation of brain cor-
pus callosum and cingulum architecture. Microscopy and Microanalysis: The Official
Journal of Microscopy Society of America, Microbeam Analysis Society, Microscopical


forest using ALOS PALSAR 50 m resolution data with multiscale GLCM analysis.
In Proceedings of the IEEE international geoscience and remote sensing symposium
(pp. 1234–1237).

LBP approach for leaf image recognition. In Proceedings of the intelligent computing
theories and applications (pp. 237–243).

tion temperatures of thermotropic liquid crystals from the local binary gray level

ified color motif co-occurrence matrix for image indexing and retrieval. Computers
& Electrical Engineering, 39(1), 762–774.

Sun, X., Wang, J., Chen, R., She, M. F. H., & Kong, L. (2012). Multi-scale local pattern co-
occurrence matrix for textural image classification. In Proceedings of the IEEE world
congress on computational intelligence (pp. 1–7).

Vision, 7, 11–32.

Tan, X., & Triggs, B. (2007). Enhanced local texture feature sets for face recognition un-
der difficult lighting conditions. Analysis and Modelling of Faces and Gestures LNCS,
4778, 168–182.

Processing, 7(11), 1602–1609.

and classification. In Proceedings of the 5th european conference on computer vision
(pp. 589–600).

Unay, D., & Ekin, A. (2008). Intensity versus texture for medical image search and re-
trieval. In Proceedings of the 5th iee international symposium on biomedical imaging:
from nano to macro (pp. 241–244).

Automation and Remote Control, 24, 774–780.

and Remote Control, 25.

marsh-like classification of celiac disease in children using local texture operators.
Computers in Biology and Medicine, 41, 313–325.

Vu, N.-S. (2013). Exploring patterns of gradient orientations and magnitudes for face
recognition. IEEE Transactions on Information Forensics and Security, 8, 295–
304.

Vu, N., Nguyen, T., & Garcia, C. (2014). Improving texture categorization with
436.

multi-scale GLCM features. International Journal of Pattern Recognition and Ar-
tificial Intelligence, 17, 17–39.

gle image by bayesian combination of edgelet part detectors. In Proceedings of the
IEEE International Conference on Computer Vision, (pp. 90–97).

Zhang, B., & Pham, T. D. (2011). Phenotype recognition with combined features and
random subspace classifier ensemble. BMC Bioinformatics, 12(1), 128.

Zhao, L., Peng, T., Han, H., Cao, W., Lou, Y., & Xie, X. (2014). Fault condition recognition
based on multi-scale co-occurrence matrix for copper flotation process. In Proceed-
ings of the 19th world congress of the international federation of automatic control
(pp. 7091–7097).