

## 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](mailto:chavolla@gmail.com)

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

Academic Editor: Qin Yuming

Copyright © 2018 Edgar Chavolla et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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 learning 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 perceptual, 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.

$$V = \max(R, G, B) \quad (1)$$

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{1 - |\max(R, G, B) + \min(R, G, B) - 1|} \quad (2)$$

$$H = \begin{cases} 60 * \left( \frac{G - B}{\max(R, G, B) - \min(R, G, B)} \right) & R = \max(R, G, B) \\ 60 * \left( \frac{2 + (B - R)}{\max(R, G, B) - \min(R, G, B)} \right) & G = \max(R, G, B) \\ 60 * \left( \frac{4 + (R - G)}{\max(R, G, B) - \min(R, G, B)} \right) & B = \max(R, G, B) \end{cases} \quad (3)$$

**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.

$$\begin{aligned} r &= \frac{R}{R + G + B} \\ g &= \frac{G}{R + G + B} \\ b &= \frac{B}{R + G + B} \end{aligned} \quad (4)$$

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

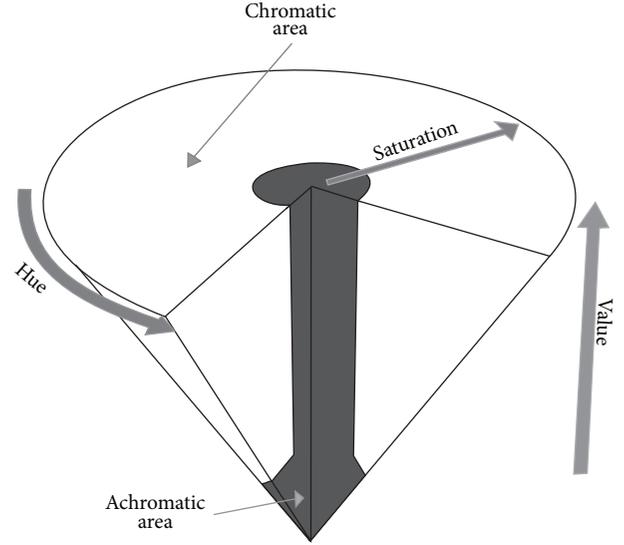


FIGURE 1: Achromatic and chromatic areas for  $HSV$  color space.

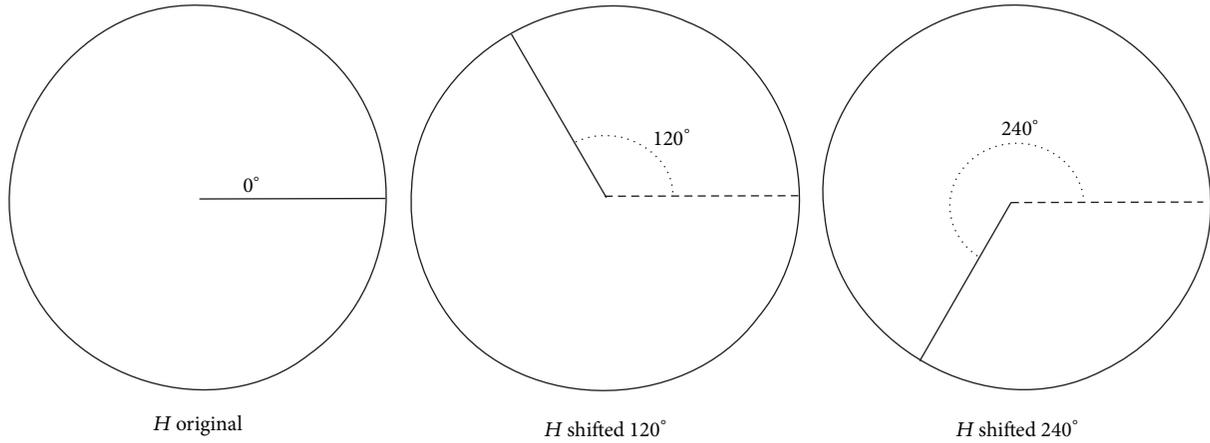
defined from 0 to 179 and the achromatic region from 200 to 255. Using the achromatic and chromatic definitions [55] adapted for  $HSV$  color space [Figure 1], the following can be stated:

- (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:

$$H = \begin{cases} (S \text{ and } V \text{ above } th) & \begin{cases} 60 * \left( \frac{G - B}{\max(R, G, B) - \min(R, G, B)} \right) & R = \max(R, G, B) \\ 60 * \left( \frac{2 + (B - R)}{\max(R, G, B) - \min(R, G, B)} \right) & G = \max(R, G, B) \\ 60 * \left( \frac{4 + (R - G)}{\max(R, G, B) - \min(R, G, B)} \right) & B = \max(R, G, B) \end{cases} \\ (S \text{ or } V \text{ below } th) & \left\{ 200 + \left( \left( \frac{V}{255} \right) * 55 \right) \right\} \end{cases} \quad (5)$$

FIGURE 2: Original  $H$  and the two shifted  $H$  representations.

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.

```

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

```

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

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

```

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

```

PSEUDOCODE 2: Cluster grouping.

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 cluster divided into two clusters. This issue does not exist in the clusters coming from the other two components.

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).

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

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

$$ACC = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (8)$$

$$Bacc = \alpha * SEN + \beta * SPC. \quad (9)$$

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 intended 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

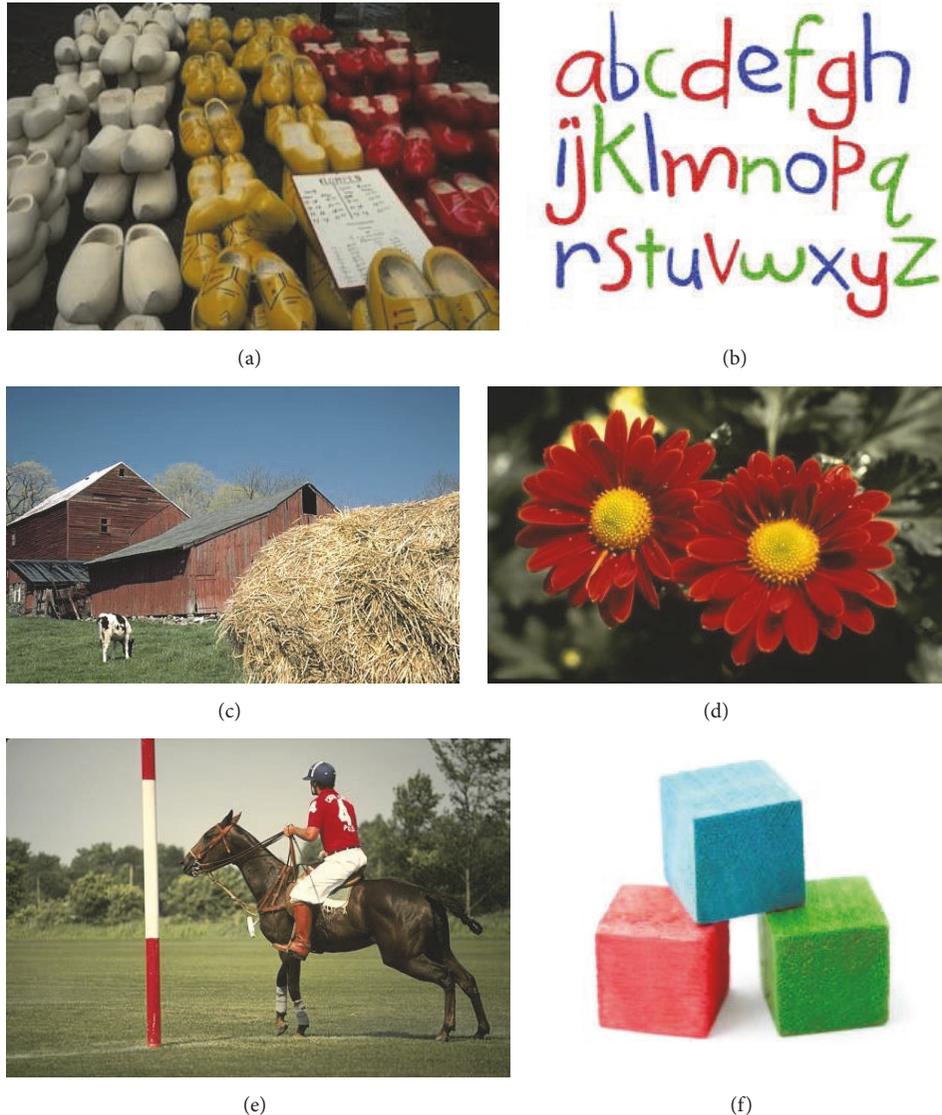


FIGURE 3: Original images for segmentation. Images (a), (c), (d), and (e) are from BSDS500. Images (b) and (f) are from Free Images website.

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++.

<i>K</i> -Means	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
Ground truth					
<i>RGB</i>					Best BAcc.: 0.85028 Mean BAcc.: 0.74197 Mean sen.: 0.62917 Mean spe.: 0.81776 Avg. time: 3.8598
<i>nRGB</i>					Best BAcc.: 0.93637 Mean BAcc.: 0.82969 Mean sen.: 0.53766 Mean spe.: 0.88299 Avg. time: 1.5083
<i>Lab</i>					Best BAcc.: 0.86040 Mean BAcc.: 0.7717 Mean sen.: 0.58555 Mean spe.: 0.85133 Avg. time: 3.5174
<i>ab</i>					Best BAcc.: 0.95228 Mean BAcc.: 0.84635 Mean sen.: 0.71681 Mean spe.: 0.90009 Avg. time: 1.9523
<i>YCrCb</i>					Best BAcc.: 0.79657 Mean BAcc.: 0.75988 Mean sen.: 0.64811 Mean spe.: 0.83186 Avg. time: 3.28
<i>CrCb</i>					Best BAcc.: 0.94294 Mean BAcc.: 0.81556 Mean sen.: 0.61425 Mean spe.: 0.87505 Avg. time: 1.5142
<i>HSV</i>					Best BAcc.: 0.91797 Mean BAcc.: 0.85888 Mean sen.: 0.72094 Mean spe.: 0.90462 Avg. time: 1.4304
<i>HS</i>					Best BAcc.: 0.91816 Mean BAcc.: 0.83095 Mean sen.: 0.50944 Mean spe.: 0.89749 Avg. time: 1.333
<i>H Orig</i>					Best BAcc.: 0.91808 Mean BAcc.: 0.76817 Mean sen.: 0.22684 Mean spe.: 0.96200 Avg. time: 0.7393
<i>H Mod</i>					Best BAcc.: <b>0.96551</b> Mean BAcc.: <b>0.87931</b> Mean sen.: 0.77377 Mean spe.: 0.90182 Avg. time (C): 1.4847 Avg. time (B): 2.3388

TABLE 2: Segmentation performance results for image “a” using GMM.

GMM	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
<i>RGB</i>					Best BAcc.: 0.87485 Mean BAcc.: 0.76002 Mean sen.: 0.65702 Mean spe.: 0.82868 Avg. time: 7.4548
<i>nRGB</i>					Best BAcc.: 0.93637 Mean BAcc.: 0.82969 Mean sen.: 0.53766 Mean spe.: 0.88299 Avg. time: 5.14159
<i>Lab</i>					Best BAcc.: 0.86040 Mean BAcc.: 0.7717 Mean sen.: 0.58555 Mean spe.: 0.85133 Avg. time: 4.5137
<i>ab</i>					Best BAcc.: 0.95228 Mean BAcc.: 0.84635 Mean sen.: 0.71681 Mean spe.: 0.90009 Avg. time: 3.1992
<i>YCrCb</i>					Best BAcc.: 0.79657 Mean BAcc.: 0.75988 Mean sen.: 0.64811 Mean spe.: 0.83186 Avg. time: 4.7854
<i>CrCb</i>					Best BAcc.: 0.94294 Mean BAcc.: 0.81556 Mean sen.: 0.61425 Mean spe.: 0.87505 Avg. time: 3.1574
<i>HSV</i>					Best BAcc.: 0.91797 Mean BAcc.: 0.85888 Mean sen.: 0.72094 Mean spe.: 0.90462 Avg. time: 4.2984
<i>HS</i>					Best BAcc.: 0.91816 Mean BAcc.: 0.83095 Mean sen.: 0.50944 Mean spe.: 0.89749 Avg. time: 3.4293
<i>H Orig</i>					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++.

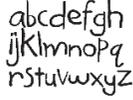
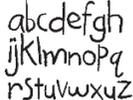
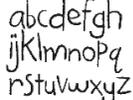
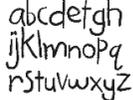
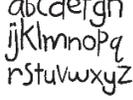
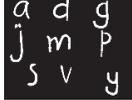
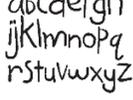
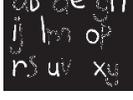
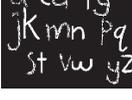
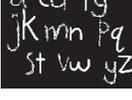
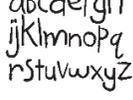
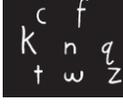
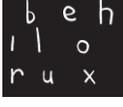
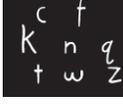
<i>K</i> -Means	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
Ground truth					
RGB					Best BAcc.: 0.99587 Mean BAcc.: 0.98778 Mean sen.: 0.93760 Mean spe.: 0.97967 Avg. time: 1.75579
nRGB					Best BAcc.: 0.99518 Mean BAcc.: 0.95903 Mean sen.: 0.61221 Mean spe.: 0.90010 Avg. time: 1.8469
Lab					Best BAcc.: 0.99575 Mean BAcc.: 0.99005 Mean sen.: 0.92404 Mean spe.: 0.98184 Avg. time: 1.113
ab					Best BAcc.: 0.99780 Mean BAcc.: 0.96863 Mean sen.: 0.72131 Mean spe.: 0.92257 Avg. time: 1.5184
YCrCb					Best BAcc.: 0.99567 Mean BAcc.: 0.98458 Mean sen.: 0.71123 Mean spe.: 0.97607 Avg. time: 1.84769
CrCb					Best BAcc.: 0.99688 Mean BAcc.: 0.96830 Mean sen.: 0.61425 Mean spe.: 0.92157 Avg. time: 1.286
HSV					Best BAcc.: 0.97252 Mean BAcc.: 0.90427 Mean sen.: 0.82520 Mean spe.: 0.94918 Avg. time: 1.41410
HS					Best BAcc.: 0.97236 Mean BAcc.: 0.90964 Mean sen.: 0.81605 Mean spe.: 0.95333 Avg. time: 1.1394
H Orig					Best BAcc.: 0.92121 Mean BAcc.: 0.84475 Mean sen.: 0.75755 Mean spe.: 0.86620 Avg. time: 1.3613
H Mod					Best BAcc.: <b>0.99589</b> Mean BAcc.: <b>0.99259</b> Mean sen.: 0.94847 Mean spe.: 0.98287 Avg. time (C): 1.7464 Avg. time (B): 3.166

TABLE 4: Segmentation performance results for image “b” using GMM.

GMM	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
RGB	abcdefgh ijklmnopq rstuvwxyZ				Best BAcc.: 0.99833 Mean BAcc.: 0.98109 Mean sen.: 0.96164 Mean spe.: 0.98576 Avg. time: 4.6552
nRGB	a cd fg jK mn Pq St vw yZ				Best BAcc.: 0.99846 Mean BAcc.: 0.95547 Mean sen.: 0.61632 Mean spe.: 0.90544 Avg. time: 6.5616
Lab	abcdefgh ijklmnopq rstuvwxyZ				Best BAcc.: 0.99733 Mean BAcc.: 0.98261 Mean sen.: 0.88530 Mean spe.: 0.98876 Avg. time: 3.3779
ab	abcdefgh ijklmnopq rstuvwxyZ				Best BAcc.: 0.99838 Mean BAcc.: 0.97665 Mean sen.: 0.84241 Mean spe.: 0.95545 Avg. time: 3.10470
YCrCb	abcdefgh ijklmnopq rstuvwxyZ				Best BAcc.: 0.99833 Mean BAcc.: 0.98288 Mean sen.: 0.91770 Mean spe.: 0.98878 Avg. time: 3.8567
CrCb	abcdefgh ijklmnopq rstuvwxyZ				Best BAcc.: 0.99829 Mean BAcc.: 0.96058 Mean sen.: 0.69106 Mean spe.: 0.91341 Avg. time: 4.0068
HSV	abcdefgh ijklmnopq rstuvwxyZ				Best BAcc.: 0.99818 Mean BAcc.: 0.91195 Mean sen.: 0.84421 Mean spe.: 0.96268 Avg. time: 3.47339
HS	abcdefgh ijklmnopq rstuvwxyZ				Best BAcc.: 0.97405 Mean BAcc.: 0.91007 Mean sen.: 0.83561 Mean spe.: 0.96637 Avg. time: 3.71440
H Orig					Best BAcc.: 0.92188 Mean BAcc.: 0.83645 Mean sen.: 0.75261 Mean spe.: 0.86185 Avg. time: 2.5643

worst 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++.

K-Means	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Metrics
Ground truth						
RGB						Best BAcc.: 0.95833 Mean BAcc.: 0.84450 Mean sen.: 0.58841 Mean spe.: 0.90370 Avg. time: 3.72890
nRGB						Best BAcc.: 0.97927 Mean BAcc.: 0.90491 Mean sen.: 0.64566 Mean spe.: 0.94043 Avg. time: 2.6793
Lab						Best BAcc.: 0.95947 Mean BAcc.: 0.84665 Mean sen.: 0.58651 Mean spe.: 0.90504 Avg. time: 2.82420
ab						Best BAcc.: 0.984993 Mean BAcc.: <b>0.96148</b> Mean sen.: 0.88592 Mean spe.: 0.97132 Avg. time: 1.0364
YCrCb						Best BAcc.: 0.96193 Mean BAcc.: 0.84934 Mean sen.: 0.64811 Mean spe.: 0.90670 Avg. time: 3.2729
CrCb						Best BAcc.: 0.98338 Mean BAcc.: 0.93794 Mean sen.: 0.86600 Mean spe.: 0.95268 Avg. time: 1.0914
HSV						Best BAcc.: 0.97474 Mean BAcc.: 0.88877 Mean sen.: 0.49019 Mean spe.: 0.95910 Avg. time: 3.01899
HS						Best BAcc.: 0.97121 Mean BAcc.: 0.86365 Mean sen.: 0.42246 Mean spe.: 0.94520 Avg. time: 3.0209
H Orig						Best BAcc.: 0.96433 Mean BAcc.: 0.92863 Mean sen.: 0.73995 Mean spe.: 0.94777 Avg. time: 0.66040
H Mod						Best BAcc.: <b>0.98993</b> Mean BAcc.: 0.94321 Mean sen.: 0.76518 Mean spe.: 0.96808 Avg. time (C): 1.9414 Avg. time (B): 2.5229

TABLE 6: Segmentation performance results for image “c” using GMM.

GMM	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Metrics
<i>RGB</i>						Best BAcc.: 0.97231 Mean BAcc: 0.86137 Mean sen.: 0.64875 Mean spe.: 0.91404 Avg. time: 7.4137
<i>nRGB</i>						Best BAcc.: 0.95474 Mean BAcc: 0.89090 Mean sen.: 0.50531 Mean spe.: 0.94607 Avg. time: 5.039
<i>Lab</i>						Best BAcc.: 0.97013 Mean BAcc: 0.86512 Mean sen.: 0.65040 Mean spe.: 0.91608 Avg. time: 4.7831
<i>ab</i>						Best BAcc.: 0.98512 Mean BAcc: 0.96448 Mean sen.: 0.90266 Mean spe.: 0.97358 Avg. time: 3.16470
<i>YCrCb</i>						Best BAcc.: 0.97427 Mean BAcc: 0.86622 Mean sen.: 0.66280 Mean spe.: 0.91694 Avg. time: 5.13540
<i>CrCb</i>						Best BAcc.: 0.98513 Mean BAcc: 0.94658 Mean sen.: 0.88602 Mean spe.: 0.95928 Avg. time: 1.9721
<i>HSV</i>						Best BAcc.: 0.98307 Mean BAcc: 0.88245 Mean sen.: 0.52690 Mean spe.: 0.94644 Avg. time: 5.60439
<i>HS</i>						Best BAcc.: 0.98099 Mean BAcc: 0.87426 Mean sen.: 0.55223 Mean spe.: 0.92978 Avg. time: 4.122
<i>H Orig</i>						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 7: Segmentation performance results for image “d” using *K*-means++.

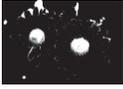
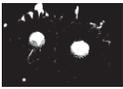
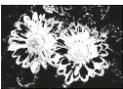
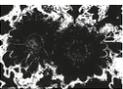
<i>K</i> -Means	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
Ground truth					
<i>RGB</i>					Best BAcc.: 0.95994 Mean BAcc.: 0.86935 Mean sen.: 0.75575 Mean spe.: 0.90860 Avg. time: 1.76310
<i>nRGB</i>					Best BAcc.: 0.90987 Mean BAcc.: 0.81386 Mean sen.: 0.54885 Mean spe.: 0.87593 Avg. time: 3.72300
<i>Lab</i>					Best BAcc.: 0.95932 Mean BAcc.: <b>0.88656</b> Mean sen.: 0.78536 Mean spe.: 0.91907 Avg. time: 1.2676
<i>ab</i>					Best BAcc.: 0.98018 Mean BAcc.: 0.81514 Mean sen.: 0.85177 Mean spe.: 0.80258 Avg. time: 0.8469
<i>YCrCb</i>					Best BAcc.: 0.96415 Mean BAcc.: 0.86543 Mean sen.: 0.74899 Mean spe.: 0.90738 Avg. time: 1.87110
<i>CrCb</i>					Best BAcc.: <b>0.98062</b> Mean BAcc.: 0.81118 Mean sen.: 0.84099 Mean spe.: 0.80067 Avg. time: 0.9742
<i>HSV</i>					Best BAcc.: 0.90939 Mean BAcc.: 0.78772 Mean sen.: 0.45141 Mean spe.: 0.85449 Avg. time: 1.25549
<i>HS</i>					Best BAcc.: 0.92218 Mean BAcc.: 0.78140 Mean sen.: 0.38454 Mean spe.: 0.87325 Avg. time: 1.144
<i>H Orig</i>					Best BAcc.: 0.91348 Mean BAcc.: 0.80163 Mean sen.: 0.48664 Mean spe.: 0.86596 Avg. time: 0.5639
<i>H Mod</i>					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

TABLE 8: Segmentation performance results for image “d” using GMM.

GMM	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
<i>RGB</i>					Best BAcc.: 0.96009 Mean BAcc.: 0.88859 Mean sen.: 0.80937 Mean spe.: 0.92209 Avg. time: 3.97900
<i>nRGB</i>					Best BAcc.: 0.94253 Mean BAcc.: 0.75979 Mean sen.: 0.05137 Mean spe.: 0.98592 Avg. time: 5.96510
<i>Lab</i>					Best BAcc.: 0.96159 Mean BAcc.: 0.89942 Mean sen.: 0.82844 Mean spe.: 0.92873 Avg. time: 3.47550
<i>ab</i>					Best BAcc.: 0.97867 Mean BAcc.: 0.82215 Mean sen.: 0.85910 Mean spe.: 0.81874 Avg. time: 1.8572
<i>YCrCb</i>					Best BAcc.: 0.96228 Mean BAcc.: 0.88119 Mean sen.: 0.79477 Mean spe.: 0.91842 Avg. time: 2.36320
<i>CrCb</i>					Best BAcc.: 0.97951 Mean BAcc.: 0.81375 Mean sen.: 0.85651 Mean spe.: 0.80762 Avg. time: 1.94140
<i>HSV</i>					Best BAcc.: 0.94228 Mean BAcc.: 0.78670 Mean sen.: 0.41777 Mean spe.: 0.86373 Avg. time: 2.6595
<i>HS</i>					Best BAcc.: 0.91606 Mean BAcc.: 0.78668 Mean sen.: 0.41516 Mean spe.: 0.86913 Avg. time: 2.21520
<i>H Orig</i>					Best BAcc.: 0.94253 Mean BAcc.: 0.78293 Mean sen.: 0.27400 Mean spe.: 0.93489 Avg. time: 2.0863

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 9: Segmentation performance results for image “e” using *K*-means++.

<i>K</i> -Means	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
Ground truth					
<i>RGB</i>					Best BAcc.: 0.99148 Mean BAcc.: 0.85533 Mean sen.: 0.55745 Mean spe.: 0.92899 Avg. time: 1.89000
<i>nRGB</i>					Best BAcc.: 0.99373 Mean BAcc.: 0.94526 Mean sen.: 0.85293 Mean spe.: 0.95088 Avg. time: 0.93870
<i>Lab</i>					Best BAcc.: 0.98950 Mean BAcc.: 0.85747 Mean sen.: 0.61804 Mean spe.: 0.92534 Avg. time: 2.7201
<i>ab</i>					Best BAcc.: 0.99226 Mean BAcc.: 0.9104 Mean sen.: 0.78740 Mean spe.: 0.91014 Avg. time: 1.02559
<i>YCrCb</i>					Best BAcc.: 0.99224 Mean BAcc.: 0.86661 Mean sen.: 0.67054 Mean spe.: 0.92761 Avg. time: 1.31560
<i>CrCb</i>					Best BAcc.: 0.99250 Mean BAcc.: 0.89663 Mean sen.: 0.80508 Mean spe.: 0.92476 Avg. time: 1.0642
<i>HSV</i>					Best BAcc.: 0.99258 Mean BAcc.: 0.88251 Mean sen.: 0.64855 Mean spe.: 0.92107 Avg. time: 1.71349
<i>HS</i>					Best BAcc.: 0.99258 Mean BAcc.: 0.92937 Mean sen.: 0.80617 Mean spe.: 0.93624 Avg. time: 1.1482
<i>H Orig</i>					Best BAcc.: 0.99241 Mean BAcc.: 0.84233 Mean sen.: 0.58515 Mean spe.: 0.87480 Avg. time: 0.58769
<i>H Mod</i>					Best BAcc.: <b>0.99601</b> Mean BAcc.: <b>0.95133</b> 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.

GMM	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
<i>RGB</i>					Best BAcc.: 0.98543 Mean BAcc.: 0.87761 Mean sen.: 0.56276 Mean spe.: 0.94142 Avg. time: 4.03769
<i>nRGB</i>					Best BAcc.: 0.99400 Mean BAcc.: 0.95521 Mean sen.: 0.89530 Mean spe.: 0.96174 Avg. time: 3.09780
<i>Lab</i>					Best BAcc.: 0.97908 Mean BAcc.: 0.86823 Mean sen.: 0.65741 Mean spe.: 0.92706 Avg. time: 4.4489
<i>ab</i>					Best BAcc.: 0.99123 Mean BAcc.: 0.90806 Mean sen.: 0.79366 Mean spe.: 0.94410 Avg. time: 1.9395
<i>YCrCb</i>					Best BAcc.: 0.98606 Mean BAcc.: 0.86326 Mean sen.: 0.68289 Mean spe.: 0.92291 Avg. time: 2.966
<i>CrCb</i>					Best BAcc.: 0.99088 Mean BAcc.: 0.90607 Mean sen.: 0.84163 Mean spe.: 0.93257 Avg. time: 2.29639
<i>HSV</i>					Best BAcc.: 0.99261 Mean BAcc.: 0.89050 Mean sen.: 0.69485 Mean spe.: 0.93121 Avg. time: 2.5547
<i>HS</i>					Best BAcc.: 0.99259 Mean BAcc.: 0.93409 Mean sen.: 0.81547 Mean spe.: 0.94257 Avg. time: 2.22500
<i>H Orig</i>					Best BAcc.: 0.99252 Mean BAcc.: 0.85474 Mean sen.: 0.63664 Mean spe.: 0.86330 Avg. time: 2.3193

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 II: Segmentation performance results for image “f” using *K*-means++.

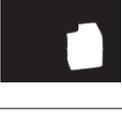
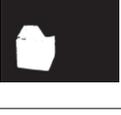
<i>K</i> -Means	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
Ground truth					
<i>RGB</i>					Best BAcc.: 0.99323 Mean BAcc: 0.94312 Mean sen.: 0.81053 Mean spe.: 0.94599 Avg. time: 2.2399
<i>nRGB</i>					Best BAcc.: 0.98536 Mean BAcc: 0.89529 Mean sen.: 0.59818 Mean spe.: 0.88083 Avg. time: 1.34869
<i>Lab</i>					Best BAcc.: 0.99398 Mean BAcc: 0.95669 Mean sen.: 0.85852 Mean spe.: 0.95288 Avg. time: 1.33930
<i>ab</i>					Best BAcc.: <b>0.99830</b> Mean BAcc: 0.94054 Mean sen.: 0.78114 Mean spe.: 0.92947 Avg. time: 1.3225
<i>YCrCb</i>					Best BAcc.: 0.99438 Mean BAcc: 0.94116 Mean sen.: 0.80296 Mean spe.: 0.94756 Avg. time: 1.143
<i>CrCb</i>					Best BAcc.: 0.99645 Mean BAcc: 0.91684 Mean sen.: 0.70957 Mean spe.: 0.90729 Avg. time: 1.19459
<i>HSV</i>					Best BAcc.: 0.96321 Mean BAcc: 0.90030 Mean sen.: 0.67543 Mean spe.: 0.94132 Avg. time: 2.8174
<i>HS</i>					Best BAcc.: 0.99215 Mean BAcc: 0.89926 Mean sen.: 0.65391 Mean spe.: 0.94095 Avg. time: 1.1508
<i>H Orig</i>					Best BAcc.: 0.98262 Mean BAcc: 0.89173 Mean sen.: 0.50620 Mean spe.: 0.93380 Avg. time: 0.68540
<i>H Mod</i>					Best BAcc.: 0.99826 Mean BAcc: <b>0.98679</b> 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.

GMM	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Metrics
<i>RGB</i>					Best BAcc.: 0.99634 Mean BAcc.: 0.93741 Mean sen.: 0.80034 Mean spe.: 0.95122 Avg. time: 5.87
<i>nRGB</i>					Best BAcc.: 0.98807 Mean BAcc.: 0.86814 Mean sen.: 0.49632 Mean spe.: 0.84997 Avg. time: 3.437
<i>Lab</i>					Best BAcc.: 0.99766 Mean BAcc.: 0.95731 Mean sen.: 0.86890 Mean spe.: 0.95453 Avg. time: 3.38859
<i>ab</i>					Best BAcc.: 0.99832 Mean BAcc.: 0.91859 Mean sen.: 0.69350 Mean spe.: 0.90564 Avg. time: 2.8533
<i>YCrCb</i>					Best BAcc.: 0.99738 Mean BAcc.: 0.94940 Mean sen.: 0.84076 Mean spe.: 0.95318 Avg. time: 3.4557
<i>CrCb</i>					Best BAcc.: 0.99872 Mean BAcc.: 0.92172 Mean sen.: 0.76444 Mean spe.: 0.91761 Avg. time: 2.2925
<i>HSV</i>					Best BAcc.: 0.93728 Mean BAcc.: 0.88837 Mean sen.: 0.68992 Mean spe.: 0.93274 Avg. time: 4.0777
<i>HS</i>					Best BAcc.: 0.97601 Mean BAcc.: 0.89289 Mean sen.: 0.59888 Mean spe.: 0.94690 Avg. time: 2.65219
<i>H Orig</i>					Best BAcc.: 0.98304 Mean BAcc.: 0.91677 Mean sen.: 0.75014 Mean spe.: 0.927527 Avg. time: 1.69729

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

Color model	Mean BAcc	Best mean BAcc	Worst mean BAcc	Mean sen.	Mean spe.
<i>RGB</i>	0.92801	0.85683	0.77453	0.71207	0.88839
<i>nRGB</i>	0.94225	0.87095	0.77021	0.64817	0.87594
<i>Lab</i>	0.93251	0.87306	0.78774	0.74322	0.90118
<i>ab</i>	0.96864	0.90492	0.82715	0.8005	0.90480
<i>YCrCb</i>	0.9260	0.86462	0.76710	0.73808	0.89486
<i>CrCb</i>	0.95407	0.87833	0.77657	0.75341	0.87851
<i>HSV</i>	0.95070	0.87095	0.76136	0.67054	0.91871
<i>HS</i>	0.94426	0.85733	0.76829	0.64076	0.90818
<i>H Orig</i>	0.93414	0.84049	0.70521	0.5838	0.89377
<i>H Mod</i>	<b>0.97581</b>	<b>0.93520</b>	<b>0.82823</b>	<b>0.83216</b>	<b>0.94343</b>

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

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

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

	Versus <i>ab</i>	Versus CrCb	Versus <i>H</i>	Versus <i>HS</i>	Versus <i>HSV</i>
<i>K</i> -Means	<i>P</i> val.: 2.2507e – 07	<i>P</i> val.: 7.3529e – 10	<i>P</i> val.: 1.1304e – 11	<i>P</i> val.: 8.7888e – 12	<i>P</i> val.: 1.3146e – 11
GMM	<i>P</i> val.: 7.7979e – 07	<i>P</i> val.: 7.3529e – 10	<i>P</i> val.: 1.1887e – 11	<i>P</i> val.: 1.0752e – 11	<i>P</i> val.: 2.5107e – 11

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

	Versus <i>Lab</i>	Versus <i>nRGB</i>	Versus <i>RGB</i>	Versus YCrCb
<i>K</i> -Means	<i>P</i> val.: 1.2802e – 08	<i>P</i> val.: 5.5030e – 11	<i>P</i> val.: 1.6585e – 09	<i>P</i> val.: 1.0108e – 09
GMM	<i>P</i> val.: 4.2636e – 08	<i>P</i> val.: 8.9224e – 11	<i>P</i> val.: 2.8298e – 08	<i>P</i> val.: 1.9830e – 09

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

- [1] D. Bone, C.-C. Lee, T. Chaspari, J. Gibson, and S. Narayanan, "Signal processing and machine learning for mental health research and clinical applications [Perspectives]," *IEEE Signal Processing Magazine*, vol. 34, no. 5, pp. 195-196, 2017.
- [2] Q. Zhang, X. Zeng, W. Hu, and D. Zhou, "A machine learning-empowered system for long-term motion-tolerant wearable monitoring of blood pressure and heart rate with Ear-ECG/PPG," *IEEE Access*, vol. 5, pp. 10547-10561, 2017.
- [3] J. Zhang, R. L. Lafta, X. Tao et al., "Coupling a fast fourier transformation with a machine learning ensemble model to support recommendations for heart disease patients in a tele-health environment," *IEEE Access*, vol. 5, pp. 10674-10685, 2017.
- [4] J. Wu, Y. Xiao, C. Xia et al., "Identification of biomarkers for predicting lymph node metastasis of stomach cancer using clinical DNA methylation data," *Disease Markers*, vol. 2017, Article ID 5745724, 7 pages, 2017.
- [5] J. K. Kim and S. Kang, "Neural network-based coronary heart disease risk prediction using feature correlation analysis," *Journal of Healthcare Engineering*, vol. 2017, Article ID 2780501, pp. 1-13, 2017.
- [6] J. Zhang, J. Xiao, J. Wan et al., "A parallel strategy for convolutional neural network based on heterogeneous cluster for mobile information system," *Mobile Information Systems*, vol. 2017, Article ID 3824765, 12 pages, 2017.
- [7] H. Chen, B. Jiang, and N. Lu, "Data-driven incipient sensor fault estimation with application in inverter of high-speed railway," *Mathematical Problems in Engineering*, vol. 2017, Article ID 8937356, 13 pages, 2017.
- [8] Y. Liu, Y. Liu, J. Liu et al., "A MapReduce based high performance neural network in enabling fast stability assessment of power systems," *Mathematical Problems in Engineering*, vol. 2017, Article ID 4030146, 12 pages, 2017.
- [9] M. Shafiq, X. Yu, A. A. Laghari, and D. Wang, "Effective feature selection for 5G IM applications traffic classification," *Mobile Information Systems*, vol. 2017, Article ID 6805056, pp. 1-12, 2017.
- [10] R. Eskandarpour and A. Khodaei, "Machine learning based power grid outage prediction in response to extreme events," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3315-3316, 2017.
- [11] Y. Yang, M. Yang, S. Huang, Y. Que, M. Ding, and J. Sun, "Multifocus image fusion based on extreme learning machine and human visual system," *IEEE Access*, vol. 5, pp. 6989-7000, 2017.
- [12] J. Kremer, K. Stensbo-Smidt, F. Gieseke, K. S. Pedersen, and C. Igel, "Big universe, big data: machine learning and image analysis for astronomy," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 16-22, 2017.
- [13] R. M. Mehmood, R. Du, and H. J. Lee, "Optimal feature selection and deep learning ensembles method for emotion

- recognition from human brain EEG sensors," *IEEE Access*, vol. 5, pp. 14797–14806, 2017.
- [14] Y. Xia, Z. Ji, A. Krylov, H. Chang, and W. Cai, "Machine learning in multimodal medical imaging," *BioMed Research International*, vol. 2017, Article ID 1278329, 2 pages, 2017.
- [15] H. Liu, C. Zhang, and D. Huang, "Extreme learning machine and moving least square regression based solar panel vision inspection," *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 7406568, 10 pages, 2017.
- [16] G. Wen, H. Li, J. Huang, D. Li, and E. Xun, "Random deep belief networks for recognizing emotions from speech signals," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 1945630, 9 pages, 2017.
- [17] R. Narayan, V. P. Singh, and S. Chakraverty, "Quantum neural network based machine translator for hindi to english," *The Scientific World Journal*, vol. 2014, Article ID 485737, 8 pages, 2014.
- [18] S. Rosenblum and G. Dror, "Identifying developmental dysgraphia characteristics utilizing handwriting classification methods," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 2, pp. 293–298, 2017.
- [19] L. Likforman-Sulem, A. Esposito, M. Faundez-Zanuy, S. Clemencon, and G. Cordasco, "EMOTHAW: A novel database for emotional state recognition from handwriting and drawing," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 2, pp. 273–284, 2017.
- [20] B. Zhou, "Statistical machine translation for speech: A perspective on structures, learning, and decoding," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1180–1202, 2013.
- [21] "The Berkeley Segmentation Dataset and Benchmark, BSDS500," <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>.
- [22] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.
- [23] A. Sironi, E. Turetken, V. Lepetit, and P. Fua, "Multiscale centerline detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1327–1341, 2016.
- [24] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 8, pp. 1558–1570, 2015.
- [25] S. Zhu, D. Cao, S. Jiang, Y. Wu, and P. Hu, "Fast superpixel segmentation by iterative edge refinement," *IEEE Electronics Letters*, vol. 51, no. 3, pp. 230–232, 2015.
- [26] J. Sigut, F. Fumero, O. Nuñez, and M. Sigut, "Automatic marker generation for watershed segmentation of natural images," *IEEE Electronics Letters*, vol. 50, no. 18, pp. 1281–1283, 2014.
- [27] J. Pont-Tuset, P. Arbeláez, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 128–140, 2017.
- [28] L. Velho, A. Frery, and J. Gomes, *Image Processing for computer graphics and vision*, Springer, Second edition, 2009.
- [29] A. Hanbury and J. Serra, "A 3D-polar coordinate colour representation suitable for image analysis. pattern recognition and image processing Group Technical Report 77," Tech. Rep., Vienna University of Technology, Vienna, Austria, 2003.
- [30] CIE, *Commission internationale de l'Eclairage proceedings*, Cambridge University Press, Cambridge, UK, 1931.
- [31] G. Finlayson and R. Xu, "Illuminant and gamma comprehensive normalisation in log RGB space," *Pattern Recognition Letters*, vol. 24, no. 11, pp. 1679–1690, 2003.
- [32] T. Kuremoto, Y. Kinoshita, L.-B. Feng, S. Watanabe, K. Kobayashi, and M. Obayashi, "A gesture recognition system with retina-V1 model and one-pass dynamic programming," *Neurocomputing*, vol. 116, pp. 291–300, 2013.
- [33] E. Blanco, M. Mazo, L. M. Bergasa, S. Palazuelos, and M. Marrón, "A method to increase class separation in the HS plane for color segmentation applications," in *Proceedings of the IEEE International Symposium on Intelligent Signal Processing (WISP '07)*, Spain, 2007.
- [34] H. Yang, X. Wang, Q. Wang, and X. Zhang, "LS-SVM based image segmentation using color and texture information," *Journal of Visual Communication and Image Representation*, vol. 23, no. 7, pp. 1095–1112, 2012.
- [35] S. Phradornpanitchakul, A. Duangchit, and R. Chaisrichaoren, "Enhanced danger detection of headlight through vision estimation and vector magnitude," in *Proceedings of the 4th Joint International Conference on Information and Communication Technology, Electronic and Electrical Engineering (JICTEE '14)*, Thailand, March 2014.
- [36] W.-M. Liu, L.-H. Wang, and Z.-F. Yang, "Application of self adapts to RGB threshold value for robot soccer," in *Proceedings of the 2010 International Conference on Machine Learning and Cybernetics (ICMLC '10)*, pp. 704–707, China, July 2010.
- [37] S. M. Khaled, M. S. Islam, M. G. Rabbani et al., "Combinatorial color space models for skin detection in sub-continental human images," in *Proceedings of the Visual Informatics, First International Visual Informatics Conference (IVIC '09)*, pp. 532–542, 2009.
- [38] A. Vadivel, S. Suralb, and A. Majumdar, "An integrated color and intensity co-occurrence matrix," *Pattern Recognition Letters*, vol. 28, no. 8, pp. 974–983, 2007.
- [39] R. Mente, B. V. Dhandra, and G. Mukarambi, "Color image segmentation and recognition based on shape and color features," *International Journal of Computer Science Engineering (IJCSSE)*, vol. 3, no. 1, pp. 2319–7323, 2014.
- [40] K. Murphy, *Machine Learning A probabilistic Perspective*, MIT Press, Cambridge Massachusetts, 2012.
- [41] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.
- [42] S. P. Lloyd, "Least squares quantization in PCM," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [43] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, 2007.
- [44] S. Arumugadevi and V. Seenivasagam, "Color image segmentation using feedforward neural networks with FCM," *International Journal of Automation and Computing*, vol. 13, no. 5, pp. 491–500, 2016.
- [45] C. Pan, D. S. Park, Y. Yang, and H. M. Yoo, "Leukocyte image segmentation by visual attention and extreme learning machine," *Neural Computing and Applications*, vol. 21, no. 6, pp. 1217–1227, 2012.
- [46] S. W. Oh and S. J. Kim, "Approaching the computational color constancy as a classification problem through deep learning," *Pattern Recognition*, vol. 61, pp. 405–416, 2017.

- [47] Q. Sang, Z. Lin, and S. T. Acton, "Learning automata for image segmentation," *Pattern Recognition Letters*, vol. 74, pp. 46–52, 2016.
- [48] M. Sridharan and P. Stone, "Structure-based color learning on a mobile robot under changing illumination," *Autonomous Robots*, vol. 23, no. 3, pp. 161–182, 2007.
- [49] K. Kim, C. Oh, and K. Sohn, "Non-parametric human segmentation using support vector machine," *IEEE Transactions on Consumer Electronics*, vol. 62, no. 2, pp. 150–158, 2016.
- [50] A. Lucchi, P. Marquez-Neila, C. Becker et al., "Learning structured models for segmentation of 2-D and 3-D imagery," *IEEE Transactions on Medical Imaging*, vol. 34, no. 5, pp. 1096–1110, 2015.
- [51] M. Gong, Y. Qian, and L. Cheng, "Integrated foreground segmentation and boundary matting for live videos," *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1356–1370, 2015.
- [52] A. Pratondo, C. Chui, and S. Ong, "Integrating machine learning with region-based active contour models in medical image segmentation," *Journal of Visual Communication and Image Representation*, 2016.
- [53] X.-Y. Wang, Q.-Y. Wang, H.-Y. Yang, and J. Bu, "Color image segmentation using automatic pixel classification with support vector machine," *Neurocomputing*, vol. 74, no. 18, pp. 3898–3911, 2011.
- [54] H. G. Li, G. Q. Wu, X. G. Hu, J. Zhang, L. Li, and X. Wu, "K-Means Clustering with Bagging and MapReduce," in *Proceedings of the 44th Hawaii International Conference on System Sciences*, pp. 1–8, 2011.
- [55] D.-C. Tseng and C.-H. Chang, "Color segmentation using perceptual attributes," in *Proceedings of the 11th IAPR International Conference on Pattern Recognition (IAPR '92)*, pp. 228–231, Netherlands, September 1992.
- [56] A. Azad, S. Pyne, and A. Pothen, "Matching phosphorylation response patterns of antigen-receptor-stimulated T cells via flow cytometry," *BMC Bioinformatics*, vol. 13, p. S10, 2012.
- [57] A. Azad, B. Rajwa, and A. Pothen, "Immunophenotype discovery, hierarchical organization, and template-based classification of flow cytometry samples," *Frontiers in Oncology*, 2016.
- [58] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [59] FreeImages, A web based free photography stock site, <http://www.freeimages.co.uk>.
- [60] K. Brodersen, C. S. Ong, K. Stephan, and J. Buhmann, "The balanced accuracy and its posterior distribution," in *Proceedings of the In 20th international conference on pattern recognition (ICPR)*, pp. 3121–3124, 2010.
- [61] L. C. Neto, G. Ramalho, J. F. S. Neto, R. Veras, and F. N. Medeiros, "An unsupervised coarse-to-fine algorithm for blood vessel segmentation in fundus images," *Expert Systems with Applications*, vol. 78, 2017.
- [62] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 80–83, 1945.
- [63] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

