Research Article

Improved Unsupervised Color Segmentation Using a Modified HSV Color Model and a Bagging Procedure in K-Means++ Algorithm

Edgar Chavolla, Arturo Valdivia, Primitivo Diaz, Daniel Zaldivar, Erik Cuevas, and Marco A. Perez

Electronics Department, CUCEI, University of Guadalajara, Avenida Revolución 1500, 44430 Guadalajara, JAL, Mexico
Correspondence should be addressed to Edgar Chavolla; chavolla@gmail.com

Received 6 October 2017; Revised 24 November 2017; Accepted 2 January 2018; Published 14 February 2018

Abstract

Accurate color image segmentation has stayed as a relevant topic between the researches/scientific community due to the wide range of application areas such as medicine and agriculture. A major issue is the presence of illumination variations that obstruct precise segmentation. On the other hand, the machine learning unsupervised techniques have become attractive principally for the easy implementations. However, there is not an easy way to verify or ensure the accuracy of the unsupervised techniques; so these techniques could lead to an unknown result. This paper proposes an algorithm and a modification to the HSV color model in order to improve the accuracy of the results obtained from the color segmentation using the K-means++ algorithm. The proposal gives better segmentation and less erroneous color detections due to illumination conditions. This is achieved shifting the hue and rearranging the $H$ equation in order to avoid undefined conditions and increase robustness in the color model.

1. Introduction

Machine learning application area is growing every day, so it is possible to find applications using machine learning in health areas [1–5], system behavior predictions [6–10], image and video analysis [11–15], and speech and writing recognition [16–20], just to mention some of the most notable and recent applications. Some of the results of these advances in machine learning can be appreciated in several applications that are widely and freely available. As a result, the usage of machine learning has become a common component in the daily modern life.

Despite the huge improvements done in machine learning in the recent years, it still requires more work and research. This can be stated from the fact that a total accuracy is not yet achieved by machine learning, and sometimes the results are still not usable. This is the reason for the present proposal, which is developed in the spirit of improving a common process performed in image analysis using machine learning.

This paper gives an overview of the color models by stating their advantages and problems found when they are implemented or are used as input in other processes. A second topic discussed in the color models overview is the concept of chromatic and achromatic separation by the exclusion or mitigation of the illumination component. The latter topic has significant relevance, since many of the issues found in color detection and segmentation come from the fact that the illumination can change the perception from a color ranging from bright white to black, visiting several tones of the same base color. The overview will set the ground on which some changes are made to the HSV perceptual color model.

Also a section regarding machine leaning algorithms is included, where the relevance of unsupervised learning is explained. Also in this section what issues can be found in the popular $K$-means algorithms is explained, as well as of course some existing techniques used to improve the classification results obtained from the usage of this algorithm (like $K$-means++). This section is used to support some minor extra changes applied in combination with some commonly used techniques that can improve the outcome from the algorithm.
The changes applied to the color model and to the classification algorithm are implemented in a way that they aid the resulting classification process. The resulting process is compared against other color models and some variants. As a testing set, the Berkley Segmentation Dataset and Benchmark BSDS500 is mainly used [21], which provides several testing images and ground truth segmentations done by different subjects. The BSDS500 dataset has been used as a testing environment by other segmentation works [22–27]. Using the BSDS500 dataset gives a more reliable ground in the testing cases.

2. Commonly Used Colors Models

Color is a property that is usually addressed in computer vision, because it becomes useful to distinguish and recognize objects or characteristics in an image. Due to its importance, several ways to describe and explain the color hue have been developed. All these developed methods can be named as color models and color spaces. A color model is a set of equations and procedures used to calculate a specific color, while a color space is the set of all the possible colors generated by a color model.

The additive, the subtractive, the perceptive, and the CIE models are among the most common color models that can be found. Other models were made especially for video and image transmission (like television broadcasting).

The additive color model consists of the mixture of two or more colors known as primary colors. The most representative model in this type is the Red-Green-Blue model or RGB. This model is widely spread, since it is the base of many electronic devices that display color (television, computers, mobile phones, etc.). This model is the simplest to implement, since it only required an amount of each primary color added over a black surface in order to obtain a given color hue [28].

The subtractive color model is similar to the additive color model; it uses a set of primary colors to obtain a color hue. The difference between additive and subtractive color models is that the subtractive model subtracts or blocks a certain amount of primary colors over a white surface instead of adding an amount of the primary colors over a black surface. The most representative subtractive color model is the Cyan-Magenta-Yellow model or CMY. This model is mostly used in printing processes.

The idea behind the perceptual color models is to create a similar process to the one that occurs when the brain processes an image. It is also referred to as a psychological interpretation of the colors. Basically this type of model splits the color in a hue component, a saturation component, and a light component. The most common models in this category are HSV, HSL, and HSI. These models have the characteristic of being represented by geometric figures, usually a cone, bicone, or cylinder. This type of geometric representation allows an easy manipulation of the color [29].

The CIE color models are those models created in the International Commission on Illumination (CIE). The CIE is a global nonprofit organization that gathers and shares information related to the science and art of light, color, vision, photobiology, and image technology [30].

This organization was the first to propose the creation of standardized color models. The most famous are CIE-RGB, CIE-XYZ, and CIE-Lab or Lab. The Lab color model is used in several image edition software tools, since it offers a robust gamut. The Lab color model has an illumination component “L” and two chromatic components “a” and “b.”

In the video and image transmissions, different color models are used. These models do not belong to a specific type and are related to the Lab color model. These models also split the color into an illumination component and two chromatic components but differ from Lab model in the way of the calculation of each component. The main purpose of these models is to adapt to the color image to the transmission processes (television broadcasting). Most of the calculations of the components in these models are meant to be used directly in analog television sets or cameras. The most common models in this category are YCbCr, YUV, and YDbDr.

2.1. Problems with the Color Models.

A reason for the existence of many colors models is that none of the color models is perfect. Any color model has failure points or is sometimes hard to manipulate. Due to the advantages and disadvantages, each color model has its own niche.

The additive and subtractive color models are easy to implement and understand, but they do not have a linear behavior. Also they are highly susceptible to illumination changes.

The CIE color models have robust gamut and the illumination component is separated from the chromatic components. The problems with the CIE color models are related to the nonlinearity behavior and the difficulty in the implementation of these models.

The models used for video and image transmissions are designed to be used in digital and analogic transmissions, so the implementation of these models for other purposes is complex.

The perceptual color models have the illumination component isolated and have a linear behavior. These models do not offer a robust gamut as the CIE models, since the perceptual color models only have one component for the chromatic information. Another issue with the perceptual color models comes from the equations used to calculate the hue and the saturation component. Hence, in case of having a white, black, or gray color, these two components could become undefined.

Equations (1), (2), and (3) are used for the HSV color model. Using these equations as example, it can be seen that in case of white or black or a gray tone the maximum and the minimum have the same value. In this case the H component (see (3)) becomes undefined. A usual workaround implemented in the most popular image processing libraries is to assign the value of zero when H is undefined. In the HSV color model, red hue has an H value of zero. So, it produces erroneous detections assigning the same hue to red, black, and gray tones.

\[ V = \max (R, G, B) \]  
\[ H = \tan^{-1} \left( \frac{S}{L - V} \right) \]  
\[ S = \sqrt{1 - \left( \frac{V}{L} \right)^2} \]  
\[ V = \frac{L + S}{2} \]
Mathematical Problems in Engineering

\[ S = \frac{\max(R, G, B) - \min(R, G, B)}{1 - |\max(R, G, B) + \min(R, G, B) - 1|} \]  

(2)

\[ H = \begin{cases} 
60 \times \frac{G - B}{\max(R, G, B) - \min(R, G, B)} & R = \max(R, G, B) \\
60 \times \frac{2 + (B - R)}{\max(R, G, B) - \min(R, G, B)} & G = \max(R, G, B) \\
60 \times \frac{4 + (R - G)}{\max(R, G, B) - \min(R, G, B)} & B = \max(R, G, B) 
\end{cases} \]  

(3)

2.2. Alternative Color Models. Due to the issues present in color models, some proposals have arrived in order to alleviate the problems found. Some of these alternative color models are variants from the existing color models, and the creation is meant to address a specific issue or to create an easier implementation of the model.

The normalized RGB or n-RGB was created specifically to help the RGB color model to deal with the illumination changes. Illumination is one of the most serious issues when color detection is performed. So the main idea behind the normalization is to use a percentage of the primary color instead of an amount. Theoretically, illumination modifies proportionality each color component, so the RBG color (50, 100, and 150) should have the same color hue as the color (5, 10, and 15) but with different illumination.

\[ r = \frac{R}{R + G + B} \]

\[ g = \frac{G}{R + G + B} \]  

(4)

\[ b = \frac{B}{R + G + B} \]

Equations (4) are used to calculate the n-RGB color space. The n-RGB space mitigates the effect of shadows and shines but also it could reduce the detection precision [31].

Another technique to improve detection processes and avoid the effect of illumination is to ignore the illumination component. This is usually done in perceptual color models and Lab-like color models, where the illumination component can be split. By applying this partial selection of components from the color models in color segmentation, some interference coming from unnecessary data like illumination or saturation can be avoided. Also as the information input from the color model is reduced, the segmentation and identification process is accelerated in the classification algorithms.

The most common cases are from the perceptual models, where the \( H \) and \( S \) components [32–34] or the \( H \) and \( V \) components [35] or only the \( H \) component [36, 37] or a mixture between the components [38] is used.

Another case is the partial usage of the Lab color model, where the \( L \) component is excluded, using only the chromatic components to perform the color detection [39].

3. Adapting HSV Color Model to \( K \)-Means

\( K \)-Means is an algorithm classified under the unsupervised learning category. Unsupervised learning algorithms are capable of discovering structures and relationships by themselves just using the input data [40].

The \( K \)-mean algorithm is commonly used in clustering processes. The algorithm was introduced by MacQueen in 1967 [41], even though the idea was conceived in 1957. The public disclosure of the algorithm was not done until 1982 [42]. The \( K \)-means algorithm is an iterative method that selects \( k \) random clusters centroids. In every iteration, the centroids are adjusted using the closest data points to each centroid. The algorithm ends when a defined iteration has been executed or a desired minimum data-centroid distance has been found. This behavior makes the \( K \)-means be referred to as an expectation maximization algorithm variant.

\( K \)-Means algorithm has some variations that are meant to improve the quality of the resulting segmentation; some popular examples are fuzzy \( C \)-means and \( K \)-means++.

The \( K \)-means algorithm does not always generate good results, mostly due to the random cluster centroid initialization. This random initialization generates problems like the case of having two cluster centroids being defined too close to each other. This would result in the misclassification of one group of related items in two different clusters. Another case is when a cluster centroid is defined far from the real data related group centroid. The random defined cluster centroid could never reach the real centroid in the amount of defined iterations.

These kinds of problem motivate researchers to propose improvements to the original \( K \)-means algorithm. Arthur and Vassilvitskii proposed an improvement to the \( K \)-means algorithm, focusing on the initialization process; they called their algorithm \( K \)-means++ [43]. Basically \( K \)-means++ uses a simple probabilistic approach to calculate the initial cluster centroids by obtaining the probability of how well a given point performs as a possible centroid.

Due to the advantages and the ease of implementation, this paper uses \( K \)-means++ in order to create more accurate results in the clustering process.

Image segmentation by using machine learning has been developed in many papers and works. We can find some recent works using neural networks [44–46], Gaussian mixture model [47, 48], support vector machine [49–51], and support vector machine with \( K \)-means family based training [52, 53]. Even though \( K \)-means algorithm is old, it is still used in image segmentation due to its ease of implementation [39, 52–54].

As it was mentioned, \( HSV \) produces undefined values when a black or white or gray tone is present in the image. This would discourage the usage of this model or using it under the premise that sometimes the color detection will fail under the previously mentioned circumstances.

An additional issue comes into account when a distance-based algorithm like \( K \)-means is used. This comes from the fact that the \( H \) component is measured like the angle of a circumference. The usage of this angle representation implies that the next \( H \) value for 359 is 0. An algorithm like \( K \)-means
detects that 359 and 0 are far from each other and they should be classified in different clusters.

The previous issue could be solved by adding additional logic in the distance measurement method by adding rules to avoid the miscalculation of the distance in the $H$ component. Using this approach could produce an excessive increase in the computational work.

The implementation of $K$-means++ does not guarantee the correct classification of the input items. $K$-Means++ improves the general outcome by providing a better start. A better start helps to find the best solution faster and/or to reduce the amount of erroneous clusters definition.

The present paper proposes an adaptation to the $HSV$ model in order to overcome the previously mentioned issues while providing basis to improve the result in the $K$-means++ algorithm.

Most of the image libraries use the 1-byte per component representation, which implies that the value for the $H$ component from the $HSV$ color model must be adapted to fit in the given space. The preferred approach is to take the half of the $H$ component (divided by two), so the $H$ component goes from 0 to 179. Regarding the $S$ and $V$ components, each has a range from 0 to 255.

This approach is preferred, since if the 2-byte representation is used, the amount of memory required to process the image increases significantly.

3.1. Modified $HSV$ Color Model Calculation. The proposed adaptation applied to $HSV$ color model addressed two important issues: the undefined values produced by the $H$ component equation [see (3)] and the discontinuity in this component when it changes from 359 to 0.

The proposed change consists in modifying the way $H$ is defined, especially when it becomes undefined. The idea is to take advantage of the unassigned values in the $H$ byte. The $H$ byte covers a range only from 0 to 179, so 180 to 255 are unassigned empty values. Basically, instead of assigning $H$ to zero when white, black, and grayscale colors are detected, these colors are assigned to a range of the empty values.

The selected range in this work for the black, white, and gray tones is from 200 to 255. The starting point was selected in a way that the separation from the last $H$ value (179) is easily detected by $K$-means. Lower starting points can be chosen, but 200 was selected in order to remark the separation between the possible clusters. The two areas defined in the $H$ component match the definition of chromatic and achromatic regions. In this case, the chromatic region is defined from 0 to 179 and the achromatic region from 200 to 255. Using the achromatic and chromatic definitions [55] adapted for $HSV$ color space [Figure 1], the following can be stated:

1. Color hue ($H$) is meaningless when the illumination ($V$) is very low (turns to black).
2. Color hue ($H$) is unstable when the saturation ($S$) is very low (turns to gray).
3. When saturation ($S$) is low and illumination ($V$) is high, the color hue ($H$) is meaningless (turns to white).

In all the cases when $H$ becomes unstable or meaningless, the achromatic zone is considered; otherwise the chromatic zone is considered. The procedure marks as achromatic an $HSV$ value when the saturation is low or the illumination is low. The previous statement requires the definition of a threshold ($th$) that indicates when the $HSV$ value is achromatic. Using the definition, this threshold must be applied to the $S$ and $V$ components and it will indicate when $S$ or $V$ values are low enough to consider $H$ meaningless.

Using the previous concepts in the $H$ equation [see (3)] results in the following equation:

$$H = \begin{cases} 
60 \left( \frac{G - B}{\max(R,G,B) - \min(R,G,B)} \right) & \text{(S and V above th)} \\
60 \left( \frac{2 + (B - R)}{\max(R,G,B) - \min(R,G,B)} \right) & \text{(S or V below th)} \\
200 + \left( \frac{V}{255} \right) \ast 55 & \text{ } 
\end{cases}$$

$R = \max(R,G,B)$

$G = \max(R,G,B)$

$B = \max(R,G,B)$

Figure 1: Achromatic and chromatic areas for $HSV$ color space.
As it was explained in the previous sections, $H$ is the angle of a circumference, so the next value from 359 is zero. In the case of the 1 byte per component representation, the next value from 179 is zero. The color hue corresponding to this discontinuity area is the red tone. This issue produces the creation of two separate clusters for the red color, even if the hues are almost the same.

Some approaches can be implemented in order to correct the possible erroneous creation of clusters. Rules in the distance measurement in the $K$-means++ algorithm when the $H$ value is close to the discontinuity region can be implemented. But this generates an important load in the computer work.

This paper proposes the usage of shifted angles for the $H$ component. This means that the discontinuity can be placed in another color hue. The creation of two shifted angles representations for $H$ is proposed, so they can be combined and eliminate the discontinuity issue. The original $H$ has the discontinuity in the red hue, the first shifted $H$ ($H_{120}$) has the discontinuity in the green hue ($120^\circ$), and the second shifted $H$ ($H_{240}$) has the discontinuity in the blue hue ($240^\circ$). As can be seen, the shift operation is done in evenly defined amounts ($120^\circ$ from each $H$ component) [Figure 2].

In the case of the 1-byte representation, the shift amount is 60 for $H_{120}$ and 120 for the $H_{240}$ (half of the original values).

The original and the two shifted $H$ components are meant to be processed by $K$-means++. This would seem to generate significant extra computational work, but this process is meant to solve the discontinuity issue in the HSV and also improve the classification performed by the $K$-means++ algorithm. This is explained in detail in the next section where the complete improvement process is exposed.

The work done in this paper uses the HSV partial model approach to eliminate the effect of illumination in component $V$ from the color process. It also excludes the $S$ component, since the main purpose is the segmentation or classification by color hue. The selection of only one component speeds up the process done by $K$-means++ by reducing the complexity in the input.

The complete process to create the input for the $K$-means++ is as in Pseudocode 1 in order to calculate $H$, $H_{120}$, and $H_{240}$ for each pixel in the RGB input image.

Since the pseudocode is set to operate in a 1 byte per channel model, the $H$ values are in the range of 0–179. After obtaining the $K$-means++ clusters, a matching and grouping operation is performed. The reason for the matching and grouping operation is to detect similar clusters and group them together. The idea is that if a cluster group has two or more members, this cluster group has more probability to be a real cluster. This approach makes those cluster groups that were affected by the discontinuity be detected and ignored as

\begin{verbatim}
set threshold_v, threshold_s //predefined thresholds
set H_Entry, H120_Entry, H240_Entry
for each pixel in RGB do
    set V with Eq. (1)
    set S with Eq. (2)
    if V > threshold_v and S > threshold_s then
        set H with the first part of Eq. (5)
        if H > = 120 then
            set H120 equals H - 120
        else
            set H120 equals H + 60
        if H > = 60 then
            set H240 equals H - 60
        else
            set H240 equals H + 120
    else
        set H with the second part of Eq. (5)
        set H120 equals H
        set H240 equals H
    add H, H120, H240 in H_Entry, H120_Entry, H240_Entry
for each entry in [H_Entry, H120_Entry, H240_Entry] do
    execute K-Means with entry
\end{verbatim}

Pseudocode 1: Modified HSV color model pseudocode for $K$-means++. 

As it was explained in the previous sections, $H$ is the angle of a circumference, so the next value from 359 is zero. In the case of the 1 byte per component representation, the next value from 179 is zero. The color hue corresponding to this discontinuity area is the red tone. This issue produces the creation of two separate clusters for the red color, even if the hues are almost the same. Some approaches can be implemented in order to correct the possible erroneous creation of clusters. Rules in the distance measurement in the $K$-means++ algorithm when the $H$ value is close to the discontinuity region can be implemented. But this generates an important load in the computer work.

This paper proposes the usage of shifted angles for the $H$ component. This means that the discontinuity can be placed in another color hue. The creation of two shifted angles representations for $H$ is proposed, so they can be combined and eliminate the discontinuity issue. The original $H$ has the discontinuity in the red hue, the first shifted $H$ ($H_{120}$) has the discontinuity in the green hue ($120^\circ$), and the second shifted $H$ ($H_{240}$) has the discontinuity in the blue hue ($240^\circ$). As can be seen, the shift operation is done in evenly defined amounts ($120^\circ$ from each $H$ component) [Figure 2].

In the case of the 1-byte representation, the shift amount is 60 for $H_{120}$ and 120 for the $H_{240}$ (half of the original values).

The original and the two shifted $H$ components are meant to be processed by $K$-means++. This would seem to generate significant extra computational work, but this process is meant to solve the discontinuity issue in the HSV and also improve the classification performed by the $K$-means++ algorithm. This is explained in detail in the next section where the complete improvement process is exposed.

The work done in this paper uses the HSV partial model approach to eliminate the effect of illumination in component $V$ from the color process. It also excludes the $S$ component, since the main purpose is the segmentation or classification by color hue. The selection of only one component speeds up the process done by $K$-means++ by reducing the complexity in the input.
they usually contain only one member. The shift operation forces the discontinuity to affect a different hue, so the other tones are not affected.

For instance, the original $H$ component is affected in the red hue by the discontinuity, so the $K$-means++ algorithm would produce a split cluster in this affected hue area. But $H120$ and $H240$ are not affected in the red hue by the discontinuity, so the $K$-means++ algorithm would produce the correct cluster for a red hue.

Another reason to apply $K$-means++ to three versions of the same information is to improve the cluster quality. Even though $K$-means++ is an improvement over $K$-means, still certain amount of the process relies on randomness, producing sometimes a not so accurate initial centroid. Performing the same classification several times helps to enforce the results by taking those groups of similar clusters with more members as the most probable real clusters. The process for the $K$-means++ clustering and grouping can be described as in Pseudocode 2.

The purpose of the matching and grouping is to take those clusters with a high similarity and group them together. This could seem to be a trivial task, but its implications make it a complex procedure. The simplest approach is to only use the Euclidean distance between the cluster centroids and group the clusters with the lowest distance [54].

The previous approach could not always produce the best result, due to the variance of the elements in the clusters or the cases of missing clusters or the case where a cluster is divided. An algorithm proposed to match clusters alleviating the possible issues found is the Mixed Edge Cover (MEC) [56]. The MEC algorithm calculates the similarities and dissimilarities between the clusters using a distance measurement that eliminates the variance issues. The Mahalanobis distance between each cluster element can be used for this purpose [57]. So this paper uses the Mahalanobis distance as the similarity measurement between the clusters.

Bagging is a technique used in machine learning, where several versions of a predictor algorithm or a classifier algorithm are used to generate a new predictor or classifiers. Usually this is done by averaging the results in predictors, and, in the case of the classifiers, a voting process is performed [58]. The proposed procedure creates groups of similar clusters and then eliminates the groups with fewer members using a voting system.

After the voting is finished in the first bagging process, a second bagging process is executed in order to create a unified cluster from each selected cluster group. The voting in the second bagging process creates a cluster from those cluster items common in two or more clusters. So if a cluster item appears in just one cluster, this is considered as noise data or a misclassified pixel.

3.2. Proposed Method’s Theoretical Ground. The issues found in color spaces are related to discontinuities and nonlinear behaviors. Classification methods based on distances like $K$-means cannot handle these issues correctly when a color classification is required. The $HSV$ model has a linear behavior in the color hue component $H$ but suffers from a discontinuity when it changes from 359 to 0.

The proposed change moves the discontinuity to different values. It creates two additional versions of the $H$ component ($H120$ and $H240$), where the discontinuity occurs in different colors hues. Performing a clustering operation on one of the components, $H$, $H120$, or $H240$, produces clusters, where the discontinuity could be manifested in the form of real clusters divided into two clusters. This issue does not exist in the clusters coming from the other two components.

Pseudocode 2: Cluster grouping.

```plaintext
set groups, result
for each clusters in [clusters_H, clusters_H120, clusters_H240] do
  for each cluster in clusters do
    if length(groups) == 0 then
      set new_group
      add cluster in new_group
      add new_group in groups
      continue
    for each group in groups do
      if distance(cluster, group) < threshold then
        add cluster in group
      else
        set new_group
        add cluster in new_group
        add new_group in groups
    for each group in groups do
      if length(group) == 1 then
        delete group
      else
        add merge(group) in result
```
Performing cluster matching and grouping over all the clusters coming from all the components generates groups, where if it contains 2 or more elements it can be considered like a real cluster; otherwise the group can be ignored. So, this process alleviates the discontinuity issue found in the $H$ component. Additionally splitting the chromatic and achromatic values allows reducing the effect of shines and reflections that can lead to incorrect classification. Also it avoids the issues happening in the $HSV$ when the pixel is a shade of gray ($H$ becomes undefined using (3)). Instead of setting the $H$ value to 0 in this case, the proposed improvement uses an unassigned value range in the $H$ component. This facilitates the clustering process by having a specific region for the chromatic tone and a separated region for the achromatic ones.

All the changes performed in the proposed improvement eliminate the discontinuity and provide a more linear input data for the $K$-means algorithm. Additionally the changes allow mitigating shadow, shines, and reflection which alter the perception of color tones. This has a positive effect in the classification performed by $K$-means compared to classification performed using other color models, producing more accurate results.

### 4. Testing and Experimentation

In the testing process, the proposed $HSV$ model is tested against other color models and the original $HSV$. The testing dataset comes from two sources, mainly the BSDS500 and a couple of images from the Free Images website [59]. From the first dataset, the ground truth is taken from the files inside the dataset, while in the second test some ground truth images were created. All the color models are processed by the $K$-means++ algorithm which is set to find 4 or 5 clusters (usually the amount of segmented objects found in the BSDS500 dataset).

Once the clusters are obtained for each tested color model, they will be evaluated using statistical measurements. Usually measurements like specificity [see (6)], sensitivity [see (7)], and accuracy [see (8)] are used in segmentation tests. These measurements use parameters like True Positive (TP, number of pixels included in the segmented object which are correctly classified), True Negative (TN, number of pixels not included in the segmented object which are correctly classified), False Positive (FP, number of pixels included in the segmented object which are incorrectly classified), and False Negative (FN, number of pixels not included in the segmented object which are incorrectly classified). This work uses balanced accuracy [see (9)] [60] in order to use the accuracy as an overall measurement, in which the specificity and sensitivity are added in certain proportion by applying the adjustment parameters $\alpha$ and $\beta$ (usually these parameters are set to 0.5).

\[
SPC = \frac{TN}{TN + FP} \quad (6)
\]

\[
SEN = \frac{TP}{TP + FN} \quad (7)
\]

\[
ACC = \frac{(TP + TN)}{(TP + FP + FN + TN)} 
\]

\[
Bacc = \alpha \times SEN + \beta \times SPC. \quad (9)
\]

Balanced accuracy measurement should give an overview of how well the test is performing, but unfortunately this is not always possible. Basically, since the parameters depend on the amount of pixels in the segmented object or outside of it, it could lead to a high balanced accuracy if a high value in specificity or sensitivity is calculated. In order to avoid this case, the parameters $\alpha$ [see (10)] and $\beta$ [see (11)] are calculated considering the number of pixels in the segmented object (VP) and the pixels outside the segmented object or background (BP) [61].

\[
\alpha = \frac{VP}{(VP + BP)} \quad (10)
\]

\[
\beta = \frac{BP}{(VP + BP)}. \quad (11)
\]

The selected color models used for the comparison are those that appear commonly in the literature:

(i) RGB
(ii) nRGB
(iii) HSV
(iv) HS Original
(v) $H$ Original ($H$ Orig)
(vi) $H$ Modified ($H$ Mod)
(vii) $Lab$
(viii) $ab$
(ix) YCbCr
(x) CbCr

The test for each color model is executed 20 times, taking the best result and the average. So a more reliable statistical comparison can be made among the color models using $K$-means++ algorithm.

In order to apply the modified $HSV$ model, it is necessary to define a threshold value, $th$ [see (5)], so the chromatic and achromatic regions can be placed in the $H$ component. After performing some tests over a group of images, it was observed that setting the threshold around 30% of the value for the $V$ and $S$ components produced the best result in the segmentation, so this threshold will be used in the tests.

Also a set of images from the selected datasets sources is selected to perform the comparison. The BSDS500 dataset is selected mainly to perform object segmentation. Color segmentation algorithms can solve the segmentation task in some of the proposed scenarios in the BSDS500 dataset. So, taking that in account, a subset of images, where the ground truth is close to color segmentation, was selected.

In order to provide more comparison data regarding the behavior of the proposed improvement, another clustering algorithm is used in the tests. Gaussian mixture model performs in a similar way to $K$-means, so implementation of
the GMM using the expectation maximization (EM) method is used to provide a comparison with a different algorithm.

For both algorithms, the conditions are similar; both perform 200 iterations. Regarding the starting point for the Gaussians in GMM, the $K$-means++ initialization algorithm is used to set the initial mean and standard deviation. It creates a scenario where a fair comparison can be made.

4.1. Test Results. A few images from the selected test dataset are exposed in order to demonstrate how every color performs in the segmentation done by $K$-means++.

The images in Figure 3 are selected to show visually the segmentation done by each of the selected color models and some metrics showing the performance.

In Tables 1–12 in the first row the ground truth clusters coming from each of the images from Figure 3 are given. The following rows contain the results from each of the segmentations produced using each of the selected color models. In the last columns, some metrics measuring the performance are given:

(i) Mean BAcc: the average balanced accuracy using all the data from all the clusters and all the iterations
(ii) Best BAcc: the best individual balanced accuracy for one cluster occurring in the iterations
(iii) Mean sen.: the average sensitivity
(iv) Mean spe.: the average specificity
(v) Avg. time: the running time for the algorithm given in seconds. This is used to measure the CPU time needed. For the proposal, the time measurement is divided into 2 phases: one for the clustering time ($C$) and another for the bagging time ($B$)

From the results in Tables 1–12, it can be seen that the proposed improvement is most of the time in the first place.
Table 1: Segmentation performance results for image "a" using K-means++.  

<table>
<thead>
<tr>
<th>K-Means</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Metrics</th>
</tr>
</thead>
</table>
| Ground truth | ![image](image1) | ![image](image2) | ![image](image3) | ![image](image4) | Best BAcc.: 0.85028  
Mean BAcc.: 0.74197  
Mean sen.: 0.62917  
Mean spe.: 0.81776  
Avg. time: 3.8598 |
| RGB      | ![image](image5) | ![image](image6) | ![image](image7) | ![image](image8) | Best BAcc.: 0.93637  
Mean BAcc.: 0.82969  
Mean sen.: 0.53766  
Mean spe.: 0.88299  
Avg. time: 1.5083 |
| nRGB     | ![image](image9) | ![image](image10) | ![image](image11) | ![image](image12) | Best BAcc.: 0.86040  
Mean BAcc.: 0.7717  
Mean sen.: 0.58555  
Mean spe.: 0.85133  
Avg. time: 3.5174 |
| Lab      | ![image](image13) | ![image](image14) | ![image](image15) | ![image](image16) | Best BAcc.: 0.95228  
Mean BAcc.: 0.84635  
Mean sen.: 0.71681  
Mean spe.: 0.90009  
Avg. time: 1.9523 |
| ab       | ![image](image17) | ![image](image18) | ![image](image19) | ![image](image20) | Best BAcc.: 0.79657  
Mean BAcc.: 0.75988  
Mean sen.: 0.64811  
Mean spe.: 0.83866  
Avg. time: 3.28 |
| YCrCb    | ![image](image21) | ![image](image22) | ![image](image23) | ![image](image24) | Best BAcc.: 0.94294  
Mean BAcc.: 0.81556  
Mean sen.: 0.61425  
Mean spe.: 0.87505  
Avg. time: 1.5412 |
| CrCb     | ![image](image25) | ![image](image26) | ![image](image27) | ![image](image28) | Best BAcc.: 0.91797  
Mean BAcc.: 0.85888  
Mean sen.: 0.72094  
Mean spe.: 0.90462  
Avg. time: 1.4304 |
| HSV      | ![image](image29) | ![image](image30) | ![image](image31) | ![image](image32) | Best BAcc.: 0.91816  
Mean BAcc.: 0.83095  
Mean sen.: 0.50944  
Mean spe.: 0.89749  
Avg. time: 1.333 |
| HS       | ![image](image33) | ![image](image34) | ![image](image35) | ![image](image36) | Best BAcc.: 0.91808  
Mean BAcc.: 0.76817  
Mean sen.: 0.22684  
Mean spe.: 0.96200  
Avg. time: 0.7939 |
| H Orig   | ![image](image37) | ![image](image38) | ![image](image39) | ![image](image40) | Best BAcc.: 0.96551  
Mean BAcc.: 0.87931  
Mean sen.: 0.77377  
Mean spe.: 0.90882  
Avg. time (C): 1.4847  
Avg. time (B): 2.3388 |
Table 2: Segmentation performance results for image "a" using GMM.

<table>
<thead>
<tr>
<th>GMM</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Metrics</th>
</tr>
</thead>
</table>
| RGB | ![Image](image1.png) | ![Image](image2.png) | ![Image](image3.png) | ![Image](image4.png) | Best BAcc: 0.87485  
Mean BAcc: 0.76002  
Mean sen.: 0.65702  
Mean spe.: 0.82868  
Avg. time: 7.4548 |
| nRGB | ![Image](image5.png) | ![Image](image6.png) | ![Image](image7.png) | ![Image](image8.png) | Best BAcc: 0.93637  
Mean BAcc: 0.82969  
Mean sen.: 0.53766  
Mean spe.: 0.88299  
Avg. time: 5.14159 |
| Lab | ![Image](image9.png) | ![Image](image10.png) | ![Image](image11.png) | ![Image](image12.png) | Best BAcc: 0.86040  
Mean BAcc: 0.7717  
Mean sen.: 0.58555  
Mean spe.: 0.85133  
Avg. time: 4.5137 |
| ab | ![Image](image13.png) | ![Image](image14.png) | ![Image](image15.png) | ![Image](image16.png) | Best BAcc: 0.95228  
Mean BAcc: 0.84635  
Mean sen.: 0.71681  
Mean spe.: 0.90009  
Avg. time: 3.1992 |
| YCrCb | ![Image](image17.png) | ![Image](image18.png) | ![Image](image19.png) | ![Image](image20.png) | Best BAcc: 0.79657  
Mean BAcc: 0.75988  
Mean sen.: 0.64811  
Mean spe.: 0.83836  
Avg. time: 4.7854 |
| CrCb | ![Image](image21.png) | ![Image](image22.png) | ![Image](image23.png) | ![Image](image24.png) | Best BAcc: 0.94294  
Mean BAcc: 0.81556  
Mean sen.: 0.61425  
Mean spe.: 0.87505  
Avg. time: 3.1574 |
| HSV | ![Image](image25.png) | ![Image](image26.png) | ![Image](image27.png) | ![Image](image28.png) | Best BAcc: 0.91797  
Mean BAcc: 0.85888  
Mean sen.: 0.72094  
Mean spe.: 0.90462  
Avg. time: 4.2984 |
| HS | ![Image](image29.png) | ![Image](image30.png) | ![Image](image31.png) | ![Image](image32.png) | Best BAcc: 0.91816  
Mean BAcc: 0.83095  
Mean sen.: 0.50944  
Mean spe.: 0.89749  
Avg. time: 3.4293 |
| H Orig | ![Image](image33.png) | ![Image](image34.png) | ![Image](image35.png) | ![Image](image36.png) | Best BAcc: 0.91808  
Mean BAcc: 0.76817  
Mean sen.: 0.22684  
Mean spe.: 0.96200  
Avg. time: 2.82860 |

And when it is not in the first place it is really close to the first place. Visually it can be seen also that the closest result to the ground truth images is the modified model. 

Aside from the previous tests, some additional tests over other images were executed. In Tables 13 and 14, the means of the results from all the tests are displayed. In addition to the measures exposed in Tables 13 and 14, two new measures were added:

(i) Best mean BAcc: the best average of an iteration  
(ii) Worst mean BAcc: the worst average from an iteration

After summarizing all the measurements, it can be noted that the proposed modification has a positive effect across the tests in $K$-means++. Performing a comparison against the results coming from GMM, the test performed over $ab$ (partial model from $Lab$) has a better performance in the
Table 3: Segmentation performance results for image “b” using K-means++.

<table>
<thead>
<tr>
<th>K-Means</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground truth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc.: 0.99587</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.98777</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen.: 0.93760</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe.: 0.97967</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 1.75579</td>
</tr>
<tr>
<td>RGB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc.: 0.99518</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.99005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen.: 0.92404</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe.: 0.98184</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 1.8469</td>
</tr>
<tr>
<td>nRGB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc.: 0.99780</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.96863</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen.: 0.72131</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe.: 0.92525</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 1.5184</td>
</tr>
<tr>
<td>Lab</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc.: 0.99688</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.96830</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen.: 0.61425</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe.: 0.92157</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 1.286</td>
</tr>
<tr>
<td>ab</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc.: 0.99668</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.96988</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen.: 0.61425</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe.: 0.92157</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 1.286</td>
</tr>
<tr>
<td>YCrCb</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc.: 0.97252</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.90427</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen.: 0.82520</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe.: 0.94918</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 1.41410</td>
</tr>
<tr>
<td>CrCb</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc.: 0.97236</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.90964</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen.: 0.81605</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe.: 0.95333</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 1.1394</td>
</tr>
<tr>
<td>HSV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc.: 0.92121</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.84475</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen.: 0.75755</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe.: 0.86620</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 1.3613</td>
</tr>
<tr>
<td>HS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc.: 0.99589</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.99259</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen.: 0.94847</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe.: 0.98287</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time (C): 1.7464</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time (B): 3.166</td>
</tr>
</tbody>
</table>
Table 4: Segmentation performance results for image “b” using GMM.

<table>
<thead>
<tr>
<th>GMM</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc: 0.99833</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.98109</td>
</tr>
<tr>
<td>RGB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen: 0.96164</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe: 0.98576</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 4.6552</td>
</tr>
<tr>
<td>nRGB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc: 0.99846</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.95547</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen: 0.61632</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe: 0.90544</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 6.5616</td>
</tr>
<tr>
<td>Lab</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc: 0.99733</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.98261</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen: 0.88530</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe: 0.98876</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 3.3779</td>
</tr>
<tr>
<td>ab</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc: 0.99838</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.97665</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen: 0.84241</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe: 0.95545</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 3.10470</td>
</tr>
<tr>
<td>YCrCb</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc: 0.99833</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.98288</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen: 0.91770</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe: 0.98878</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 3.8567</td>
</tr>
<tr>
<td>CrCb</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc: 0.99829</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.96058</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen: 0.69106</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe: 0.91341</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 4.0068</td>
</tr>
<tr>
<td>HSV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc: 0.99818</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.91195</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen: 0.84421</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe: 0.96268</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 3.47339</td>
</tr>
<tr>
<td>HS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc: 0.97405</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.91007</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen: 0.83561</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe: 0.96637</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 3.71440</td>
</tr>
<tr>
<td>H Orig</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc: 0.92188</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.83645</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen: 0.75261</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe: 0.86185</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 2.5643</td>
</tr>
</tbody>
</table>

worse BAcc and the best BAcc measurements, but in the average (mean BAcc) the proposed method has a better score.

In order to correctly validate the experimental results, a statistical test is performed over the balanced accuracy observed in the comparison results. In this case, the Wilcoxon test is conducted. The Wilcoxon test is a nonparametric test used when a normal distribution cannot be guaranteed in the data. Its null hypothesis, over two different results, considers that the two compared populations come from the same distribution [62]. The Wilcoxon method has been commonly used to compare algorithms behaviors in order to verify which one has a better performance using normalized values (from 0 to 1) [63]. The Wilcoxon signed-rank sum is set to use the right tail. Under such conditions, the alternative hypothesis is that the first population data has a higher median than the second population data. Therefore, the first
Table 5: Segmentation performance results for image “c” using K-means++.

<table>
<thead>
<tr>
<th>K-Means</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Cluster 5</th>
<th>Metrics</th>
</tr>
</thead>
</table>
| Ground truth | ![Image](image1) | ![Image](image2) | ![Image](image3) | ![Image](image4) | ![Image](image5) | Best BAcc: 0.958333  
Mean BAcc: 0.844500  
Mean sen.: 0.588411  
Mean spe.: 0.903700  
Avg. time: 3.72890 |
| RGB | ![Image](image6) | ![Image](image7) | ![Image](image8) | ![Image](image9) | ![Image](image10) | Best BAcc: 0.979227  
Mean BAcc: 0.904491  
Mean sen.: 0.645666  
Mean spe.: 0.940431  
Avg. time: 2.67930 |
| nRGB | ![Image](image11) | ![Image](image12) | ![Image](image13) | ![Image](image14) | ![Image](image15) | Best BAcc: 0.959470  
Mean BAcc: 0.846650  
Mean sen.: 0.586510  
Mean spe.: 0.905040  
Avg. time: 2.82420 |
| Lab | ![Image](image16) | ![Image](image17) | ![Image](image18) | ![Image](image19) | ![Image](image20) | Best BAcc: 0.979227  
Mean BAcc: 0.904491  
Mean sen.: 0.645666  
Mean spe.: 0.940431  
Avg. time: 2.67930 |
| ab | ![Image](image21) | ![Image](image22) | ![Image](image23) | ![Image](image24) | ![Image](image25) | Best BAcc: 0.961480  
Mean BAcc: 0.849340  
Mean sen.: 0.648110  
Mean spe.: 0.906700  
Avg. time: 3.27290 |
| YCrCb | ![Image](image26) | ![Image](image27) | ![Image](image28) | ![Image](image29) | ![Image](image30) | Best BAcc: 0.961930  
Mean BAcc: 0.849340  
Mean sen.: 0.648110  
Mean spe.: 0.906700  
Avg. time: 3.27290 |
| CrCb | ![Image](image31) | ![Image](image32) | ![Image](image33) | ![Image](image34) | ![Image](image35) | Best BAcc: 0.983380  
Mean BAcc: 0.937940  
Mean sen.: 0.866000  
Mean spe.: 0.952680  
Avg. time: 1.09140 |
| HSV | ![Image](image36) | ![Image](image37) | ![Image](image38) | ![Image](image39) | ![Image](image40) | Best BAcc: 0.974740  
Mean BAcc: 0.888777  
Mean sen.: 0.490190  
Mean spe.: 0.959100  
Avg. time: 3.01899 |
| HS | ![Image](image41) | ![Image](image42) | ![Image](image43) | ![Image](image44) | ![Image](image45) | Best BAcc: 0.971210  
Mean BAcc: 0.863650  
Mean sen.: 0.422460  
Mean spe.: 0.945200  
Avg. time: 3.02090 |
| H Orig | ![Image](image46) | ![Image](image47) | ![Image](image48) | ![Image](image49) | ![Image](image50) | Best BAcc: 0.964330  
Mean BAcc: 0.928630  
Mean sen.: 0.739950  
Mean spe.: 0.947770  
Avg. time: 0.66040 |
| H Mod | ![Image](image51) | ![Image](image52) | ![Image](image53) | ![Image](image54) | ![Image](image55) | Best BAcc: 0.989930  
Mean BAcc: 0.943210  
Mean sen.: 0.765180  
Mean spe.: 0.968080  
Avg. time (C): 1.94140  
Avg. time (B): 2.52290 |
### Table 6: Segmentation performance results for image “c” using GMM.

<table>
<thead>
<tr>
<th>GMM</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Cluster 5</th>
<th>Metrics</th>
</tr>
</thead>
</table>
| RGB | ![Image](image1) | ![Image](image2) | ![Image](image3) | ![Image](image4) | ![Image](image5) | Best BAcc.: 0.97231  
Mean BAcc: 0.86137  
Mean sen.: 0.64875  
Mean spe.: 0.91404  
Avg. time: 7.4137 |
| nRGB | ![Image](image6) | ![Image](image7) | ![Image](image8) | ![Image](image9) | ![Image](image10) | Best BAcc.: 0.95474  
Mean BAcc: 0.89090  
Mean sen.: 0.50531  
Mean spe.: 0.94607  
Avg. time: 5.039 |
| Lab | ![Image](image11) | ![Image](image12) | ![Image](image13) | ![Image](image14) | ![Image](image15) | Best BAcc.: 0.97013  
Mean BAcc: 0.86512  
Mean sen.: 0.65040  
Mean spe.: 0.91608  
Avg. time: 4.7831 |
| ab | ![Image](image16) | ![Image](image17) | ![Image](image18) | ![Image](image19) | ![Image](image20) | Best BAcc.: 0.98512  
Mean BAcc: 0.96448  
Mean sen.: 0.90266  
Mean spe.: 0.97358  
Avg. time: 3.16470 |
| YCrCb | ![Image](image21) | ![Image](image22) | ![Image](image23) | ![Image](image24) | ![Image](image25) | Best BAcc.: 0.97427  
Mean BAcc: 0.86622  
Mean sen.: 0.66280  
Mean spe.: 0.91694  
Avg. time: 5.13540 |
| CrCb | ![Image](image26) | ![Image](image27) | ![Image](image28) | ![Image](image29) | ![Image](image30) | Best BAcc.: 0.98533  
Mean BAcc: 0.94658  
Mean sen.: 0.88602  
Mean spe.: 0.95928  
Avg. time: 1.9721 |
| HSV | ![Image](image31) | ![Image](image32) | ![Image](image33) | ![Image](image34) | ![Image](image35) | Best BAcc.: 0.98307  
Mean BAcc: 0.88245  
Mean sen.: 0.52690  
Mean spe.: 0.94644  
Avg. time: 5.60439 |
| HS | ![Image](image36) | ![Image](image37) | ![Image](image38) | ![Image](image39) | ![Image](image40) | Best BAcc.: 0.98099  
Mean BAcc: 0.87426  
Mean sen.: 0.55223  
Mean spe.: 0.92978  
Avg. time: 4.122 |
| H Orig | ![Image](image41) | ![Image](image42) | ![Image](image43) | ![Image](image44) | ![Image](image45) | Best BAcc.: 0.96963  
Mean BAcc: 0.92605  
Mean sen.: 0.73088  
Mean spe.: 0.94629  
Avg. time: 1.90460 |

Population data has the balanced accuracy obtained by the proposed approach, while the second has the results obtained by the other color models using the K-means and the GMM algorithms (Tables 15 and 16).

As can be observed in Tables 15 and 16, all the tests reject the null hypothesis about the two data populations being the same using an alpha value of 0.05. The alternative hypothesis stating that the improved color model has a better outcome is selected.

Regarding the time measurements, the proposed method has a similar time to any 3-channel color model used in the tests for K-means++. But the total time used for the proposed improvements is higher when it is considered together with the bagging process. This is expected since the bagging
<table>
<thead>
<tr>
<th>K-Means</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Metrics</th>
</tr>
</thead>
</table>
| Ground truth | ![Image](image1.png) | ![Image](image2.png) | ![Image](image3.png) | ![Image](image4.png) | Best BAcc.: 0.95994  
Mean BAcc: 0.869935  
Mean sen: 0.75575  
Mean spe: 0.90860  
Avg. time: 1.76310 |
| RGB | ![Image](image5.png) | ![Image](image6.png) | ![Image](image7.png) | ![Image](image8.png) | Best BAcc.: 0.90987  
Mean BAcc: 0.81386  
Mean sen: 0.54885  
Mean spe: 0.87593  
Avg. time: 3.72300 |
| nRGB | ![Image](image9.png) | ![Image](image10.png) | ![Image](image11.png) | ![Image](image12.png) | Best BAcc.: 0.95932  
Mean BAcc: **0.88656**  
Mean sen: 0.78536  
Mean spe: 0.91907  
Avg. time: 1.2676 |
| Lab | ![Image](image13.png) | ![Image](image14.png) | ![Image](image15.png) | ![Image](image16.png) | Best BAcc.: 0.98018  
Mean BAcc: 0.81514  
Mean sen: 0.85177  
Mean spe: 0.80258  
Avg. time: 0.8469 |
| ab | ![Image](image17.png) | ![Image](image18.png) | ![Image](image19.png) | ![Image](image20.png) | Best BAcc.: 0.96415  
Mean BAcc: 0.86543  
Mean sen: 0.74899  
Mean spe: 0.90738  
Avg. time: 1.87110 |
| YCrCb | ![Image](image21.png) | ![Image](image22.png) | ![Image](image23.png) | ![Image](image24.png) | Best BAcc.: 0.98062  
Mean BAcc: 0.81118  
Mean sen: 0.84099  
Mean spe: 0.80067  
Avg. time: 0.9742 |
| CrCb | ![Image](image25.png) | ![Image](image26.png) | ![Image](image27.png) | ![Image](image28.png) | Best BAcc.: 0.90939  
Mean BAcc: 0.78772  
Mean sen: 0.45141  
Mean spe: 0.85449  
Avg. time: 1.25549 |
| HSV | ![Image](image29.png) | ![Image](image30.png) | ![Image](image31.png) | ![Image](image32.png) | Best BAcc.: 0.92218  
Mean BAcc: 0.78140  
Mean sen: 0.38454  
Mean spe: 0.87325  
Avg. time: 1.144 |
| HS | ![Image](image33.png) | ![Image](image34.png) | ![Image](image35.png) | ![Image](image36.png) | Best BAcc.: 0.91348  
Mean BAcc: 0.80163  
Mean sen: 0.48664  
Mean spe: 0.86596  
Avg. time: 0.5639 |
| H Orig | ![Image](image37.png) | ![Image](image38.png) | ![Image](image39.png) | ![Image](image40.png) | Best BAcc.: 0.97414  
Mean BAcc: 0.87874  
Mean sen: 0.64856  
Mean spe: 0.92610  
Avg. time (C): 1.9026  
Avg. time (B): 2.6005 |
| H Mod | ![Image](image41.png) | ![Image](image42.png) | ![Image](image43.png) | ![Image](image44.png) | |
operations involve a significant amount of computation, especially when the clusters need to be matched as mentioned in Section 3. It is worth mentioning that, using different cluster matching procedures, the total time can decrease, but this study is out of scope of the present work.

Performing a comparison against the time observed for GMM, it can be seen that the execution time is usually higher than any \( K \)-means tests executed (including the time needed for the proposed method). This can be explained considering that the EM algorithm used in GMM involves several operations that require a considerable amount of time.

### 5. Conclusions

Even though nowadays deep learning techniques and complex machine learning algorithms are being used with significant success, the unsupervised learning algorithms

<table>
<thead>
<tr>
<th>GMM</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td><img src="image1" alt="RGB Cluster 1" /></td>
<td><img src="image2" alt="RGB Cluster 2" /></td>
<td><img src="image3" alt="RGB Cluster 3" /></td>
<td><img src="image4" alt="RGB Cluster 4" /></td>
<td>Best BAcc.: 0.96009 Mean BAcc: 0.88859 Mean sen.: 0.80937 Mean spe.: 0.92209 Avg. time: 3.97900</td>
</tr>
<tr>
<td>nRGB</td>
<td><img src="image5" alt="nRGB Cluster 1" /></td>
<td><img src="image6" alt="nRGB Cluster 2" /></td>
<td><img src="image7" alt="nRGB Cluster 3" /></td>
<td><img src="image8" alt="nRGB Cluster 4" /></td>
<td>Best BAcc.: 0.94253 Mean BAcc: 0.75979 Mean sen.: 0.05137 Mean spe.: 0.98592 Avg. time: 5.96510</td>
</tr>
<tr>
<td>Lab</td>
<td><img src="image9" alt="Lab Cluster 1" /></td>
<td><img src="image10" alt="Lab Cluster 2" /></td>
<td><img src="image11" alt="Lab Cluster 3" /></td>
<td><img src="image12" alt="Lab Cluster 4" /></td>
<td>Best BAcc.: 0.96159 Mean BAcc: 0.89942 Mean sen.: 0.82844 Mean spe.: 0.92873 Avg. time: 3.47550</td>
</tr>
<tr>
<td>ab</td>
<td><img src="image13" alt="ab Cluster 1" /></td>
<td><img src="image14" alt="ab Cluster 2" /></td>
<td><img src="image15" alt="ab Cluster 3" /></td>
<td><img src="image16" alt="ab Cluster 4" /></td>
<td>Best BAcc.: 0.97867 Mean BAcc: 0.82215 Mean sen.: 0.85910 Mean spe.: 0.81874 Avg. time: 1.8572</td>
</tr>
<tr>
<td>YCrCb</td>
<td><img src="image17" alt="YCrCb Cluster 1" /></td>
<td><img src="image18" alt="YCrCb Cluster 2" /></td>
<td><img src="image19" alt="YCrCb Cluster 3" /></td>
<td><img src="image20" alt="YCrCb Cluster 4" /></td>
<td>Best BAcc.: 0.96228 Mean BAcc: 0.88119 Mean sen.: 0.79477 Mean spe.: 0.91842 Avg. time: 2.36320</td>
</tr>
<tr>
<td>CrCb</td>
<td><img src="image21" alt="CrCb Cluster 1" /></td>
<td><img src="image22" alt="CrCb Cluster 2" /></td>
<td><img src="image23" alt="CrCb Cluster 3" /></td>
<td><img src="image24" alt="CrCb Cluster 4" /></td>
<td>Best BAcc.: 0.97951 Mean BAcc: 0.81375 Mean sen.: 0.85651 Mean spe.: 0.80762 Avg. time: 1.94140</td>
</tr>
<tr>
<td>HSV</td>
<td><img src="image25" alt="HSV Cluster 1" /></td>
<td><img src="image26" alt="HSV Cluster 2" /></td>
<td><img src="image27" alt="HSV Cluster 3" /></td>
<td><img src="image28" alt="HSV Cluster 4" /></td>
<td>Best BAcc.: 0.94228 Mean BAcc: 0.78670 Mean sen.: 0.41777 Mean spe.: 0.86373 Avg. time: 2.6595</td>
</tr>
<tr>
<td>HS</td>
<td><img src="image29" alt="HS Cluster 1" /></td>
<td><img src="image30" alt="HS Cluster 2" /></td>
<td><img src="image31" alt="HS Cluster 3" /></td>
<td><img src="image32" alt="HS Cluster 4" /></td>
<td>Best BAcc.: 0.91606 Mean BAcc: 0.78668 Mean sen.: 0.41516 Mean spe.: 0.86913 Avg. time: 2.21520</td>
</tr>
<tr>
<td>H Orig</td>
<td><img src="image33" alt="H Orig Cluster 1" /></td>
<td><img src="image34" alt="H Orig Cluster 2" /></td>
<td><img src="image35" alt="H Orig Cluster 3" /></td>
<td><img src="image36" alt="H Orig Cluster 4" /></td>
<td>Best BAcc.: 0.94253 Mean BAcc: 0.78293 Mean sen.: 0.27400 Mean spe.: 0.93489 Avg. time: 2.0863</td>
</tr>
</tbody>
</table>
Table 9: Segmentation performance results for image "e" using \( K \)-means++.

<table>
<thead>
<tr>
<th>K-Means</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Metrics</th>
</tr>
</thead>
</table>
| Ground truth | ![Cluster 1](image1) | ![Cluster 2](image2) | ![Cluster 3](image3) | ![Cluster 4](image4) | Best BAcc: 0.99148  
Mean BAcc: 0.85533  
Mean sen.: 0.55745  
Mean spe.: 0.92899  
Avg. time: 1.89000 |
| RGB | ![Cluster 1](image5) | ![Cluster 2](image6) | ![Cluster 3](image7) | ![Cluster 4](image8) | Best BAcc: 0.99373  
Mean BAcc: 0.94526  
Mean sen.: 0.85293  
Mean spe.: 0.95088  
Avg. time: 0.93870 |
| nRGB | ![Cluster 1](image9) | ![Cluster 2](image10) | ![Cluster 3](image11) | ![Cluster 4](image12) | Best BAcc: 0.98950  
Mean BAcc: 0.85747  
Mean sen.: 0.61804  
Mean spe.: 0.92534  
Avg. time: 2.7201 |
| Lab | ![Cluster 1](image13) | ![Cluster 2](image14) | ![Cluster 3](image15) | ![Cluster 4](image16) | Best BAcc: 0.99226  
Mean BAcc: 0.9104  
Mean sen.: 0.78740  
Mean spe.: 0.9104  
Avg. time: 1.02559 |
| ab | ![Cluster 1](image17) | ![Cluster 2](image18) | ![Cluster 3](image19) | ![Cluster 4](image20) | Best BAcc: 0.99224  
Mean BAcc: 0.86661  
Mean sen.: 0.67054  
Mean spe.: 0.92761  
Avg. time: 1.31560 |
| YCrCb | ![Cluster 1](image21) | ![Cluster 2](image22) | ![Cluster 3](image23) | ![Cluster 4](image24) | Best BAcc: 0.99250  
Mean BAcc: 0.89663  
Mean sen.: 0.80508  
Mean spe.: 0.92476  
Avg. time: 1.0642 |
| CrCb | ![Cluster 1](image25) | ![Cluster 2](image26) | ![Cluster 3](image27) | ![Cluster 4](image28) | Best BAcc: 0.99258  
Mean BAcc: 0.88251  
Mean sen.: 0.64855  
Mean spe.: 0.92107  
Avg. time: 1.71349 |
| HSV | ![Cluster 1](image29) | ![Cluster 2](image30) | ![Cluster 3](image31) | ![Cluster 4](image32) | Best BAcc: 0.99258  
Mean BAcc: 0.92937  
Mean sen.: 0.80617  
Mean spe.: 0.93624  
Avg. time: 1.1482 |
| HS | ![Cluster 1](image33) | ![Cluster 2](image34) | ![Cluster 3](image35) | ![Cluster 4](image36) | Best BAcc: 0.99241  
Mean BAcc: 0.84233  
Mean sen.: 0.58515  
Mean spe.: 0.87480  
Avg. time: 0.58769 |
| H Orig | ![Cluster 1](image37) | ![Cluster 2](image38) | ![Cluster 3](image39) | ![Cluster 4](image40) | Best BAcc: 0.99601  
Mean BAcc: 0.95133  
Mean sen.: 0.78958  
Mean spe.: 0.94632  
Avg. time (C): 1.3908  
Avg. time (B): 2.28200 |
Table 10: Segmentation performance results for image "e" using GMM.

<table>
<thead>
<tr>
<th>GMM</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td>Best BAcc: 0.98543 Mean BAcc: 0.87761 Mean sen.: 0.56276 Mean spe.: 0.94142 Avg. time: 4.03769</td>
</tr>
<tr>
<td>nRGB</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td>Best BAcc: 0.99400 Mean BAcc: 0.95521 Mean sen.: 0.89530 Mean spe.: 0.96174 Avg. time: 3.09780</td>
</tr>
<tr>
<td>Lab</td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
<td>Best BAcc: 0.97908 Mean BAcc: 0.86823 Mean sen.: 0.65741 Mean spe.: 0.92706 Avg. time: 4.4489</td>
</tr>
<tr>
<td>ab</td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
<td>Best BAcc: 0.99123 Mean BAcc: 0.90806 Mean sen.: 0.79366 Mean spe.: 0.94410 Avg. time: 1.9385</td>
</tr>
<tr>
<td>YCrCb</td>
<td><img src="image17.png" alt="Image" /></td>
<td><img src="image18.png" alt="Image" /></td>
<td><img src="image19.png" alt="Image" /></td>
<td><img src="image20.png" alt="Image" /></td>
<td>Best BAcc: 0.98606 Mean BAcc: 0.86326 Mean sen.: 0.68289 Mean spe.: 0.92291 Avg. time: 2.966</td>
</tr>
<tr>
<td>CrCb</td>
<td><img src="image21.png" alt="Image" /></td>
<td><img src="image22.png" alt="Image" /></td>
<td><img src="image23.png" alt="Image" /></td>
<td><img src="image24.png" alt="Image" /></td>
<td>Best BAcc: 0.99088 Mean BAcc: 0.90607 Mean sen.: 0.84163 Mean spe.: 0.93257 Avg. time: 2.29639</td>
</tr>
<tr>
<td>HSV</td>
<td><img src="image25.png" alt="Image" /></td>
<td><img src="image26.png" alt="Image" /></td>
<td><img src="image27.png" alt="Image" /></td>
<td><img src="image28.png" alt="Image" /></td>
<td>Best BAcc: 0.99261 Mean BAcc: 0.89050 Mean sen.: 0.69485 Mean spe.: 0.93121 Avg. time: 2.5547</td>
</tr>
<tr>
<td>HS</td>
<td><img src="image29.png" alt="Image" /></td>
<td><img src="image30.png" alt="Image" /></td>
<td><img src="image31.png" alt="Image" /></td>
<td><img src="image32.png" alt="Image" /></td>
<td>Best BAcc: 0.99259 Mean BAcc: 0.93409 Mean sen.: 0.81547 Mean spe.: 0.94257 Avg. time: 2.22500</td>
</tr>
<tr>
<td>H Orig</td>
<td><img src="image33.png" alt="Image" /></td>
<td><img src="image34.png" alt="Image" /></td>
<td><img src="image35.png" alt="Image" /></td>
<td><img src="image36.png" alt="Image" /></td>
<td>Best BAcc: 0.99252 Mean BAcc: 0.85474 Mean sen.: 0.63664 Mean spe.: 0.86330 Avg. time: 2.3193</td>
</tr>
</tbody>
</table>

are still an attractive option. The advantage of unsupervised techniques like K-means resides in the fact that they require no training; the implementation is relatively simple and they do not require excessive computational resources.

The exposed paper presents a modified version of the HSV model, which is merely a different presentation of the $H$ component applying small changes. It is important to mention that these changes are meant to be used to help the $K$-means algorithm and not to be used as a new color model or for different purposes where the effect of the changes could lead to unexpected results. A proper study should be performed before using the changes in other cases or applications.
<table>
<thead>
<tr>
<th>K-Means</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Metrics</th>
</tr>
</thead>
</table>
| Ground truth | ![Cluster 1](image1) | ![Cluster 2](image2) | ![Cluster 3](image3) | ![Cluster 4](image4) | Best BAcc.: 0.99323  
Mean BAcc.: 0.94312  
Mean sen.: 0.81053  
Mean spe.: 0.94599  
Avg. time: 2.2399 |
| RGB     | ![Cluster 1](image5) | ![Cluster 2](image6) | ![Cluster 3](image7) | ![Cluster 4](image8) | Best BAcc.: 0.98536  
Mean BAcc.: 0.89529  
Mean sen.: 0.59818  
Mean spe.: 0.88083  
Avg. time: 1.34869 |
| nRGB    | ![Cluster 1](image9) | ![Cluster 2](image10) | ![Cluster 3](image11) | ![Cluster 4](image12) | Best BAcc.: 0.99398  
Mean BAcc.: 0.95669  
Mean sen.: 0.85852  
Mean spe.: 0.95288  
Avg. time: 1.33930 |
| Lab     | ![Cluster 1](image13) | ![Cluster 2](image14) | ![Cluster 3](image15) | ![Cluster 4](image16) | Best BAcc.: 0.99830  
Mean BAcc.: 0.94054  
Mean sen.: 0.78114  
Mean spe.: 0.92947  
Avg. time: 1.3225 |
| ab      | ![Cluster 1](image17) | ![Cluster 2](image18) | ![Cluster 3](image19) | ![Cluster 4](image20) | Best BAcc.: 0.99438  
Mean BAcc.: 0.94116  
Mean sen.: 0.80296  
Mean spe.: 0.94756  
Avg. time: 1.143 |
| YCrCb   | ![Cluster 1](image21) | ![Cluster 2](image22) | ![Cluster 3](image23) | ![Cluster 4](image24) | Best BAcc.: 0.99645  
Mean BAcc.: 0.91684  
Mean sen.: 0.70957  
Mean spe.: 0.90729  
Avg. time: 1.19459 |
| CrCb    | ![Cluster 1](image25) | ![Cluster 2](image26) | ![Cluster 3](image27) | ![Cluster 4](image28) | Best BAcc.: 0.96321  
Mean BAcc.: 0.90030  
Mean sen.: 0.67543  
Mean spe.: 0.94320  
Avg. time: 2.8174 |
| HSV     | ![Cluster 1](image29) | ![Cluster 2](image30) | ![Cluster 3](image31) | ![Cluster 4](image32) | Best BAcc.: 0.99215  
Mean BAcc.: 0.89292  
Mean sen.: 0.65391  
Mean spe.: 0.94095  
Avg. time: 1.1508 |
| HS      | ![Cluster 1](image33) | ![Cluster 2](image34) | ![Cluster 3](image35) | ![Cluster 4](image36) | Best BAcc.: 0.98262  
Mean BAcc.: 0.89173  
Mean sen.: 0.50620  
Mean spe.: 0.93380  
Avg. time: 0.68540 |
| H Orig  | ![Cluster 1](image37) | ![Cluster 2](image38) | ![Cluster 3](image39) | ![Cluster 4](image40) | Best BAcc.: 0.99826  
Mean BAcc.: 0.98679  
Mean sen.: 0.91741  
Mean spe.: 0.99612  
Avg. time (C): 1.84471  
Avg. time (B): 3.11008 |
Table 12: Segmentation performance results for image "f" using GMM.

<table>
<thead>
<tr>
<th>GMM</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc: 0.99634</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.93741</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen.: 0.80034</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe.: 0.95122</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 5.87</td>
</tr>
<tr>
<td>RGB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nRGB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc: 0.98807</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.86814</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen.: 0.49632</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe.: 0.84997</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 3.437</td>
</tr>
<tr>
<td>Lab</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ab</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc: 0.99766</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.95731</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen.: 0.86890</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe.: 0.95453</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 3.38859</td>
</tr>
<tr>
<td>YCrCb</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CrCb</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc: 0.99832</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.91859</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen.: 0.69350</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe.: 0.90564</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 2.8533</td>
</tr>
<tr>
<td>HSV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Best BAcc: 0.99738</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean BAcc: 0.94940</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean sen.: 0.84076</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean spe.: 0.95318</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg. time: 3.4557</td>
</tr>
<tr>
<td>H Orig</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 13: Final averages from all the tests for all the color spaces for K-means++.

<table>
<thead>
<tr>
<th>Color model</th>
<th>Mean BAcc</th>
<th>Best mean BAcc</th>
<th>Worst mean BAcc</th>
<th>Mean sen.</th>
<th>Mean spe.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>0.92801</td>
<td>0.85683</td>
<td>0.77453</td>
<td>0.71207</td>
<td>0.88839</td>
</tr>
<tr>
<td>nRGB</td>
<td>0.94225</td>
<td>0.87095</td>
<td>0.77021</td>
<td>0.64817</td>
<td>0.87594</td>
</tr>
<tr>
<td>Lab</td>
<td>0.93251</td>
<td>0.87306</td>
<td>0.78774</td>
<td>0.74322</td>
<td>0.90118</td>
</tr>
<tr>
<td>HSV</td>
<td>0.95070</td>
<td>0.87095</td>
<td>0.77657</td>
<td>0.75341</td>
<td>0.87851</td>
</tr>
<tr>
<td>ab</td>
<td>0.96864</td>
<td>0.90492</td>
<td>0.82715</td>
<td>0.8005</td>
<td>0.90480</td>
</tr>
<tr>
<td>CrCb</td>
<td>0.95407</td>
<td>0.87833</td>
<td>0.77657</td>
<td>0.75341</td>
<td>0.87851</td>
</tr>
<tr>
<td>YCrCb</td>
<td>0.9260</td>
<td>0.86462</td>
<td>0.76710</td>
<td>0.73808</td>
<td>0.89486</td>
</tr>
<tr>
<td>HS</td>
<td>0.94426</td>
<td>0.85733</td>
<td>0.76829</td>
<td>0.67054</td>
<td>0.91871</td>
</tr>
<tr>
<td>H Orig</td>
<td>0.93414</td>
<td>0.84049</td>
<td>0.70521</td>
<td>0.5838</td>
<td>0.89377</td>
</tr>
<tr>
<td>H Mod</td>
<td><strong>0.97581</strong></td>
<td><strong>0.93520</strong></td>
<td><strong>0.82823</strong></td>
<td><strong>0.83216</strong></td>
<td><strong>0.94343</strong></td>
</tr>
</tbody>
</table>
The changes in the $H$ component allow and force using bagging in the resulting $K$-means $+$ $+$ clusters. So, the usage of bagging procedure and the chromatic/achromatic separation in the $H$ component improve the outcome from the color segmentation.

### Conflicts of Interest

The authors declare that there are no conflicts of interest.

### References


---

**Table 14**: Final averages from all the tests for all the color spaces for GMM.

<table>
<thead>
<tr>
<th>Color model</th>
<th>Mean BAcc</th>
<th>Best mean BAcc</th>
<th>Worst mean BAcc</th>
<th>Mean sen.</th>
<th>Mean spe.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>0.88575</td>
<td>0.90196</td>
<td>0.85548</td>
<td>0.74395</td>
<td>0.92508</td>
</tr>
<tr>
<td>nRGB</td>
<td>0.87772</td>
<td>0.89974</td>
<td>0.95050</td>
<td>0.53360</td>
<td>0.9254</td>
</tr>
<tr>
<td>Lab</td>
<td>0.89226</td>
<td>0.90416</td>
<td>0.86582</td>
<td>0.74341</td>
<td>0.92888</td>
</tr>
<tr>
<td>ab</td>
<td>0.90574</td>
<td>0.93919</td>
<td>0.90347</td>
<td>0.77589</td>
<td>0.91127</td>
</tr>
<tr>
<td>YCrCb</td>
<td>0.88432</td>
<td>0.90150</td>
<td>0.86464</td>
<td>0.74329</td>
<td>0.92262</td>
</tr>
<tr>
<td>CrCb</td>
<td>0.89592</td>
<td>0.91717</td>
<td>0.85903</td>
<td>0.77130</td>
<td>0.90246</td>
</tr>
<tr>
<td>HSV</td>
<td>0.87142</td>
<td>0.89272</td>
<td>0.83729</td>
<td>0.65110</td>
<td>0.92527</td>
</tr>
<tr>
<td>HS</td>
<td>0.86954</td>
<td>0.87888</td>
<td>0.81174</td>
<td>0.62142</td>
<td>0.92431</td>
</tr>
<tr>
<td>$H$</td>
<td>0.86954</td>
<td>0.87888</td>
<td>0.81174</td>
<td>0.62142</td>
<td>0.62142</td>
</tr>
</tbody>
</table>

**Table 15**: Wilcoxon tests between the proposed improvement and the $K$-means and GMM algorithms using $ab$, CrCb, $H$, $HSV$, and $H$-$S$ $V$ color models.

<table>
<thead>
<tr>
<th></th>
<th>Versus $ab$</th>
<th>Versus CrCb</th>
<th>Versus $H$</th>
<th>Versus $HSV$</th>
<th>Versus $HSV$</th>
</tr>
</thead>
</table>

**Table 16**: Wilcoxon tests between the proposed improvement and the $K$-means and GMM algorithms using $Lab$, nRGB, RGB, and YCrCb color models.

<table>
<thead>
<tr>
<th></th>
<th>Versus $Lab$</th>
<th>Versus nRGB</th>
<th>Versus RGB</th>
<th>Versus YCrCb</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$-Means</td>
<td>$P$ val.: 1.2802e$-08$</td>
<td>$P$ val.: 5.5030e$-11$</td>
<td>$P$ val.: 1.6585e$-09$</td>
<td>$P$ val.: 1.0108e$-09$</td>
</tr>
<tr>
<td>GMM</td>
<td>$P$ val.: 4.2636e$-08$</td>
<td>$P$ val.: 8.9224e$-11$</td>
<td>$P$ val.: 2.8298e$-08$</td>
<td>$P$ val.: 1.9830e$-09$</td>
</tr>
</tbody>
</table>


[56] The Berkeley Segmentation Dataset and Benchmark BSDS500, https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bbsd/.


