06: Segmentation Techniques

Sergio Steven Leal Cuellar* and Mateo Rueda Molano†,
Department of Biomedical Engineering, University of the Andes, Bogota, Colombia
Email: *ss.leal10@uniandes.edu.co, †ms.rueda10@uniandes.edu.co,

***Abstract*—In this report the process of segmentation over some images from a partition of the BS data set will be describe. In order to develop quality code that enables to make a segmentation on several color spaces and using different clustering methods a function on python3 was done. The function developed was done having in mind a segmentation in the RGB, HSV and Lab color spaces with clustering methods like K-means, Gaussian mixture, Hierarchical clustering and watersheds. The results obtained varied depending on the images used. In general, the performance of the algorithm is quite low due to the noise that causes over-segmentation. Finally, the function proposed presents some limitations that can be overcame applying some pre-processing to the images e.g apply a Gaussian filter in order to diminish the number of objects in the image.**

I. Introduction

Image segmentation is widely used in medical image processing, face recognition, pedestrian detection, iris recognition, video suirvellance, etc. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. In practice, it is often not interested in all parts of the image, but only for some certain areas which have the same characteristics [3]. Image segmentation is based on certain criteria to divide an input image into a number of the same nature of the category in order to extract the area which people are interested in. The current image segmentation techniques include region-based segmentation, edge detection segmentation, segmentation based on clustering, segmentation based on weakly-supervised learning in CNN and more [4].

Segmentation based on clustering is an important tool for many researches. There are many different methods and one of the most popular and used is k-means clustering algorithm. K-means stores k centroids that it uses to define clusters. A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid. There are given a training set x(1),...,x(m), and the desire is to group the data into a few cohesive "clusters." Here, we are given feature vectors for each data point x(i) Rn as usual; but no labels y(i) (making this an unsupervised learning problem). The goal is to predict k centroids and a label c(i) for each data point. The k-means clustering minimizes euclidean distance between each datapoint (training data) and each centroid in order to group data in clusters [5]. Each cluster center is replaced by the coordinate-wise average of all data points that are closest to it, the algorithm finishes when it converges to a local minimum of the within-cluster sum of squares [5].

Another important method of clustering is the Gaussian mixture which involves the usage of a Mahalanobis distance to relate the samples with their cluster centers instead of using their nearest neighbors. In addition, a group of samples are trying to be described with a Gaussian distribution and so the whole data can be described with a mixture of Gaussian distributions. As a consequence, a soft assignment is done for every cluster and responsibilities are calculated for each group. Finally, the parameters of the Gaussian distribution are being re calculated taking into account the responsibilities previously assigned for each group.[1]

Watersheds is a segmentation algorithm inspired on the flooding of a topological surface starting from some points in the space. Those starting points are local minimums in the gray scale image (pixels with lower values than their nearest pixels). The labels for different regions can be established flooding regions which contain a different local minimum are near to meet. Nevertheless, due to the fact that regions are being related with a single local minimum, realizing a over-segmentation is a problem. In order to avoid this, an alternative is imposing a arbitrary number of local minimums on the image.[1]

Finally, hierarchical clustering implies the generation of very small clusters for similar data at first and then generate bigger clusters conformed by the association of smaller clusters. With that sense, a set of clusters of different sizes made up from other clusters is obtained.

II. Materials and Methods

A. Database description

In order to develop the function to make a segmentation in various color spaces and using different clustering methods a small set of images was taken from the whole Berkeley Segmentation Dataset. [2] In the train set, there was 60 different images in RGB with a size of 481*321 ,each image with a set of annotations done by different humans. There are two types of annotations, annotations for the regions and annotations for the boundaries. The test set is made up of 28 images each of them with their different annotations for the regions and for the boundaries for each human.

B. Method description

The overall method used in the function to make the segmentation was; First, to establish the color space desired and get the 3 characteristic matrix for each color space. If the desired color space was Lab o HSV, the values for the matrix were normalized in a scale from 0 to 255 to make them comparable to RGB because the values of this color spaces were small and with negative values in some cases. Then, the matrix of the color space was reshaped to a size of (m*n,3). Nevertheless, if the desired color space was +xy, two columns of the spacial location were added so that the final matrix had a shape of (m*n,5). Next, the desired type of clustering method (kmeans, gmm, hierarchical or watersheds) and the number of clusters was set and the model was fitted. Finally, the labels were obtained and plotted in order to evaluate the method. A particular processing was done to the images which were pretended to be segmented with hierarchical clustering. This was to re scale the images to a fourth of their original size due to problems in the memory used to generate the different hierarchical clusters and the distance between them which implies a lot of computational memory. The parameters 'numberOfClusters' specifies the number of clusters, and so the number of regions obtained in the final segmentation that will be used to describe the samples. This parameter is set by the user and the number of clusters determines how well the algorithm will segment. For example, if the samples are naturally grouped in 4 clusters ,because of its intrinsic characteristics, a clustering method using 3 clusters will not describe the data in a proper way because 2 of the original clusters will be explained by just 1 cluster. In the case of the watershed method, the number of clusters is related to number of local minimums desired for a image.

The evaluation method that was used to measure the performance of the algorithm was; to apply a sobel filter to the segmentation obtained and to an arbitrary annotation for the boundaries in order to get their borders. Then, the Jaccard Index was applied between the borders of the segmentation and the borders of the annotation.

III. Results

Below is an example of the segmentation performed by our algorithm and the edges of the segmentation, basis for the development of our metric.

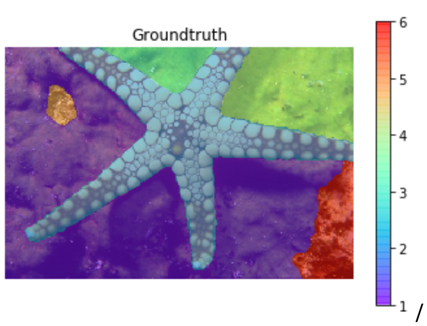

Figure 1. Segmentation annotation of the database for the image 12003

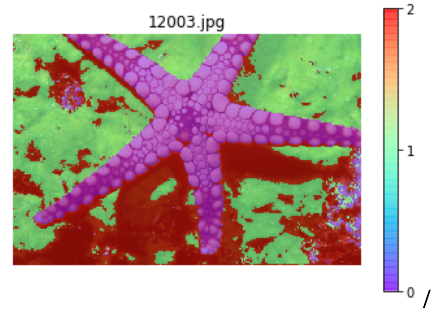

Figure 2. Segmentation result of our algorithm for the image 12003. GMM method was used, lab color space and k=4

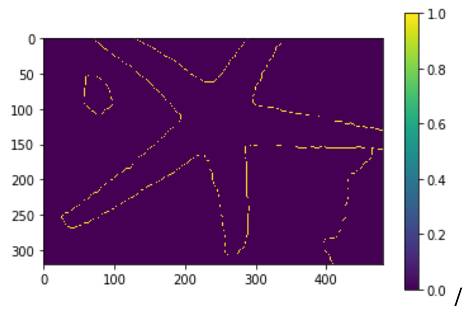

Figure 3. Segmentation result of our algorithm for the image 12003. GMM method was used, lab color space and k=4

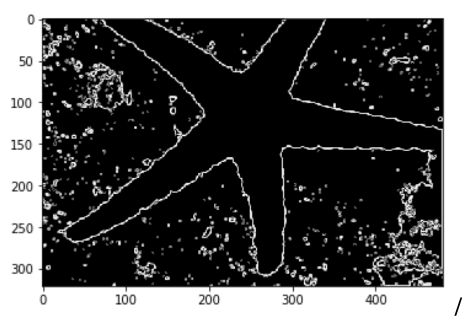

Figure 4. Borders in our segmentation algorithm. GMM method was used, lab color space and k=4

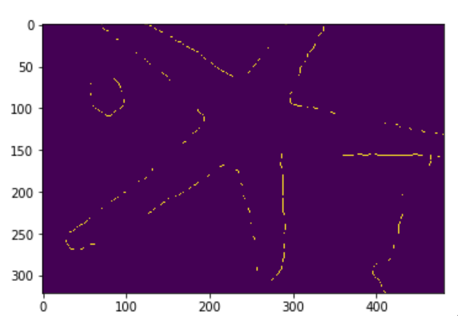

Figure 5. Borders in our segmentation algorithm. GMM method was used, lab color space and k=4

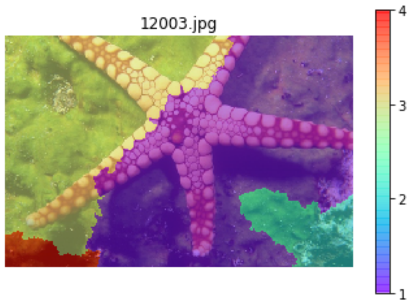

Figure 6. Segmentation result by watershed method. Lab color space and 4 minima

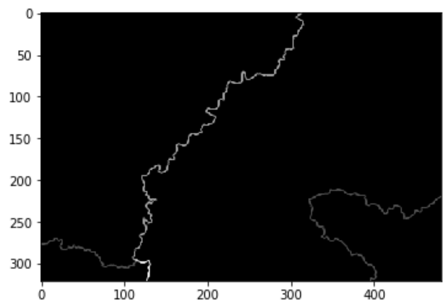

Figure 7. Border segmentation result by watershed method. Lab color space and 4 minima

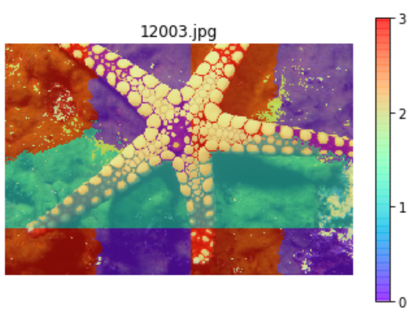

Figure 8. Segmentation result by gmm method. RGB+xy color space and 4 minima

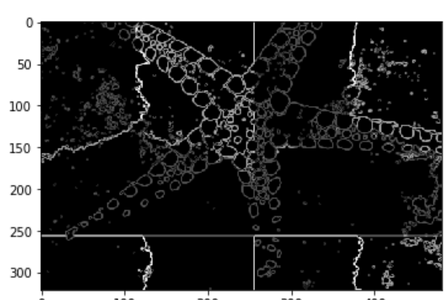

Figure 9. Border segmentation result by gmm method. RGB+xy color space and 4 minima

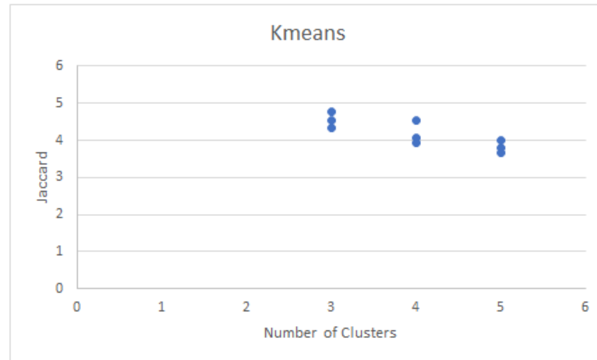

Figure 10. Border segmentation result by gmm method. RGB+xy color space and 4 minima

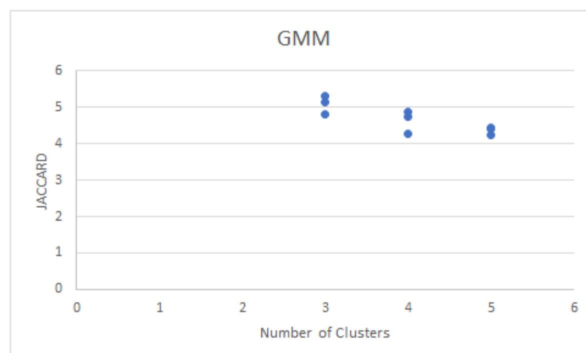

Figure 11. Border segmentation result by gmm method. RGB+xy color space and 4 minima

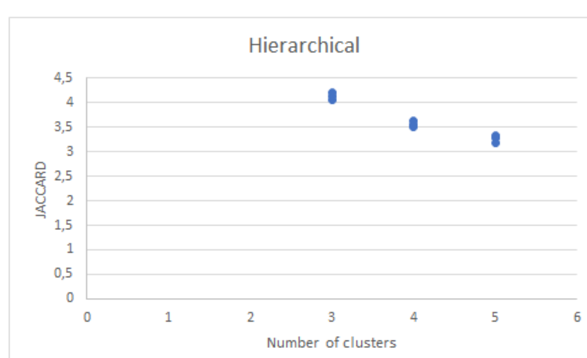

Figure 12. Border segmentation result by gmm method. RGB+xy color space and 4 minima

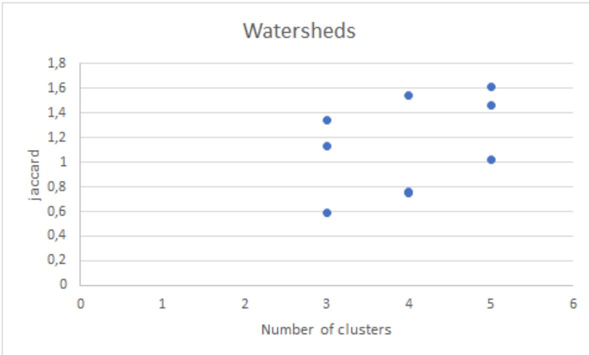

Figure 13. Border segmentation result by gmm method. RGB+xy color space and 4 minima

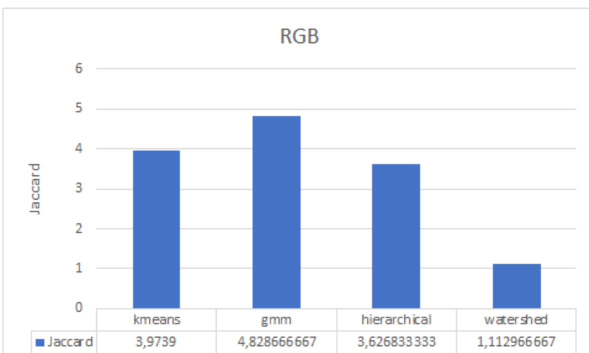

Figure 14. Jaccard vs. Number of clusters. The graph only shows RGB color space

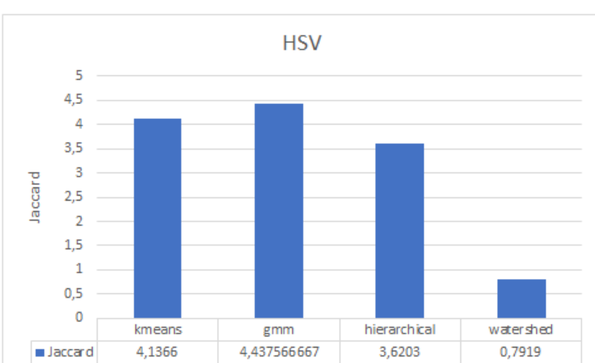

Figure 15. Jaccard vs. Number of clusters. The graph only shows HSV color space

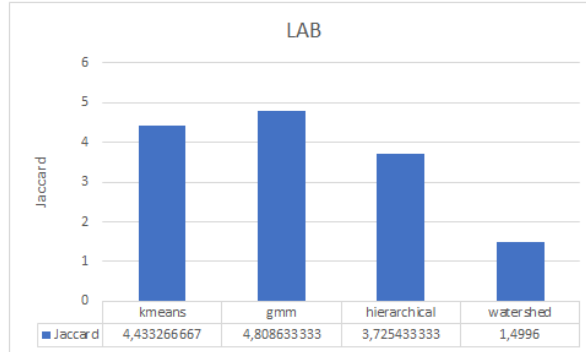

Figure 16. Jaccard vs. Number of clusters. The graph only shows LAB color space

IV. Discussion

Due to the existence of different annotations from every different human, a evaluation method that considers common objects across all the annotations can be done. Then, compare the frequent objects in annotations with the segmentation generated. Another alternative is to mean all the annotations and as a result obtain just one single annotations.

The methodology was evaluated comparing between edges on the segmentation and annotation because it has a low sensitivity to the variations in the number of clusters used. Also, because important regions with strong edges will not differ by a lot across segments with different clusters. The main shortcoming for this strategy is that the appearance of small artifacts (usually because of details in the object itself or changes in the illumination) in the image will cause the sobel filter to identify borders in homogeneous regions or inside the object. Because of this, the segmentation will have far more regions than the annotation, this is a over-segmentation which will diminish the Jaccard index obtained because of the size of the union of both segmentation and annotation. This issue can be seen in figure 4 which is the result of our border segmentation in comparison to figure 3. There a lot of artifacts which are not the border of the star but small changes in textures and colors and this is the reason why our evaluation methodology (Jaccard index with respect to edges) was so low. The intersection of the image as we see in figure 5 was satisfactory however the union considers all the artifacts that are not in the image annotation.

Several color spaces are used because using just RGB color space, makes the segmentation just between objects that differ in their colors. Adding another color space like HSV, enables to distinguish objects which differ in the saturation (across materials) and taking into account the gray scale information. Also, in Lab space is also considered the gray scale information at the same time that color information in considered. In addition, because in some spaces like RGB or HSV, the distance between the colors is not measurable through euclidean distance. At the same time, is reasonable to use an xy information vectors in order to relate the clusters also by their spacial location, nearby pixels in the matrix.

The color spaces that gave the overall best results were Lab and then HSV. This can be seen in figures 14 15 16 which show what Lab color space has the highest Jaccard index overall. The previous situation occurs because these color spaces give more discriminate information such as brightness and saturation which can lead to more representative clusters than just the color. Nevertheless, depending on the image some methods give better results than others. In general, the +xy spaces tend to give the worst segmentation, or even the best segmentation but just in few cases. This may happen because local information is not always useful because some objects are not homogeneously distributed nor connected in X and in Y, leading to have a preference for other type of information rather than the spacial one (figures 8 9.

The best method was GMM and then Kmeans and could be seen in figures 10 and 11. GMM may be the best because the distance is not euclidean and adapts normally depending on data. The worst method turned out to be watersheds because the number of minimums was imposed manually. Usually the algorithm itself defines minimums points searching connectivity between different objects. Additionally, the best K was 3 and may be due to the over-segmentation of our algorithm. A small k can decrease to some extent the over-segmentation and return a higher rate of Jaccard (figures 10 11 12 13) The results shown above prove that one of the most important limitations of the method is to distinguish objects surrounded by particles or to identify objects that are not homogeneous across their extension. In addition, objects with a lot of details in themselves are often over-segmented and a lot of small objects are segmented apart from the bigger objects which they belong.

A condition that can be related to the low performance of the algorithm is that an specific number of clusters is used to make the segmentation of all the images. This is not proper because, as explained before, depending on the image some methods would work better than the others for a specific number of clusters. Also, the lack of a big pre-processing on the images may explain the appearance of small objects,noise and over-segmentation obtained in the results. In relation with that, the over-segmentation makes that the union between the borders of the segmentation and the borders of the annotations became a bigger set than the intersection. Related to the evaluation strategy, a significant source of error is that many times the borders obtained with the sobel filter for the segmentation are not located in the same pixels that the borders obtained with sobel for the annotations. In consequence, the intersection of those two images will present several discontinuities and the Jaccard index will be low because of the big size of the union of those two images in comparison to the size of their intersection.

V. Conclusions

In conclusion, the algorithm proposed successfully develops a segmentation across several color spaces using different methods of clustering. Nevertheless, it has considerable limitations, due to the variety of the characteristics on the images, the necessity of having a arbitrary number of clusters for all the images makes it not good enough to generalize and to make the segmentation of all the images properly. Also, all of the methods tend to identify small details in the images and because of that over-segmentation is done. This can be improved by making a pre-processing on the images to decrease the amount of details which may cause to create a over-segmentation. This pre-processing could be to apply a Gaussian filtering to every image in order to smooth weak details.

A possible way to improve the evaluation strategy may be to establish a threshold of distance between the segmentation borders and the annotation rather than just applying a intersection between these two. In that sense, if the border of the segmentation is far more distant than the threshold, it might not be consider. But, if its inside the threshold it must be counted in the intersection (numerator) to calculate the Jaccard Index. In this way, the evaluation strategy will consider just the borders of the identify objects and not just evaluating the performance in finding the same boundaries at the same pixels as it was done for the strategy used. Finally, considering all the set of the annotations rather than just an arbitrary one could increase considerably the Jaccard index due to the fact that some of the annotations segmented specific details for every object.

References

[1] R. Szeliski, "Computer Vision: Algorithms and Applications." Springer, Sep-2010.

[2] P. Arbelaez, C. Fowlkes, and D. Martin, " The Berkeley Segmentation Dataset and Benchmark," David A. Patterson EECS at UC Berkeley, 2007. [Online]. Available: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/. [Accessed: 08-Mar-2019].

[3] N. Dhanachandra, K. Manglem, and Y. J. Chanu, "Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm," Procedia Computer Science, vol. 54, pp. 764–771, 2015.

[4] Y. Song and H. Yan, "Image Segmentation Techniques Overview," 2017 Asia Modelling Symposium (AMS), 2017.

[5] H. Tibshirani , "Gaussian Mixture Models." Stanford University, California, 2008.