



Color image segmentation using saturated RGB colors and decoupling the intensity from the hue

Farid García-Lamont¹ · Jair Cervantes¹ · Asdrúbal López-Chau² · Sergio Ruiz-Castilla¹

Received: 31 January 2019 / Revised: 2 August 2019 / Accepted: 22 September 2019

Published online: 05 November 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

Although the RGB space is accepted to represent colors, it is not adequate for color processing. In related works the colors are usually mapped to other color spaces more suitable for color processing, but it may imply an important computational load because of the non-linear operations involved to map the colors between spaces; nevertheless, it is common to find in the state-of-the-art works using the RGB space. In this paper we introduce an approach for color image segmentation, using the RGB space to represent and process colors; where the chromaticity and the intensity are processed separately, mimicking the human perception of color, reducing the underlying sensitiveness to intensity of the RGB space. We show the hue of colors can be processed by training a self-organizing map with chromaticity samples of the most saturated colors, where the training set is small but very representative; once the neural network is trained it can be employed to process any given image without training it again. We create an intensity channel by extracting the magnitudes of the color vectors; by using the Otsu method, we compute the threshold values to divide the intensity range in three classes. We perform experiments with the Berkeley segmentation database; in order to show the benefits of our proposal, we perform experiments with a neural network trained with different colors by subsampling the RGB space, where the chromaticity and the intensity are processed jointly. We evaluate and compare quantitatively the segmented images obtained with both approaches. We claim to obtain competitive results with respect to related works.

Keywords RGB space · Color image segmentation · Self-organizing maps · Otsu method

1 Introduction

Image segmentation by color features is a, relatively, recent topic addressed, because color representation demands high hardware resources. This aspect has changed the last years because of the technology advances, but also due to the necessity to develop algorithms for color image processing applied in different areas where color analysis is important. For

✉ Farid García-Lamont
fgarcial@uaemex.mx

instance, for food analysis the color is employed to determine the ripeness of fruits [2, 7] or illness detection of fruits [10]; in medicine the color is used to recognize ulcer tissue [12], detection of breast tumors [20], white blood cell counting [34] or study of human eye [16]; among others areas [4, 8, 19, 29, 53, 54].

The classic segmentation techniques are classified as [11]: edge detection, threshold, histogram-thresholding and region-based methods. These techniques, developed to process grayscale images, are extended to process color images. Nevertheless, they do not always success for color processing because they are designed to process the intensity not the chromaticity.

Other techniques widely employed for color image segmentation are based on clustering techniques [1, 5, 11, 25, 28, 40, 42] mainly fuzzy c-means (FCM) [13, 22, 38, 43, 55] and/or unsupervised neural networks (NN) [11, 21, 31, 35, 44]. Despite FCM and NN have been employed successfully for this purpose, they have important disadvantages.

FCM demands to define a priori, usually by the user, the number of clusters the data, pixels, are grouped; so, the number of parts, clusters, an image should be segmented is set depending on the nature of the scene in order not to lose color characteristics of the scene. Besides, the FCM algorithms tend to form the groups with the same size; that is, the elements of the data are distributed such that the groups have almost the same number of elements. Hence, small parts within the image with a specific color are not segmented adequately. But also, the clustering of the elements depends on the initial values of the groups' centers.

For instance, in image (a) of Fig. 1 there are essentially three sections: the sky, the moon and the trees in blue, white and black, respectively. The reader can suppose that the image can be segmented by setting 3 clusters. However, as shown in image (b), after applying FCM with 3 clusters the moon does not appear in the image. Experimentally, we found the moon is successfully segmented until 12 clusters are defined, see image (c); nevertheless, the same part is not always segmented. Image (d) is obtained by processing the input image for 12 clusters, and the moon is not segmented; the explanation is that the initial values of the group's centers are not adequate.

Other example is the image (e) of Fig. 1; in this image there are several small parts with different colors. Six colors are appreciated within the image: flowers in purple, red and yellow, grass and trees in green, two persons dressed in black, a person dressed in white and part of the water in white. Therefore, the image is segmented by setting 6 clusters; image (f) shows the

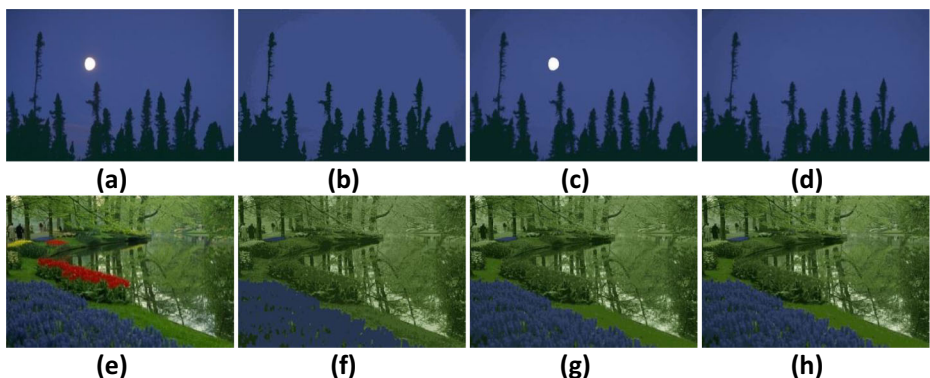


Fig. 1 Input images **a** and **e**; images segmented using FCM with **b** 3 clusters, **c** 12 clusters and **d** input image segmented again with 12 clusters. Images **f**, **g** and **h** segmented using FCM with 6, 14 and 21 clusters, respectively

resulting image. The red and yellow flowers are not segmented with their respective color, intuitively the reader can assume that the image should be segmented in more than six parts, thus, images (g) and (h) are obtained by setting 14 and 21 clusters, respectively.

Although the images are segmented in more parts, the flowers in red and yellow are not segmented in their corresponding colors. But also, notice that in image (f) the flowers in purple are segmented homogeneously with the same hue and intensity, in images (g) and (h) the same part is segmented in purple but with two kinds of intensity; that is, the image becomes over segmented when the number of cluster increases and the small parts are not segmented in their corresponding color.

On the other hand, unsupervised NN are also frequently employed for color image segmentation, especially self-organizing maps (SOM) [21, 31, 35, 44]. The NN is trained with the pixels' colors of the given image; then, the image is processed with such NN. The drawback with this approach is the NN must be trained every time a novel image is given, because the NN just learns to recognize the colors of the given image, and not always can recognize the colors of different images.

Humans learn to recognize the colors just once in life, so, they do not need to learn to recognize them every time they need to identify a given color; they just employ their knowledge acquired previously. Thus, a possible solution for color image segmentation is to train a NN to recognize different colors.

The RGB space is one of the well-known models employed to represent colors; although the RGB space is accepted by most of the image processing community to represent colors, such space is not adequate for color processing because the color differences cannot be computed using the Euclidean distance and it is sensitive to illumination [35]. But given that most of the image acquisition hardware employ the RGB space to represent colors, the development of algorithms for processing RGB color images has been incentivized; but also, the computational load may be huge if the RGB colors are mapped to a different color space for color processing, because almost all of the mathematical operations are not linear [15].

A logical proposal would be to use all the colors of the RGB space to train a NN. But given that in the RGB space the chromaticity and the intensity are not decoupled many colors would be misclassified because several colors with the same chromaticity can have different intensities.

Thus, the contributions of this paper are:

1. The color images are segmented by mimicking the human perception of color, where the chromaticity is processed separately from the intensity, employing the RGB space.
2. We train an unsupervised NN to recognize the chromaticity of different colors, by training it with chromaticity samples of the most saturated colors of the RGB space; the training set is considerably smaller, but very representative, than if the NN would be trained with all the colors of the RGB space. Once the NN is trained, it can be used to process any image without training it again.
3. Since the chromaticity and the intensity are not decoupled in the RGB space, we create an intensity channel by extracting the magnitudes of the color vectors. We employ the Otsu method to compute the threshold values such that the range of the intensity values is divided in three classes. Certain robustness before non-uniform illumination is achieved.

By avoiding mapping colors between color spaces, our proposal contributes with the green computing context. Green computing is a term referring to the efficient use of all

computational resources to minimize the impact on the environment. Some algorithms used in multimedia have a high algorithmic complexity, which has a negative impact on energy consumption [3]. High energy consumption due to execution of algorithms becomes prohibitive in particular scenarios, such as running on mobile devices or on embedded systems; utilizing efficiently the energy of mobile devices is mandatory in this and other cases [9]. Therefore, the generation of methods that allows performing the same tasks as other algorithms is very valuable from the point of view of green computing. The method proposed in this paper is an alternative to other segmentation algorithms, with the advantage of working on original RGB color space, avoiding unnecessary transformations.

The paper is organized as follows: related works are reviewed in Section 2, in Section 3 we introduce our the segmentation proposal, the experiments performed and results obtained are presented in Section 4, in Section 5 the results are discussed and compared with previous works and the paper closes with conclusions in Section 6.

2 Related works

In this section we present the related works we found by reviewing the state of the art on segmentation of images by color features. Next we show the papers we found addressing this topic.

Chang et al. [5] propose a dynamic niching clustering algorithm based on individual-connectedness for unsupervised classification with no prior knowledge. It automatically computes the optimal number of clusters as well as the cluster centers of the data set based on the compact k-distance neighborhood algorithm. With the adaptive selection of the number of the nearest neighbor and the individual-connectedness algorithm, the algorithm achieves several sets of connecting individuals and each set composes an independent niche. Each set of connecting individuals corresponds to a homogeneous cluster and this ensures the separability of an arbitrary data set.

In reference [1] is adapted a clustering stability validation method to automatically segment images. The authors show that clustering and validation acting together as a data-driven process able to find the optimum number of partitions according to the proposed color-texture feature extraction.

Schu and Scharcanski [40] propose a non-supervised clustering method based on adaptive Bayesian trees. A Bayesian framework is proposed for seeking modes of the underlying discrete distribution of the data, which is represented by hierarchical clusters found using the adaptive Bayesian trees approach. The proposed inherent hierarchical tree structure is explored to represent color images hierarchically.

In reference [28] is presented an algorithm that combines k-means clustering with graph cut. The k-means algorithm is applied to make an initial clustering, and the optimal number of clusters is automatically determined by a compactness criterion that is established to find clustering with maximum intercluster distance and minimum intracluster variance. Then, a multiple terminal vertices weighted graph is constructed based on an energy function and the image is segmented according to a minimum cost multiway cut.

Gharieb, Gendy and Abdelfattah [13] present a proposal to fuzzify the c-means algorithm by incorporating both local and membership data. The local membership information is incorporated via two membership relative entropy functions, which measure the information proximity of the membership function of each pixel to the membership average in the immediate spatial neighborhood. By minimizing the membership functions moves the

membership function of a pixel toward its average in the pixel neighborhood. The local data is incorporated into the algorithm by adding to the standard distance a weighted distance computed from the locally smoothed data. The algorithm assigns a pixel to the cluster more likely existing in its immediate neighborhoods.

In reference [44] is introduced a method to perform image segmentation for the color and textured images with a two-step approach. Firstly, self-organizing neurons based on NN are used for clustering the input image; later, multiphase active contour model is used to get various segments of an image. The contours are initialized in the active contour model using the self-organizing maps obtained in the first step.

Ong et al. [35] present a two-stage hierarchical NN based on SOM for color image segmentation. The first stage of the network uses a two-dimensional feature map, which captures the dominant colors of an image. The second stage employs a one-dimensional feature map to control the number of color clusters that is used for segmentation.

Khan and Jaffar [21] address the segmentation of color images as a clustering problem and a fixed length genetic algorithm. An objective function is proposed to evaluate the quality of the segmentation and the fitness of a chromosome. A SOM is used to compute the number of segmented in order to set the length of a chromosome automatically.

In reference [18] is introduced an adaptive and unsupervised approach based on Voronoi regions. The method employs a hybrid of spatial and feature space Dirichlet tessellation followed by inter-Voronoi region proximal cluster merging to automatically find the number of clusters and clusters centroids. Because of the Voronoi regions are smaller with respect to the whole image; Voronoi region-wise clustering improves the efficiency and accuracy of the number of clusters and cluster centroid estimation process.

Heidary and Caulfield [17] present a genetic algorithm for color classification, it designs well-fitted color space prolate ellipsoids that envelop the training pixels. The ellipsoids are then used to classify unlabeled pixels of the image in accordance with their color in order to partition the image.

The fuzzy c-partition entropy approaches have two main limitations: the partition number needs to be manually tuned for different input and the methods can process grayscale images only. In reference [51] is addressed these limitations proposing an unsupervised multilevel segmentation algorithm. A bi-level segmentation operator which employs binary graph cuts to maximize fuzzy 2-partition entropy and segmentation smoothness. The proposed algorithm picks the color channel that can best segment the image into two labels, and then iteratively selects channels to further split each label until convergence.

Li et al. [27] propose the multiphase multiple piecewise constant and geodesic active contour model, which describes multiple objects and background with intensity inhomogeneity. In order to make the optimization more efficient and limit the approximate error, four-color labeling theorem is introduced which can limit the multiple layer graph within three layers, representing four phases. A new and alternative method named heuristic four-color labeling is also proposed, which aims to generate more reasonable color maps with a global view of the whole image.

In reference [41] is presented a non-supervised clustering method based on adaptive Bayesian trees, for seeking nodes of the underlying discrete distribution of the input data, and the data is represented by hierarchical clusters using the adaptive Bayesian trees approach. The clustering technique introduced is applied for color image segmentation, exploring the inherent hierarchical tree structure of the proposed approach to represent color images hierarchically.

Mignotte [33] estimates a segmentation map into regions from a boundary representation. The author defines a non-stationary model, MRF model, with long-range pairwise interaction whose potentials are estimated from the probability of the presence of an edge at each pair of pixels. The paper shows that an efficient and interesting strategy to complex region-based segmentation models consists in averaging soft contour maps and using the MRF reconstruction model to achieve an accurate segmentation map into regions.

Wang et al. [45] propose regularized tree partitioning approaches, where the normalized cut and average cut criteria over a tree are studied. The authors provide the properties that result in an efficient algorithm for normalized tree partition and average tree partitioning. The relations between the solutions of both approaches over the maximum weight spanning tree of a graph are presented.

In reference [46] is presented a sparse global/local affinity graph over superpixels, to capture short- and long-range grouping cues, enabling perceptual grouping laws, including proximity, similarity and continuity through a suitable graph-cut algorithm. Color, texture and shape features are evaluated for their effectiveness in perceptual segmentation. A gravitation law is presented based on empirical observations and divide superpixels adaptively into small, medium and large-sized sets. Global grouping is achieved using medium-sized superpixels through a sparse representation of superpixels' features by solving a minimization problem.

Khelifi and Mignotte [23] propose a fusion model of image segmentation based on multi-objective optimization, which aims to overcome the limitation and bias caused by a single criterion, and to provide a final improved segmentation. The proposed fusion model combines the region-based variation of information criterion and the contour-based F-measure criterion using an entropy-based confidence weighting factor. The authors propose an extended local optimization procedure, to optimize the energy-based model, based on superpixels and derived from the iterative conditional mode algorithm. This multi-objective median partition-based approach has emerged as an appealing alternative to the use of traditional segmentation fusion models.

Long et al. [30] proposed a novel segmentation method based on deep nets for image classification and transfer learning. The architecture they proposed fused features across layers of convolutional networks to define a nonlinear local-to-global representation that are tuned end-to-end. This method can achieve state-of-the-art segmentation performance; however, it requires a training step with images before segmentation, which is a high computational burden that takes about three days on a single GPU.

Another segmentation method that uses convolutional neural network (CNN) is the one presented by Girshick and colleagues in [14]. To segment an image, around 2000 regions are extracted from it, then a CNN is applied to each region to compute a fixed-length feature vector; these vectors are used by a SVM to classify each region. Although this method achieves about 48% segmentation mean accuracy with the images used for testing, it seems it is more adequate for object identification instead of the segmentation task.

A method for retinal blood vessels segmentation was developed by Peng et al. in [52]. They take advantage of certain properties of retinal images (for example, vessels have curvatures; they are piecewise linear and gradually change in intensity along their lengths) to propose a mathematical radial projection that is combined with SVM classifier to segment an image. This method is successful for retinal blood vessels segmentation; however, it is unsuitable to segment other types of images. Some applications such as motion, action or object retrieval [37, 48–50] can be benefited with segmentation methods specifically designed for this purpose.

The idea of dealing directly with data without any transformation has been adopted in some other works. For example, in [6], Chang proposed to apply compound rank-k projections to the

matrices that represent images instead of transform them first into vectors. This approach preserves the correlations within the matrix and decreases the computation complexity.

By reviewing the works above, it is important to remark that several of them employ clustering techniques, mainly FCM and SOMs. As mentioned before, FCM demands to define a priori the number of clusters the data is grouped, and small parts within the image may not be segmented. When SOMs are used, they are usually trained every time with the colors of the given image to process.

In several works process the colors using color spaces where the chromaticity is decoupled from the intensity. Because of most of the image acquisition hardware employs the RGB space to represent colors, it involves mapping the colors to other color spaces, but the computational load by mapping colors between colors spaces may be huge because of the non-linear mathematical operations. It is important to mention that despite using color spaces where the hue and the intensity are decoupled, in the proposed approaches found in the literature process the hue and the intensity jointly.

Nevertheless, despite the drawbacks we have mentioned, in different works the RGB space is employed to process colors; but, such approaches for image segmentation propose to extract and include other features.

In our approach, the images are segmented using the RGB space, without mapping the colors to other color space. A NN is employed to process the chromaticity of colors, which is trained with chromaticity samples of the most saturated colors of the RGB space. The NN can process any image without training the NN every time a novel image is given. An intensity channel is created by computing the color vectors' magnitude of the given image; we propose to group the intensity in three classes using the Otsu method. By processing the chromaticity and the intensity separately, robustness before non-uniform illumination is achieved, to some extent.

3 Segmentation approach

In this section we present our proposal which consists mainly on two stages: 1) recognition of the colors by chromaticity features using an unsupervised NN trained to recognize the chromaticity of colors employing just chromaticity samples of the most saturated colors of the RGB space, obtaining a training set considerably small, but whose elements are very representative. The idea is that each neuron learns to recognize the hue of a specific color, and then they are activated when the hue of a color or a similar one feeds the NN; 2) in order to classify the colors by the intensity, we create an intensity channel by extracting the magnitudes of the color vectors, then the values of such channel are grouped in three classes; thereby certain robustness before non-uniform illumination is achieved. We employ the Otsu method to compute the optimal thresholds to establish the intensity ranges and then to group the intensities in three classes.

3.1 RGB color space

The RGB space is based on the Cartesian coordinate system where colors are points defined by vectors that extend from the origin [15], where black is located in the origin and white in the opposite corner to the origin. Images (a) of Fig. 2 show the RGB space schematically, while the images (b) and (c) show the appearances of the outer and inner faces of the RGB space, respectively.

The color of a pixel p is a linear combination of the basis vectors red, green and blue, which can be written as:

$$\phi_p = r_p \hat{i} + g_p \hat{j} + b_p \hat{k} \tag{1}$$

Where the scalars r , g and b are the red, green and blue components of the color vector, respectively. It is important to remark the features of the color vectors [15]:

1. The orientation represents the chromaticity
2. The magnitude models the intensity

The number of colors of the RGB space is infinite, but in the image processing field the RGB space is discretized; usually, the range of each component is $[0,255] \subset \mathbb{Z}$.

3.2 Training set for chromaticity recognition

Our proposal mimics the human perception of color, where humans recognize the colors first by the chromaticity and then by the intensity. One of the steps of our approach consists on to train a NN to recognize the chromaticity of different colors; hence, a training set is built with chromaticity samples of different colors. But, which and how many colors must be selected as elements for the training set? The logical solution is to employ all the colors of the RGB space. As mentioned in Section 3.1, the usual range of values of each color component is $[0,255] \subset \mathbb{Z}$, thus, in such space there are $256^3 = 16777216$ colors. If a NN is trained with such large number of colors, many colors would be misclassified because several colors with the same chromaticity can have different intensities; but also, the NN may become over trained.

If we analyze the chromaticity changes of the colors within the RGB space, it is easy to appreciate that such changes are not so abrupt. That is, Fig. 3 shows the colors located in different middle planes of the RGB cube. The images of the first, second and third rows show the R-G, R-B and B-G planes, respectively. The images of the columns (a) and (f) show the inner and outer faces of the cube, respectively.

Notice that the colors tend to be less saturated according they “move away” from the inner faces of the cube; however, it is easy to appreciate the chromaticity changes are not so important, see images of column (b). From the middle of the cube, the chromaticity changes significantly with respect to the chromaticity of the inner planes, see images of columns (c) and

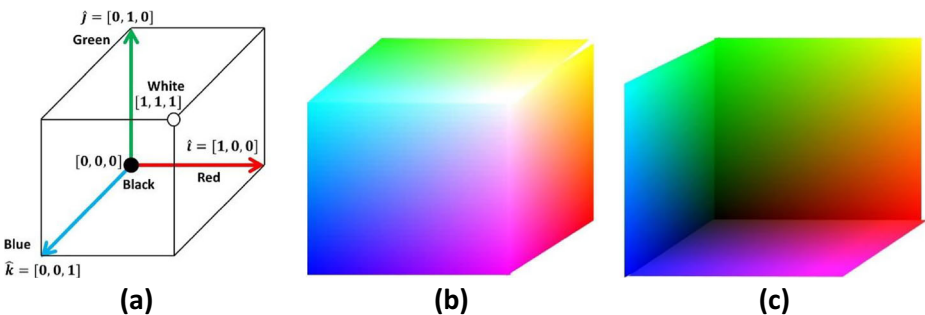


Fig. 2 Image a schematic representation of the RGB color space; images a and b show the appearances of the outer and inner faces of the cube, respectively

(d). In the following planes, columns (d) and (e), the chromaticity resembles the chromaticity of the outer planes, column (f); nevertheless, all these hues can be found in the inner faces.

For instance, in the image (f) of the R-G row, the blue, cyan and magenta colors can be appreciated; the blue and cyan colors can be found in the image (a) of the row B-G, while the magenta color can be appreciated in the image (a) of the row R-B. Other example, in the image (f) of the R-B row, the green and yellow colors can be appreciated; these colors can be found in the image (a) of the R-G row.

Thus, we claim extracting just samples of colors located in the inner planes of the cube; i.e., the most saturated colors of the RGB space. In each plane there are $256^2 = 65536$ colors, but we are interested on the chromaticity. As stated in Section 3.1, the orientation of the color vectors models the chromaticity; then, the elements of the training set are vectors located in the inner faces of the RGB cube but at different orientations.

Therefore, for this study, we employ color vectors placed every 3° . That is, the training set is built as follows. The elements of the set Θ are numbers multiple of three in the range between 0 and 90.

$$\Theta = \{3n | 0 \leq n \leq 30, n \in \mathbb{Z}\} \tag{2}$$

The sets S and C are built by computing the sinus and cosine values of the elements of the set Θ , respectively.

$$S = \{\sin\theta_k | \forall \theta_k \in \Theta\} \tag{3}$$

$$C = \{\cos\theta_k | \forall \theta_k \in \Theta\} \tag{4}$$

Using the Cartesian product, in the sets P_1, P_2 and P_3 there are placed the color vectors of the inner faces of the RGB cube for the planes R-G, G-B and R-B, respectively.

$$P_1 = C \times S \times \{0\} \tag{5}$$

$$P_2 = \{0\} \times C \times S \tag{6}$$

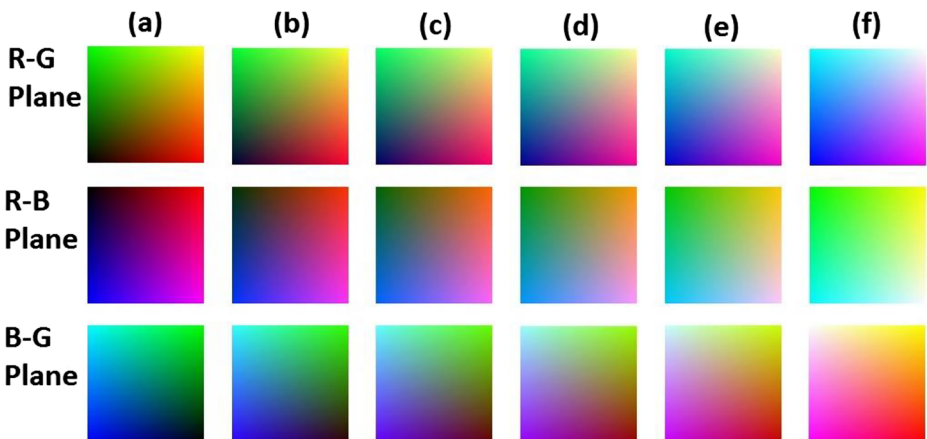


Fig. 3 Middle planes of the RGB cube. First, second and third planes show the R-G, R-B and B-G planes, respectively

$$P_3 = S \times \{0\} \times C \quad (7)$$

Finally, the training set P is obtained by the union of the sets denoted in Eqs. (5), (6) and (7).

$$P = \bigcup_{i=1}^3 P_i \quad (8)$$

Note that the number of elements of the training set P is 93, and the vectors of the training set are unit-length. An unsupervised NN is trained with the set P ; once the NN is trained, it can be employed to process any image without training it again. It is important to remark that under this approach the NN processes the chromaticity of colors separated from the intensity. In the following Section 3.3 we show how the intensity channel is created and how the intensity levels are computed for our purpose.

3.3 Computing the intensity ranges

In the RGB space there is not an intensity channel, as other color spaces, decoupled from the chromaticity; the intensity is implicit in the magnitude of the vectors. This is one of the reasons the RGB space is sensitive to non-uniform illumination; nevertheless, we claim the image can be processed successfully under our approach. Thus, an intensity channel is created by extracting the magnitudes of the color vectors. The magnitude of the color vectors is computed using the Euclidean distance.

After the colors are first recognized by the chromaticity: red, green, blue, etc., they are sub classified by the intensity; for instance: light red or dark red. In this work, we use the Otsu method because it has been employed widely for segmentation of grayscale images; this method computes the optimal threshold δ by maximizing the between-class variance σ_B^2 , which is equivalent to minimizing the within-class variance σ_W^2 , since the total variance is constant for different partitions [36], that is:

$$\delta = \arg \max_{0 < \delta < L} \sigma_B^2(\delta) = \arg \min_{0 < \delta < L} \sigma_W^2(\delta) \quad (9)$$

Where L is the number of intensity levels.

The Otsu method can be extended to divide the intensity levels in several ranges by computing different thresholds to define the size of every range; in this study we consider three intensity classes: dark, medium and bright. Therefore, the intensity levels are divided in four ranges. It is important to remark that although the intensity levels are divided in four ranges, we group the intensity levels in three classes, where the second and third ranges are merged. Experimentally we found that if the intensity is divided initially in three ranges, the images tend to be dark. Thus, the intensity levels are divided in four ranges, but we merge the second and third ranges, obtaining three intensity classes.

Hence, we compute three thresholds: δ_1 , δ_2 and δ_3 that divide the pixels' intensities of the given image in four ranges. The extended version of the Otsu method to compute three thresholds is performed with the following mathematical operations. Let the Eq. (10) be the objective function to maximize:

$$\{\delta_1, \delta_2, \delta_3\} = \arg \max_{0 < \delta_1 < \delta_2 < \delta_3 < L} \sigma_B^2(\delta_1, \delta_2, \delta_3) \quad (10)$$

Where

$$\sigma_B^2(\delta_1, \delta_2, \delta_3) = \sum_{j=1}^4 \omega_j (\mu_j - \mu_T)^2 \tag{11}$$

$$\omega_k = \sum_{i=\delta_{k-1}+1}^{\delta_k} p_i \tag{12}$$

$$\mu_k = \sum_{i=\delta_{k-1}+1}^{\delta_k} \frac{ip_i}{\omega_k} \tag{13}$$

$$\mu_T = \sum_{i=0}^L ip_i \tag{14}$$

Where $p_i = n_i/N$, n_i is the number of pixels at intensity level i and N is the total of pixels of the image; the thresholds $\delta_0 = 0$ and $\delta_4 = L$. The magnitude of the vectors is often a real number; but the Otsu method requires integer values to represent the intensity levels. Thus, $n_i = \# \{k | f(\|\phi_k\|) = i, \forall \phi_k \in \Phi\}$, where $\#$ denotes the cardinality of the set, $f: \mathbb{R} \rightarrow \mathbb{Z}$ is a function that extracts the integer part of a real number and $\Phi = \{\phi_1, \dots, \phi_m\}$ is the set of color vectors of the input image.

The range of intensity levels is defined by the magnitudes of the smallest and largest color vectors within the RGB space. The vector with the lowest magnitude corresponds to the black color; that is, considering that the usual range of the red, green and blue components is $[0,255]$, let $\phi_b = [0, 0, 0]$ be the color vector of black color, therefore $\|\phi_b\| = 0$. While $\phi_w = [255,255,255]$ is the largest vector representing the brightest white color, this vector extends from the origin to the opposite corner of the RGB cube, see Fig. 2; therefore $\|\phi_w\| = 255\sqrt{3}$. So, the range of the intensity values is $[0, 255\sqrt{3}]$. Note that L is an integer number, and the highest intensity value is $255\sqrt{3} \approx 441.673$; hence, only the integer part of this number is employed, thus $L = 442$.

3.4 Proposed approach algorithm

Due to the NN is trained with chromaticity samples of different colors, the NN cannot recognize the white color because it is not a chromaticity. The white color can be assumed as the least saturated color. Thus, before processing a color, we verify whether if that color is white or not.

For such purpose, the orientation of the input color vector is compared with respect to the orientation of the vector representing white color. If the angle between both vectors is smaller than a given threshold then the color is not saturated enough to represent a chromaticity, so, the input vector is set as the vector that models the white color; otherwise, the input vector models a color different than white and it is processed by the NN.

In other words, let ϕ_p be the input color vector and let $\phi_w^* = [1, 1, 1]$ be the color vector reference that models the white color; the vector $\hat{\phi}_p$ is computed as follows:

$$\hat{\phi}_p = \begin{cases} \phi_w^*, & \Delta\theta_p \leq \delta_\theta \\ \arg \max_{\mathbf{w}_i} (\mathbf{w}_i \cdot \phi_p), & \Delta\theta_p > \delta_\theta \end{cases} \quad (15)$$

Where δ_θ is a given threshold value, \mathbf{w}_i is the weight vector of the neuron i and $\Delta\theta_p$ is the orientation difference between the vectors ϕ_p and ϕ_w^* , computed with:

$$\Delta\theta_p = \cos^{-1} \left(\frac{\phi_p \cdot \phi_w^*}{\|\phi_p\| \|\phi_w^*\|} \right) \quad (16)$$

Notice that $\hat{\phi}_p$ represents either the chromaticity the winner neuron learnt to recognize or the white color; the next step is to define the intensity. The color's intensity is computed by verifying in which range lays the magnitude of the input vector, that is:

$$\lambda_p = \begin{cases} 0, & 0 \leq \|\phi_p\| < \delta_1 \\ 127, & \delta_1 \leq \|\phi_p\| < \delta_3 \\ 255, & \delta_3 \leq \|\phi_p\| \leq L \end{cases} \quad (17)$$

Where δ_1 , δ_2 and δ_3 are computed with the Otsu method as explained in Section 3.3. As we stated before, if the intensity levels are divided in three ranges, the images tend to be dark. So, the intensity levels are divided in four ranges, but we merge the second and third ranges, obtaining three intensity classes. We consider as dark those colors whose intensity is in the range $[0, \delta_1)$; while the colors whose intensity is in the ranges $[\delta_1, \delta_3)$ and $[\delta_3, L]$ are considered as medium and bright colors, respectively, where $L = 442$.

Finally, the input color vector ϕ_p of the pixel p is substituted by the vector φ_p as follows:

$$\varphi_p = \lambda_p \cdot \hat{\phi}_p \quad (18)$$

In summary, segmenting an image with our proposal consists of the following steps, let $\Phi = \{\phi_1, \dots, \phi_m\}$ be the set of color vectors of the given image:

1. Build a training set with chromaticity samples of different saturated colors with Eqs. (2) to (8).
2. Train an unsupervised NN with the training set built in step 1.
3. Obtain the intensity channel of the given image by computing the magnitudes of the color vectors $\phi_p \in \Phi$, using the Euclidean distance.
4. Compute the thresholds δ_1 , δ_2 and δ_3 with the Otsu method, as explained in Section 3.3.
5. For all the color vectors $\phi_p \in \Phi$, with Eq. (15), verify if the color vector ϕ_p represents the white color or a chromaticity, obtaining the vector $\hat{\phi}_p$.
6. Obtain the magnitude of the color vector ϕ_p and compute the value of λ_p with Eq. (17).
7. The color vector ϕ_p is substituted with the vector φ_p obtained with Eq. (18).

In the following section we present the resulting images by applying our proposal, we perform the experiments using unsupervised NNs at different sizes.

4 Experimental set up and results

The experiments are performed using SOMs, which are a kind of unsupervised NN; they are based on finding the winning neuron before external stimuli. In other words, the output

neurons compete between them so as to find the best match with the external stimuli. We employ the inner product to measure the match between each neuron of the NN and the external pattern; that is [24]:

$$\mathbf{w}_k = \arg \max_{\mathbf{w}_i} (\mathbf{w}_i \cdot \mathbf{x}_p) \quad (19)$$

Where \mathbf{w}_k is the weight vector of the winning neuron and \mathbf{x}_p is the external pattern. The weight of the winning neuron \mathbf{w}_k is updated with the Kohonen learning rule [24]:

$$\mathbf{w}_k(j) = (1-\alpha)\mathbf{w}_k(j-1) + \alpha\mathbf{x}_p \quad (20)$$

Where $0 < \alpha < 1$ is the learning rate and j is the iteration number. During training, the weight vectors of the neighbor neurons, located at a defined distance within the neuron array, are also updated.

We perform experiments on the images of the Berkeley Segmentation Database¹ (BSD) because it is becoming the benchmark for testing segmentation algorithms of color images [21]. The BSD contains 500 color images of size 481×321 for which the ground truth is known. The ground truth is employed to compute quantitatively the performance of the segmentation algorithms and then to compare them with related works. Later, in the discussion section we present the metrics employed for the quantitative evaluation.

All the 500 images of the BSD are processed; because of space constraints within the paper, we show 20 images we select from the BSD, so as to show them as examples of the images obtained with our approach, see Fig. 4.

The number of colors a NN can recognize depends on its size. Thus, in Section 4.1 we show the images obtained employing a 3×3 -neuron SOM and a 5×5 -neuron SOM. During the experiments we found that the threshold value $\delta_\theta = 8.12^\circ$ is suitable for this purpose.

On the other hand, we have mentioned before that employing all the colors of the RGB space to train the NNs is not adequate, because of the chromaticity and the intensity are not decoupled, the NN misclassifies the colors with the same chromaticity but with different intensity. In order to show this aspect, in Section 4.2 we perform experiments on the images of Fig. 4, using other 3×3 and 5×5 -neuron SOMs trained with several color samples of the RGB space, where the chromaticity and the intensity are processed jointly. Because of the huge number of colors within the RGB space, the training set is built as follows:

$$T = \{5n | 0 \leq n \leq 51, n \in \mathbb{Z}\}^3 \quad (21)$$

Notice that the training set contains $52^3 = 140608$ elements or colors. We show the images obtained with this approach.

4.1 Results obtained with our approach

Figure 5 shows the resulting images by processing the images of Fig. 4 using a 3×3 -neuron SOM. In most of the parts of the images' sections are segmented with a uniform or homogeneous color and in most of the cases the hue of the segmented areas corresponds to the hue of the original image.

¹ www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

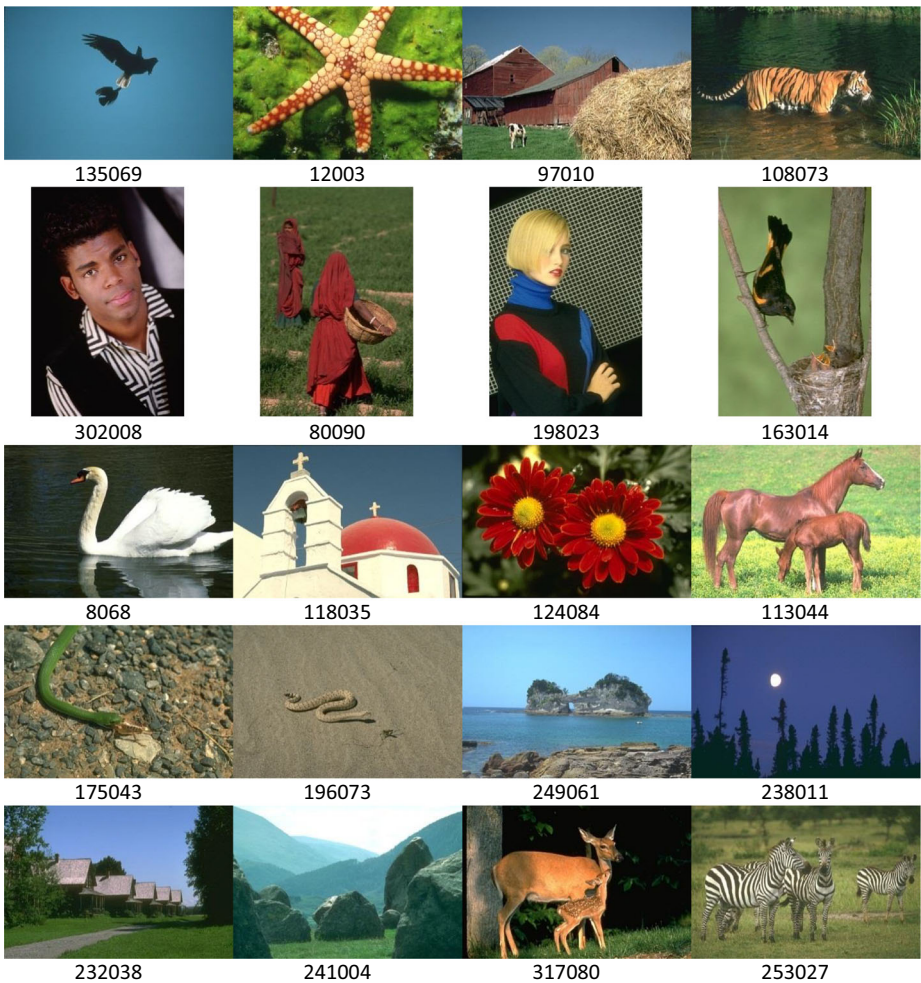


Fig. 4 Images employed for experiments, taken from the Berkeley segmentation database

By increasing the size of the NN, more colors can be recognized, but the areas with the same color may not be segmented homogeneously with the same hue. Figure 6 shows the resulting images employing the 5×5 -neuron SOM.

The segmentation of the images of both figures is the same or very similar; the images 135069, 232038, 253027, 118035 and 249061 are segmented with the same areas and colors.

In other images the segmentation is the same but with different colors; in the image 12003 the star is segmented with two kinds of yellow hue using the 3×3 -neuron SOM, while using the 5×5 -neuron SOM the star is segmented in yellow and red hues. It is easy to appreciate in image 97010, of Fig. 5, the parts of the sky, grass, barn and chaff segmented in blue, green, orange and light orange, respectively, and other parts in white, black and/or gray. The resulting image 97010 shown in Fig. 6 is almost the same, except for the barn and the chaff which are segmented with different orange hues.



Fig. 5 Image obtained with a 3×3 -neuron SOM

The images obtained, from image 108073, with both NNs are very similar; the difference is the reflex of the tiger on the water, which is better defined in Fig. 6. In the reflex there are the orange hue of the tiger and the green hue of the grass. Most of the parts in white, gray and black of the image 302,008 are homogenous in both resulting images. The face is segmented with similar brown color; the lips are segmented in red and purple in Figs. 5 and 6, respectively. The differences between the segmented images obtained from the image 80090 lies in the hue of the basket and the soil; in the image processed by the 3×3 -neuron SOM the basket and the soil are segmented in green, while the same part is segmented in orange by using the 5×5 -SOM.

The segmented images from image 198023 with both NNs have two differences. In the image of Fig. 5 the hair is segmented with the same hue of the skin; while in the resulting image with the 5×5 -neuron SOM, the hair and the skin of the face and the hand are segmented with green-like hue and the lips are segmented in red. The differences of the resulting images by processing the image 163014 are the hue of the background and the branches; where the brown parts of the trunk segmented by the 3×3 -neuron SOM, the same parts are segmented in green using the 5×5 -neuron SOM.

In image 8068, both segmented images are the same, but the color of the swan's head is orange and green when the 3×3 and 5×5 -neuron SOM are used, respectively. The images obtained from the image 124084 with both NNs very are similar, but the sections of the image



Fig. 6 Image obtained with a 5×5 -neuron SOM

obtained with the 5×5 -neuron SOM are less homogeneous. For instance, the petals are segmented with two kinds of red hue; the centers of the flowers are segmented in yellow but the parts around the centers are segmented in orange and the leaves of the background are segmented in two kinds of green hue.

The segmented images of the horses of the image 113044 are almost the same, both in shape and color. In the segmented image obtained with the 5×5 -neuron SOM the background is segmented in green and yellow hues, while in the image segmented with the other SOM the background is segmented only with green hue. The differences between the segmented images by processing the image 175043 lie on the stones and the soil. The stones are segmented in gray and white using the 3×3 -neuron SOM, while some stones are segmented in green using the 5×5 -neuron SOM. The soil is segmented in brown and orange using the 3×3 and 5×5 -neuron SOM, respectively.

The image 196073 can be considered as a difficult image to segment because in this case the hues of the snake and the sand are very similar; however, in the resulting images the snake is segmented from the sand in yellow and orange employing the 3×3 and 5×5 -neuron SOM, respectively. In both images the sand is segmented in gray and white. The resulting images by processing the image 238011 are almost the same; where the sky, the moon and the trees are segmented homogeneously in blue, white and black, respectively. Except the image obtained using the 5×5 -neuron SOM, where some parts around the moon and the sky are segmented in purple.

The segmented images obtained from the image 241004 are basically segmented in 4 parts: the sky in white, the hills in blue, the grass in green and the rocks in black. In resulting image using the 5×5 -neuron SOM, some parts of the stones are segmented in green and some areas of the grass are segmented in two kinds of green hue. The segmentation differences obtained by processing the image 317080 lies on the deer, which are segmented in yellow and orange using the 3×3 and 5×5 -neuron SOM, respectively; but also, notice that the image processed by the 5×5 -neuron SOM, the back of the adult deer is segmented in two different hues.

4.2 Results obtained by subsampling the RGB space

Figure 7 shows the images obtained by employing the 3×3 -neuron SOM trained with the elements of the set T , see Eq. (21). The segmentations of the resulting images are not as homogenous as the images obtained with our approach, the chromaticity does not always correspond to the respective segmented part and the colors are not totally saturated. But also, colors with the same chromaticity but with different intensity are classified as if they were different hue.

By using the 5×5 -neuron SOM the segmentation of the images is improved because in different images the colors of the segmented areas are more homogenous, several parts are

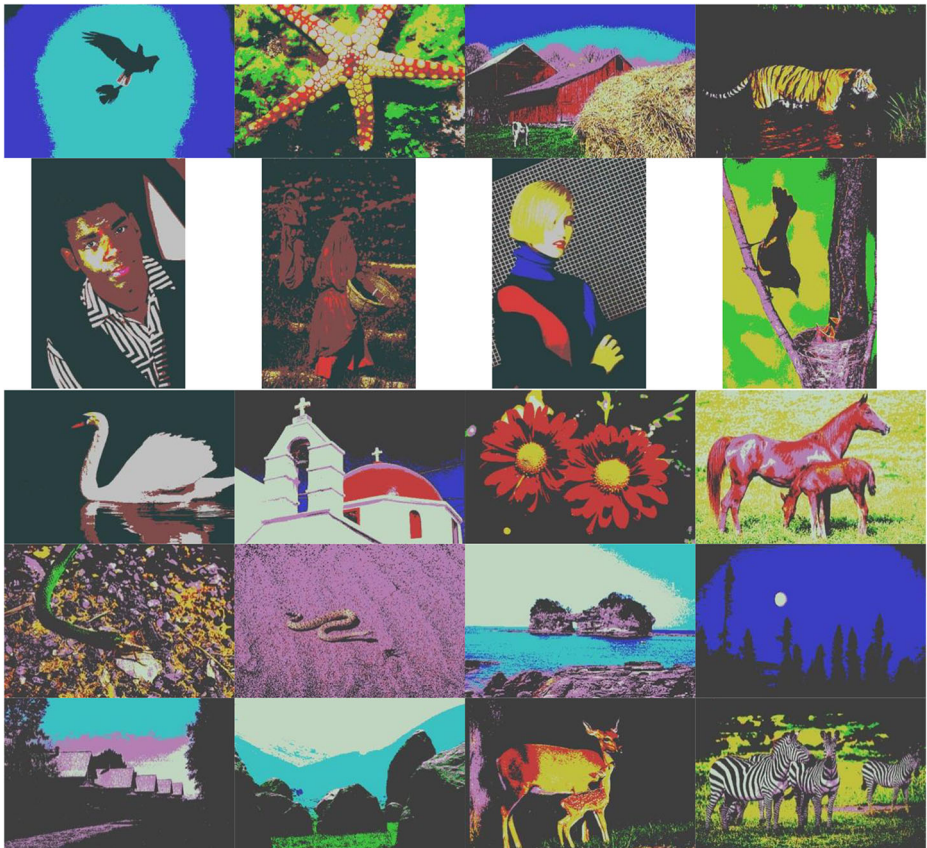


Fig. 7 Images obtained with a 3×3 -SOM and $52^3 = 140608$ RGB color samples

segmented with the same or similar colors of the input image and the colors are more saturated; but in many cases the intensity of the colors influences negatively the color classification, see Fig. 8.

The sky of the image 135069 of Fig. 7 is segmented in blue and cyan, and a small part in black; while the same parts of the image of the Fig. 8 are segmented in cyan and blue and the birds in black and white. The segmented images obtained from the image 12003 are almost the same; it is easy to appreciate that the colors of the image shown in Fig. 8 are more saturated than the image of the Fig. 7. The most notable differences between the images obtained from the image 97010 are: in the image 97010 of Fig. 7 the sky is segmented in blue and cyan, while in the same image shown in Fig. 8 the sky is segmented in cyan and gray. But also, the image of Fig. 7 is darker than the image of Fig. 8.

The image 108073 of Fig. 7 is very dark, the tiger is segmented in yellow but its reflex on the water is fuzzy and some grass is segmented. In the image obtained with the second SOM, the tiger is segmented in orange and some parts in yellow, the reflex of the tiger and more grass are better segmented. In the image 302008 of Fig. 7, the face of the man is segmented in brown and some parts in yellow and white, the lips are segmented in pink; the rest of the image is segmented in white and black. While the in image processed with the second SOM, the face of the person is segmented with different colors.



Fig. 8 Images obtained with a 5×5 -SOM and $52^3 = 140608$ RGB color samples

The image 80090 processed with the first SOM is segmented basically in red and black. The segmented image 80090 of Fig. 8 is improved with respect to the segmented image shown in Fig. 7; the grass is segmented in green, the clothes in red, the soil and the basket in brown. The image 198023 of Fig. 7 resembles the image obtained with our proposal, but part of the face is segmented in white and part of the red area of the sweater is segmented in two intensity classes. By processing the same image with the 5×5 -neuron SOM, the blue and red parts of the sweater are segmented in two kinds of intensities; and the skin and the hair are segmented in two kinds of yellow hue, the lips are segmented in red.

The most notable difference between the images obtained from the image 163014 is the background. In the image of the Fig. 7 the background is segmented in green, yellow and a small part in black, while in the image of the Fig. 8 the background is segmented in gray and some parts in yellow. The swan of the image 8068 using the 3×3 -neuron SOM is segmented in white, but the reflex of the swan over the water is segmented in red. The segmentation of the swan shown in Fig. 8 resembles the images obtained with our approach, but some parts of the swan's reflex on the water are segmented in green.

The resulting image of the image 118035 of Fig. 7 is more homogeneous than the image obtained using the second SOM. Most of the background of the image 124084 processed by the 3×3 -neuron SOM is segmented in black; almost only the flowers are appreciated. In the same image shown in Fig. 8, the leaves are segmented in green, the petals and the flowers' centers are segmented in two classes of red hue and yellow, respectively.

The image 113044 of Fig. 8 is more homogenous than the image obtained with the 3×3 -neuron SOM. With the first SOM, the background of the image is segmented mainly in yellow; the grass below the horses is segmented in green; while with the second SOM the background and the horses are segmented in green and brown, respectively. In Fig. 7, the stones of the image 175043 are segmented in pink, the soil in yellow and the snake in green and black. The same image, shown in Fig. 8, the snake, the stones and the soil are segmented in two kinds of green hue, gray and brown and some parts in gray, respectively. The resulting images of the image 196073 are similar, but the colors of the image shown in Fig. 8 are better classified because the colors are alike to the colors of the input image.

It is easy to appreciate in image 249061 of Fig. 7 how the intensity of the colors influences the recognition; it is notable how the sky is segmented in white and blue. Also, the image same image segmented with the 5×5 -neuron SOM improves the image segmented with the 3×3 -neuron SOM, however the hue of the water is not homogenous. The images obtained from the image 238011 are similar; in the image processed by the 3×3 -neuron SOM, there are parts of the sky in black, in the image obtained with the second SOM, the pixels around the moon are segmented in gray.

Most of the image 232038 of Fig. 7 is segmented in black, the sky is segmented in cyan, pink and white, the path and the roofs of the houses are segmented in pink. In the image processed by the second SOM resembles the image obtained with our proposal; the difference lies in the sky which is segmented in two different hues. The image 241004 of Fig. 7 is alike to the segmented images obtained with our proposal, the sky, the mountains, the stones and the grass are segmented in white, cyan, black and green, respectively. The image 241004 of Fig. 8 is segmented in several areas, but they are not segmented homogeneously.

The image 317080, shown in Fig. 7, is mostly segmented in black; the deer can be appreciated but not the tree and the grass. The image processed by the second SOM resembles the resulting image with our approach, but not all the areas are segmented homogeneously, for instance, the deer are segmented in two kinds of orange hue but with different intensities. Several parts of the image 253027, processed by the 3×3 -neuron SOM, are segmented in

black; the zebras and the grass are easy to appreciate them in their respective color. The image obtained with the 5×5 -neuron SOM is not as obscure as the segmented image obtained with the first SOM, the grass is segmented in green, the zebras are segmented in black and white and some small parts in red and green.

The segmentation of the images is improved when the size of the NN is increased, since the NN can recognize more colors. Nevertheless, not all the areas are segmented homogeneously because they may have different chromaticities or intensities. Despite the NN is able to recognize more colors, the undesired effects of the non-uniform illumination cannot be avoided, while with our approach such undesired effects are reduced.

The experiments were performed on a PC, where the processor is an Intel Core i7-4770 at 3.4GHz and 16GB RAM; the algorithms were implemented in Matlab R2016a under Windows 8.1 platform. On average, the running time to process an image under our approach is 4.5 s.

5 Discussion

In this section we discuss how the size of the NN influences the image segmentation, especially when the NNs are trained with color samples of the RGB space, where the larger the NNs are the more colors can recognize, but the segmented areas are less homogeneous and effects of the intensity cannot be avoided or reduced.

We evaluate quantitatively the resulting images; although there are not standard metrics for this purpose, the probabilistic rand index (PRI), variation of information (VOI) and global consistency error (GCE) metrics are the most employed [11]. With the quantitative evaluation is possible to compare the performance of different algorithms presented in previous works. We compare quantitatively the segmented images obtained under our proposal and the ones processed by the NNs trained with color samples of the RGB space, where the chromaticity and the intensity are processed jointly.

Although the colors are processed in the RGB space, with our proposal certain robustness before illumination is achieved by processing the chromaticity and the intensity separately, despite these features are not decoupled in the RGB space.

5.1 Size of the NN

The number of colors the NN can recognize depends on its size; the larger the NN is the more areas within the image may be segmented by the NN, but the hue of the segmented areas may be less homogenous. Nevertheless, with our proposal the segmented parts keep hue homogeneity. The resulting images with our approach, Section 4.1, and using both NNs are very alike because they have almost the same segmented areas and hues. Using the larger NN there are segmented some little parts that the smaller NN does not segment; besides, the hues of the segmented areas are more alike to the hues of the input images. Notice that, with both NNs, the areas are homogeneously segmented.

In other words, with our proposal the appearances of the segmented images are similar between them, despite using NNs at different sizes. Theoretically, the NN can recognize up to 93 colors because that is the number of elements of the training set; nevertheless, it is possible to recognize more colors if the number of chromaticity samples of the most saturated colors is increased.

Notice that the segmentation of the images using the NNs trained with samples of RGB colors is improved, when the size of the NN is augmented, see Section 4.2. There are segmented more

areas and the colors assigned are alike to the colors of the input image. But not all the areas are segmented homogeneously because, in one hand, since the NN is able to recognize more colors, an area whose pixels have similar chromaticity can be segmented with different colors, see images 198023 and 124084 of Fig. 8; on the other hand, the intensity of the colors influences such that the areas may be segmented with different hues, see 135069 and 232038 of Fig. 8.

The disadvantage of processing the chromaticity and the intensity jointly is that the colors with the same chromaticity but with different intensity can be recognized as different hues. As we have stated previously, the image segmentation may be improved if the size of the NN is increased; however, the undesired effects of the intensity still affect the segmentation. That is, despite the size of the NN is augmented, the intensity effects are not eliminated.













For instance, Table 1 shows the resulting images using 7×7 , 9×9 and 11×11 -neuron SOMs; these images resemble, up to a point, the images obtained with our approach, but it is easy to appreciate how areas with the same hue are segmented with different chromaticity because of the intensity differences.

Notice from Table 1 that the sky of the images 135,069 is segmented with two or three different hues. The horses of the images 113,044 are segmented with different hues and intensities; the sky of the images 97,010 and 232,038 are segmented in two or three different hues and intensities.

5.2 Quantitative evaluation

As we state in Section 4, we use the images of the BSD because it is becoming the benchmark for testing segmentation algorithms of color images [21]. The BSD contains 500 color images

Table 1 Images obtained by using a 7×7 , 9×9 and 11×11 -neuron SOMs trained with $52^3 = 140,608$ RGB color samples

7x7-neuron SOM	9x9-neuron SOM	11x11-neuron SOM
		
		
		
		

of size 481×321 for which the ground truth is known; for each of these images, the database provides between 4 and 9 human segmentations in form of label maps in order to evaluate quantitatively the resulting images.

According to the related works we reviewed, despite several metrics have been propose, there have not been already defined absolute metrics to evaluate the algorithms' performance quantitatively. However, we have found in different recent works [11, 27, 33, 40, 46] the PRI, VOI and GCE are often employed, leading to suppose they are becoming the standard metrics for qualitative evaluation.

The PRI compares the image obtained from the tested algorithm to a set of manually segmented images [11]. Let $\{I_1, \dots, I_m\}$ and S be the ground truth set and the segmentation provided by the tested algorithm, respectively. L_i^k is the label of pixel x_i in the k th manually segmented image, and L_i^S is the label of pixel x_i in the tested segmentation. The PRI index is computed with:

$$PRI(S, I_k) = \frac{2}{n(n-1)} \sum_{i,j,l < j} \left(p_{i,j}^{c_{i,j}} \left(1 - p_{i,j} \right)^{1-c_{i,j}} \right) \tag{22}$$

Where n is the number of pixels, $c_{i,j}$ is a Boolean function: $c_{i,j} = 1$ if $L_i^k = L_j^S$, $c_{i,j} = 0$ otherwise; $p_{i,j}$ is the expected value of the Bernoulli distribution for the pixel pair.

The VOI index measures the sum of loss of information and the gain between two clusters belonging to the lattice of possible partitions in the following way [11]:

$$VOI(S, I_k) = H(S) + H(I_k) - 2F(S, I_k) \tag{23}$$

Where H is the entropy $-\sum_{i=1}^c (n_i/n) \log(n_i/n)$, n_i being the number of points belonging to the i th cluster, c is the number of clusters, and F is the mutual information between two clusters defined as:

$$F(S, I_k) = \sum_{i=1}^{c_S} \sum_{j=1}^{c_{I_k}} \frac{n_{i,j}}{n} \log \frac{n_i n_j}{n^2} \tag{24}$$

Where $n_{i,j}$ is the number of points in the intersection of cluster i of S and j of I_k ; c_S and c_{I_k} are the number of clusters of S and I_k , respectively.

The GCE computes how a segmented image is viewed as the refinement of other. A measure of error at each pixel x_i can be written as [11]:

$$C(S, I_k, x_i) = \frac{|R(S, x_i) \setminus R(I_k, x_i)|}{|R(S, x_i)|} \tag{25}$$

Where $|\cdot|$ is the cardinality, \setminus is the set difference and $R(S, x_i)$ is the set of pixels corresponding to the region in segmentation S that contains the pixel x_i . The measure enforces all local refinements to be in the same direction, this is defined as:

$$GCE(S, I_k, x_i) = \frac{1}{n} \min \left(\sum_{i=1}^n C(S, I_k, x_i), \sum_{i=1}^n C(I_k, S, x_i) \right) \tag{26}$$

The ranges of the PRI, VOI and GCE metrics are $[0, 1]$, $[0, \infty)$ and $[0, 1]$, respectively. The higher the value of PRI, the better the segmentation is; similarly, the lower the values of VOI

Table 2 Average performance obtained with SOMs with different sizes, using the metrics PRI, VOI and GCE by processing all the images of the BSD

NN	Our approach			Subsampling the RGB space		
	PRI	VOI	GCE	PRI	VOI	GCE
SOM3 × 3	0.7499	2.1500	0.3316	0.6176	2.8886	0.3412
SOM5 × 5	0.7557	2.2123	0.3359	0.6789	3.2196	0.4262
SOM7 × 7	0.7670	2.2123	0.3389	0.7043	3.4095	0.4494
SOM9 × 9	0.7684	2.4929	0.3436	0.7133	3.6634	0.4532
SOM11 × 11	0.7752	2.5181	0.3440	0.7155	3.8152	0.4481

and GCE the better the segmentation is with respect to the ground truth. Table 2 shows the average quantitative evaluation of the segmented images obtained with our approach and the NNs trained with colors of the RGB spaces.

According to the Table 2, our proposal obtains the best results. The PRI value is slightly improved with the largest NNs, the highest PRI value is obtained with the 11 × 11-neuron SOM; while the lowest VOI and GCE values are obtained with the 3 × 3-neuron SOM. Taking into a count the results obtained with our approach, we consider convenient to use either the 5 × 5 or 7 × 7-neuron SOM, because the differences of the quantitative evaluations between the SOMs are low. In other words, the performances of a large SOM or a small SOM are almost the same; but also, by using a “small” SOM less computational resources are demanded.

With the SOMs trained with color samples of the RGB space, it is easy to appreciate that the larger the SOM is the higher the PRI value is; while the smaller the SOM the smaller the VOI and GCE values are. Nevertheless, the quantitative evaluation is lower than the obtained with our proposal.

In Table 3 we compare the performance rate obtained with our proposal, using the 7 × 7-neuron SOM, with respect to the values reported in different works employing the BSD as

Table 3 Average rates comparison between related works and our proposal by processing all the images of the BSD

Authors	PRI	VOI	GCE	Color spaces employed
Schu and Scharcanski [40]	0.8148	–	0.1701	L*a*b*
Liang and Zhang [28]	0.7712	2.5616	0.2932	RGB
Khan and Jaffar [21]	0.8332	1.9239	–	L*a*b*
Yin, Qian & Gong [51]	0.7769	2.3067	0.2215	L*a*b*
Li et al. [27]	0.8073	1.8260	0.2079	RGB
Mignotte [33]	0.7900	–	0.2100	RGB, HSV, YIQ, XYZ, L*a*b*, L*u*v*, I ₁ I ₂ I ₃ , H ₁ H ₂ H ₃ , YCbCr and TSL
Wang et al. [45]	0.8039	2.0210	0.2066	RGB and L*u*v*
Wang et al. [46]	0.8384	1.8012	0.1934	RGB and L*a*b*
Khelifi and Mignotte [23]	0.8100	1.9700	0.2100	RGB, HSV, YIQ, XYZ, L*a*b*, L*u*v*, I ₁ I ₂ I ₃ , H ₁ H ₂ H ₃ , YCbCr and TSL
Medeiros and Scharcanski [32]	0.7900	–	–	L*a*b*
Wang et al. [47]	0.7800	–	–	RGB and L*a*b
Li, Wu and Chang [26]	0.8319	1.6849	0.1779	L*a*b*
Our proposal: SOM7 × 7	0.7670	2.2123	0.3389	RGB

benchmark. The table shows the resulting values using the PRI, VOI and GCE metrics and also the color spaces used in each work.

The performance rate of our approach is close to the performance rates of the works cited. It is important to remark that in our proposal we employ just the RGB space to process the colors, while most of the works cited in the Table 3 extract the color features represented in different color spaces, which are more suitable for color processing, like L*a*b* or HSV. The works that employ the RGB space to process colors, also extract and employ color features from different color spaces; for instance, in references [46, 47] the RGB space is used along with the L*a*b* space.

On the other hand, it is also important to mention that the works cited in both the Table 3 and Section 2, the color characteristics of the image are complemented with other features extracted from the given image, usually texture and spatial characteristics. For instance, the RGB space is employed to process the colors in the references [27, 28], but in the segmentation proposals add spatial characteristics.

We claim our proposal is competitive because, we obtain close results to the reported by the references cited in Table 3, where we just extract color features from the RGB space and without using other characteristics from the given image.

5.3 Robustness before illumination

One of the most difficult problems in image processing is to eliminate or avoid the undesired effects of the non-uniform illumination. For color image processing such problem is often addressed by mapping the colors to color spaces where the chromaticity is decoupled from the intensity; but mapping the colors to different color spaces may involve a huge computational load. As we have stated before, the RGB space is very sensitive to non-uniform illumination because the chromaticity is not decoupled from the intensity. However, the RGB space is widely used for color processing, because it is easy to understand since it is a 3D Cartesian system, as presented in Section 3.1, so, in different previous works prefer still using the RGB space; also, as we state in Section 1, because most of the image acquisition hardware employs the RGB space to represent colors.

Several works that employ the RGB space process the color channels separately [11, 39], but given the correlation between the color channels, the hue of the colors may suffer undesired changes. Other related works [11, 28] process the colors as vectors, orientation and magnitude jointly, but the colors with the same chromaticity and with different intensity are classified as if they were different hues, see images of Figs. 7 and 8 and Table 1.

With our approach the chromaticity and the intensity of the colors are processed separately, achieving, to some extent, robustness before illumination. The SOM classifies the colors by their chromatic features because the NN is trained with chromaticity samples of the most saturated colors of the RGB space, as we show in the Section 3. The color vector of the input pixel is updated with the weight vector of the SOM's winning neuron. It is important to remark that all the weight vectors of the SOM's neurons are unit length; therefore, the vector does not contain intensity data about the color, but the orientation of the vector that models the chromaticity is implicit in the vector. In other words, let $\mathbf{w}_k = [r_k, g_k, b_k]$ be the weight vector of the neuron k , the direction cosines of this vector are $\cos\alpha_k = r_k / |\mathbf{w}_k|$, $\cos\beta_k = g_k / |\mathbf{w}_k|$ and $\cos\theta_k = b_k / |\mathbf{w}_k|$. Since $|\mathbf{w}_k| = 1$, therefore, the components of \mathbf{w}_k are the cosines of the angles between the vector and the basis vector.

The intensity of the colors is defined by modifying the length or magnitude of the winning neurons' weight vectors, as explained in Sections 3.3 and 3.4; notice that in this step the

orientation data of the vectors is not employed, just the magnitude. Hence, with our proposal, certain illumination robustness is obtained despite the colors are processed in the RGB space, see images of Figs. 5 and 6.

6 Conclusions

In this paper we have introduced a proposal for image segmentation by color features, where the RGB space is employed to represent the colors, without mapping the colors to other color spaces more suitable for color processing. The chromaticity and intensity of the colors are processed separately, emulating the human perception of color. The chromaticity is processed by a self-organizing map trained with a small set of chromaticity samples of the most saturated colors; once the self-organizing is trained, it can be employed without training it every time a novel image is given.

Since there is not an intensity channel in the RGB space as other color spaces, we create an intensity channel by computing the magnitudes of the color vectors. The intensity is processed using the Otsu method, where the threshold values to divide the range of intensities in three classes are computed. Despite the underlying sensitiveness of the RGB space to illumination, certain robustness before illumination is obtained by processing separately the chromaticity and the intensity.

In the resulting images with the self-organizing maps trained with several colors of the RGB space, the chromaticity recognition is influenced by the intensity of the colors; thus, such approach is not as robust to illumination as our proposal.

In the quantitative evaluation, using as benchmark the Berkeley segmentation database, the images obtained with our approach obtains better results than the obtained with the resulting images with the self-organizing maps trained with several colors of the RGB space.

According to the quantitative evaluation, the highest evaluation is obtained with the 11×11 neuron self-organizing map, however, the difference of the quantitative evaluation is small; thus, we consider that a 5×5 or 7×7 -neuron self-organizing map is suitable to obtain acceptable segmented images. The quantitative evaluation of our approach is similar, to some extent, to the values obtained in previous works. Considering that in the cited works for color image segmentation, in one hand, employ different color spaces where the chromaticity is decoupled from the intensity; on the other hand, there are extracted and included other characteristics from the given image in order to segment the images. Hence, we claim our approach is competitive.

It is important to mention that the quantitative evaluation is subjective because the segmented images are compared with respect to the defined ground truth provided by the Berkeley segmentation database; where the ground truth is a set of label maps manually made by a human. But, the perception of color between humans may be different. Nevertheless, the Berkeley segmentation database and the metrics we employ are widely used in the related works to evaluate the performance of the proposed algorithms for color image segmentation.

Hence, the future research direction is to apply and perform experiments with our proposal in the different areas we mention in the introduction section: food analysis, medicine, among other disciplines where the color processing plays an important role. But also, the color features extracted with our approach can be used to segment and/or recognize objects alongside with other possible characteristics extracted from the images. Thereby, the versatility and segmentation performance of our approach can be observed, but also, to study and propose possible improvements to our approach so as to enhance the color recognition.

References

1. Bayá AE, Larese MG, Namías R (2017) Clustering stability for automated color image segmentation. *Expert Syst Appl* 86:258–273
2. Behroozi-Khazaei N, Reza-Maleki M (2017) A robust algorithm based on color features for grape clusters segmentation. *Comput Electron Agric* 142:41–49
3. Bhavana N, Sanjay K (2013) Impact of algorithms on green computing. *IJCA proceedings on international conference on green computing and technology*, pp 15–17
4. Chakraborty BK, Bhuyan MK, Kumar S (2017) Combining image and global pixel distribution model for skin colour segmentation. *Pattern Recogn Lett* 88:33–40
5. Chang D, Zhao Y, Liu L, Zheng C (2016) A dynamic niching clustering algorithm based on individual-connectedness and its application to color image segmentation. *Pattern Recogn* 60:334–347
6. Chang X, Feiping N, Wang S, Yang Y, Zhou X, Zhang C (2016) Compound rank-k projections for bilinear analysis. *IEEE Trans Neural Netw Learn Syst* 27(7):1502–1513
7. Chen X, Yang SX (2013) A practical solution for ripe tomato recognition and localization. *J Real-Time Image Process* 8:35–51
8. Chen W, Wang K, Jiang H, Li M (2016) Skin color modeling for face detection and segmentation: a review and a new approach. *Multimed Tool Appl* 75:839–862
9. De-Thu H, Min C, Trong-Thua H, Chu HH (2019) Energy consumption optimization for green device-to-device multimedia communications. *Future Gener Comput Syst* 92:1131–1141
10. Dubey SR, Jalal AS (2016) Apple disease classification using color, texture and shape features from images. *Signal Image Video Process* 10:819–826
11. Garcia-Lamont F, Cervantes J, Lopez A, Rodriguez L (2018) Segmentation of images by color features: a survey. *Neurocomputing* 292:1–27
12. Garcia-Zapirain B, Shalaby A, El-Baz A, Elmaghraby A (2017) Automated framework for accurate segmentation of pressure ulcer images. *Comput Biol Med* 90:137–145
13. Gharieb RR, Gendy G, Abdelfattah A (2017) C-means clustering fuzzified by two membership relative entropy functions approach incorporating local data information for noisy image segmentation. *Signal Image Video Process* 11:541–548
14. Girshick R, Donahue F, Darrell T, Malik J (2016) Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans Pattern Anal Mach Intell* 38(1):142–158
15. Gonzalez RC, Woods RE (2002) *Digital image processing*. Prentice Hall, Upper Saddle River
16. Hassan G, Hassanien AE (2018) Retinal fundus vasculature multilevel segmentation using whale optimization algorithm. *Signal Image Video Process* 12:263–270
17. Heidary K, Caulfield HJ (2014) Color classification using margin-setting with ellipsoids. *Signal Image Video Process* 8:1245–1262
18. Hettiarachchi R, Peters JF (2017) Voronoi region-based adaptive unsupervised color image segmentation. *Pattern Recogn* 65:119–135
19. Joumani SE, Mechkouri SE, Zennouh R, Kadmiri OW, Masmoudi L (2017) Segmentation method based on multiobjective optimization for very high spatial resolution satellite images. *EURASIP J Image Video Process* 2017:26
20. Kermani S, Samadzadehghadam N, EtehadTavakol M (2015) Automatic color segmentation of breast infrared images using a Gaussian mixture model. *Optik* 126:3288–3294
21. Khan A, Jaffar MA (2015) Genetic algorithm and self organizing map based fuzzy hybrid intelligent method for color image segmentation. *Appl Soft Comput* 32:300–310
22. Khan A, Ullah J, Jaffar MA, Choir TS (2014) Color image segmentation: a novel spatial fuzzy genetic algorithm. *Signal Image Video Process* 8:1233–1243
23. Khelifi L, Mignotte M (2017) EFA-BMFM: a multi-criteria framework for the fusion of colour image segmentation. *Inf Fusion* 38:104–121
24. Kohonen T (1990) The self-organizing map. *Proc IEEE* 78:1464–1480
25. Lezoray O, Charrier C (2009) Color image segmentation using morphological clustering and fusion with automatic scale selection. *Pattern Recogn Lett* 30:397–406
26. Li Z, Wu ZM, Chang SF (2012) Segmentation using superpixels: a bipartite graph partitioning approach. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp 789–796
27. Li K, Tao W, Liu X, Liu L (2018) Iterative image segmentation with feature driven heuristic four-color labeling. *Pattern Recogn* 76:69–79
28. Liang B, Zhang J (2014) Algorithm based on k-means clustering and graph cut. *Math Probl Eng*. <https://doi.org/10.1155/2014/464875>
29. Liang L, Zhang ZM (2017) Structure-aware enhancement of imaging mass spectrometry data for semantic segmentation. *Chemom Intell Lab Syst* 171:259–265

30. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. *IEEE conference on computer vision and pattern recognition*, pp 3431–3440
31. López-Rubio FL, Domínguez E, Palomo EJ, López-Rubio E, Luque-Baena RM (2016) Selecting the color space for self-organizing map based foreground detection in video. *Neural Process Lett* 43:345–361
32. Medeiros R, Scharcanski J, Wong A (2013) Natural scene segmentation based on a stochastic texture region merging approach. In: 2013 IEEE International conference on acoustics, speech, and signal processing. *IEEE*, pp 1464–1467
33. Mignotte M (2014) A non-stationary mrf for image segmentation from a soft boundary map. *Pattern Anal Appl* 17:129–139
34. Mohd Safuan SN, Md Tomari MR, Wan Zakaria WN (2018) White blood cell (WBC) counting analysis in blood smear images using various color segmentation methods. *Meas* 116:543–555
35. Ong SH, Yeo NC, Lee KH, Venkatesh YV, Cao DM (2002) Segmentation of color images using a two-stage self-organizing network. *Image Vis Comput* 20:279–289
36. Otsu N (1979) A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern SMC-9*:62–66
37. Qinkun X, Ren S (2018) Action recognition based on hierarchical dynamic Bayesian network. *Multimed Tool Appl* 77(6):6955–6968
38. Rajaby E, Ahadi SM, Aghaeinia H (2016) Robust color image segmentation using fuzzy c-means with weighted hue and intensity. *Digit Signal Process* 51:170–183
39. Rajinikanth V, Couceiro MS (2015) RGB histogram based color image segmentation using firefly algorithm. *Procedia Comput Sci* 46:1449–1457
40. Schu G, Scharcanski J (2018) A new method for building adaptive Bayesian trees and its application in color image segmentation. *Expert Syst Appl* 98:57–71
41. Schu G, Scharcanski J (2018) A new method for building adaptive Bayesian trees and its application in color image segmentation. *Expert Syst Appl* 98:57–71
42. Siang Tan K, Mat Isa NA (2011) Color image segmentation using histogram thresholding – fuzzy c-means hybrid approach. *Pattern Recogn* 44:1–15
43. Siang Tan K, Mat Isa NA, Hong Lim W (2013) Color image segmentation using adaptive unsupervised clustering approach. *Appl Soft Comput* 13:2017–2036
44. Sridevi M, Mala C (2017) Self-organizing neural networks for image segmentation based on multiphase active contour. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-017-3045-1>
45. Wang J, Jiang H, Jia Y, Hua XS, Zhang C, Quan L (2014) Regularized tree partitioning and its application to unsupervised image segmentation. *IEEE Trans Image Process* 23:1909–1922
46. Wang X, Tang Y, Masnou S, Chen L (2015) A global/local affinity graph for image segmentation. *IEEE Trans Image Process* 24:1399–1411
47. Wang Y, Liu C, Zhu F, Boushey CJ, Delp EJ (2016) Efficient superpixel based segmentation for food analysis. In: 2016 IEEE International Conference on Image Processing (ICIP). *IEEE*, pp 2544–2548
48. Xiao Q, Wang H, Li F, Gao Y (2011) 3D object retrieval based on a graph model descriptor. *Neurocomputing* 74(17):3486–3493
49. Xiao Q, Yichuang L, Wang H (2014) Motion retrieval based on switching Kalman filters model. *Multimed Tool Appl* 72(1):951–966
50. Xiao Q, Yi Q, Wang H (2015) Motion retrieval using weighted graph matching. *Soft Comput* 19(1):133–144
51. Yin S, Qian Y, Gong M (2017) Unsupervised hierarchical image segmentation through fuzzy entropy maximization. *Pattern Recogn* 68:245–259
52. You X, Peng Q, Yuan Y, Cheung YM, Lei J (2011) Segmentation of retinal blood vessels using the radial projection and semi-supervised approach. *Pattern Recogn* 44(10-11):2314–2324
53. Zhang L, Verma B (2017) Superpixel-based class-semantic texon occurrences for natural roadside vegetation segmentation. *Mach Vis Appl* 28:293–311
54. Zhang S, You Z, Wu Z (2017) Plant disease leaf image segmentation based on superpixel clustering and EM algorithm. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-017-3067-8>
55. Zhou Z, Zhao X, Zhu S (2018) K-harmonic means clustering algorithm using feature weighting for color image segmentation. *Multimed Tool Appl* 77:15139–15160



Farid García-Lamont received a B.Sc. degree in Robotics from ESIME-IPN, Mexico in 2000; in 2004 obtained a M.Sc. in Automatic Control from the Centro de Investigación y de Estudios avanzados del IPN (CINVESTAV-IPN), Mexico. In 2010 received a PhD degree in Computer Science from CINVESTAV-IPN. His research interests are pattern recognition, applications of artificial intelligence and robotics.



Jair Cervantes received the B.Sc. degree in Mechanical Engineering from Orizaba Technologic Institute, Veracruz, Mexico, in 2001, a M.Sc. degree and a PhD from CINVESTAV-IPN, Mexico, in 2005 and 2009, respectively. His research interests include support vector machine, pattern classification, neural networks, fuzzy logic and clustering.



Asdrúbal López-Chau received a PhD in Computer Science from the Department of Computer Science at CINVESTAV-IPN Mexico, in 2013, a M.Sc. degree in Computer Engineering from CIC-IPN Mexico in 2000, and a B.Sc. degree in Electronic Engineering from ESIME-IPN Mexico in 1997. His research interests include machine learning computational intelligence and data mining.



Sergio Ruiz-Castilla received the M.S. degree in Computer Science from Post Graduate College in Texcoco, Mexico in 2005. His research interests include geographic information systems, pattern recognition and neural networks.

Affiliations

Farid García-Lamont¹ · Jair Cervantes¹ · Asdrúbal López-Chau² · Sergio Ruiz-Castilla¹

Jair Cervantes
chazarra17@gmail.com

Asdrúbal López-Chau
alchau@uaemex.mx

Sergio Ruiz-Castilla
jsergioruizc@gmail.com

¹ Universidad Autónoma del Estado de México, Centro Universitario UAEM Texcoco, Av. Jardín Zumpango s/n, Fraccionamiento El Tejocote, CP 56259 Texcoco, Estado de México, Mexico

² Universidad Autónoma del Estado de México, Centro Universitario UAEM Zumpango, Camino viejo a Jilotzingo continuación Calle Rayón, CP 55600 Zumpango, Estado de México, Mexico