Lab 06: Segmentation

Adrian S. Volcinschi M.
Universidad de los Andes
https://uniandes.edu.co/en
as.volcinschi@uniandes.edu.co

Abstract

Image segmentation is a very hard task to achieve. The first approach was to think of it as a clustering method since we didn't knew what we wanted exactly. The purpose of this is to separate different objects from the same image. This was done using the four principal clustering methods (Kmeans, Gaussian Mixture Models, Hierarchical clustering, and Watersheds) with different color spaces such as RGB, HSV, and Lab. The results showed that watershed had good outcomes if the markers were good enough. Otherwise, Gaussian Mixture Models and Kmeans showed decent results without using a lot of resources compared to Hierarchical clustering.

1. Introduction

The problem of segmentation has been studied since the early 1970's. It began as a split problem of either finding edges or segmenting the image. One of the first works in edge detection date back to 1973 with Sobel and his 3x3 isotropic gradient operator [7]. Later on in 1979 we would find the use of watershed for contour detection [1]. The state of art for segmenting images was seen as a clustering problem [2] up until the early 2000's when the boundary detector of Fowlkes was presented as a supervised problem [6].
The objective of this clustering approach of segmentation is to find centroids, in the case of Kmeans, GMM and Hierarchical Kmeans, or regional minima in Watersheds and assign labels depending on this so called centroids or regions. The final expected result is an image where every object is isolated from each other through a mask of labels that differentiate one from another. This problem is of immense relevance since it can help us divide an image or even a video into independent objects which can give us a lot of information about them and their surroundings.

2. Materials and Methods

The objective is to segment different images of the subsampled BSDS dataset found in `http://157.253.196.67/BSDS_small.zip`. Some pictures of this subsampled dataset are shown in figure 1.

Figure 1: Pictures of the subsampled BSDS dataset

To do this a function called segmentByClustering was created. The parameters of this functions are a RGB image, a clustering method (Kmeans, GMM's, Hierarchical Kmeans, and Watersheds), a color space (RGB, HSV, Lab, RGB+XY, HSV+XY, and LAB+XY), and a number K of clusters.

2.1. Clustering Methods

2.1.1 Kmeans

Kmeans is the simplest clustering method which consists of choosing K random points as centroids, assign labels to all other points depending on which centroid is closer to them using Euclidean distance and then recalculating the centroids. The idea is to iterate this many times until no significant changes in the centroids occur [9].

2.1.2 Gaussian Mixture Models (GMM)

Gaussian Mixture Models has the same intuition as Kmeans but has two big differences. First, it assumes the points distribute normally in K Gaussian probability distributions. Second, it doesn't assign a hard label like Kmeans but rather assigns a probability of being part of each Gaussian distribution [10].

2.1.3 Hierarchical Kmeans

Hierarchical Kmeans consist of constructing a hierarchy of clusters. This can be done in two ways: bottom-up and top-down. The bottom-up process assumes each data point is a cluster on its own and starts joining clusters based on distance. The top-down process assumes all the data points form a single cluster and then it applies Kmeans with a higher K than 1 and starts to divide the single cluster into multiple ones until each data point is a cluster on its own [5]

2.1.4 Watersheds

Watershed transform algorithm consists of picking some regional minima points first. When this points are chosen, either by hand or automatic methods such as h-extendedmins, we impose them as absolute minimums so they are the only minimums in the image. If we think an image or the gradient of an image as a topological surface with holes in this minimums we can flood the surface and get the divisory lines. This divisory lines will separate the objects and create the corresponding superpixels [4].

2.2. Color Space

Color space is an important feature to take into account since different spaces gives us different information about the image. For example, the HSV+XY takes into account the Hue, Saturation, and Value as well as positional values of the pixels (x,y). Since the amount of channels vary between 3 and 5, depending if the spatial information is taken into account, I needed to create a way to control the weights of each channel. Additionally, since the images have a size of 481x321 the X and Y channels are going to go from 1 to 481 and 1 to 321 respectively, this channels would have a greater weight than the color channels that go from 0 to 255 in the case of RGB. To solve this I transformed the range of each channel, color and spatial, to [0,255] and then took the average of each channel in every pixel to get a 481x321 image rather than 481x321x(3 or 5).

2.3. Evaluation method

Since an evaluation method needed to be created, the metric I thought was an intersection of histograms between the resulting segmentation and the third groundtruth segmentation. Since my segmentation could have a different amount of regions depending on the K parameter and the groundtruth didn't I needed a way to control this and make a somewhat reliable metric. To do this I took the image with lower number of regions (segmentation or groundtruth) and made its histogram. For the other image, which had more regions, I created a histogram with n bins equally distributed where n is the number of bins of the previous histogram. Additionally, since my labels of segmentation are not necessarily in the same order of those in the groundtruth I sorted the resulting histograms to then calculate the intersection between them. Finally, I normalized this intersection over the size of the image (481x321) to get a number from 0 to 1 that represents the information better than a number of pixels.

3. Results

Using the function for different images with different images, color spaces, clustering methods, and number of clusters I was able to obtain some decent results alongside very bad ones. For example, a very good result is shown in figure 2 while a very bad result is show in figure 3.

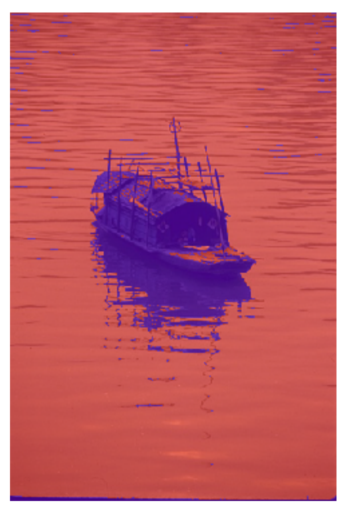

Figure 2: Segmentation with kmeans and K = 2

Image	8143				
K	3	Best		Worst	
Colorspace	RGB	Watershed	0.8911	Hierarchical	0.6533
	HSV	Watershed	0.9726	Hierarchical	0.8395
	Lab	Kmeans	0.9102	Hierarchical	0.7738
	RGB+XY	Kmeans	0.8875	Watershed	0.6361
	HSV+XY	Kmeans	0.9398	Watershed	0.6356
	Lab+XY	GMM	0.996	Watershed	0.6353

K	5	Best		Worst	
Colorspace	RGB	Kmeans	0.9022	Hierarchical	0.6772
	HSV	Watershed	0.8806	Hierarchical	0.7368
	Lab	Watershed	0.8405	Hierarchical	0.6080
	RGB+XY	Kmeans	0.8702	Watershed	0.5040
	HSV+XY	GMM	0.7977	Watershed	0.5170
	Lab+XY	Kmeans	0.8236	Watershed	0.5682

K	7	Best		Worst	
Colorspace	RGB	Watershed	0,8239	Kmeans	0,7611
	HSV	Kmeans	0,8131	Watershed	0,7403
	Lab	GMM	0,7675	Kmeans	0,7415
	RGB+XY	Kmeans	0,7529	Watershed	0,3711
	HSV+XY	Hierarchical	0,8032	Watershed	0,4925
	Lab+XY	GMM	0,7328	Watershed	0,4581

Image	41004				
K	3	Best		Worst	
Colorspace	RGB	Kmeans	0,944	Watershed	0,6111
	HSV	Watershed	0,9178	GMM	0,7038
	Lab	GMM	0,9872	Hierarchical	0,5354
	RGB+XY	GMM	0,9829	Hierarchical	0,8219
	HSV+XY	Kmeans	0,8774	Watershed	0,609
	Lab+XY	GMM	0,8947	Watershed	0,6117

K	5	Best		Worst	
Colorspace	RGB	Kmeans GMM Hierarchical	0,6085	Watershed	0,4263
	HSV	GMM Hierarchical Watershed	0,6085	Kmeans	0,5702
	Lab	All	0,6085	All	0,6085
	RGB+XY	GMM Hierarchical Watershed	0,6085	Kmeans	0,5617
	HSV+XY	Hierarchical	0,6085	Watershed	0,4257
	Lab+XY	All	0,6085	All	0,6085

K	7	Best		Worst	
Colorspace	RGB	Kmeans GMM	0,6085	Watershed	0,4263
	HSV	All	0,6085	All	0,6085
	Lab	All	0,6085	All	0,6085
	RGB+XY	All	0,6085	All	0,6085
	HSV+XY	GMM Hierarchical	0,6085	Watershed	0,4259
	Lab+XY	All	0,6085	All	0,6085

Table 1: Variation of color space, method and K for two images

Figure 3: Segmentation with watershed and K = 6

To see if the stated metric is a good I varied the K, color space, and method and run the metric on 2 images. The table can can be found in table 1. According to this the best K, color space and method for the first image, 8143, is GMM with K = 3 in Lab+XY. This segmentation is shown in figure. In a similar way, the best combination for the second image, 41004, is GMM with K = 3 in Lab. This images and their corresponding segmentations can be found in figure 4

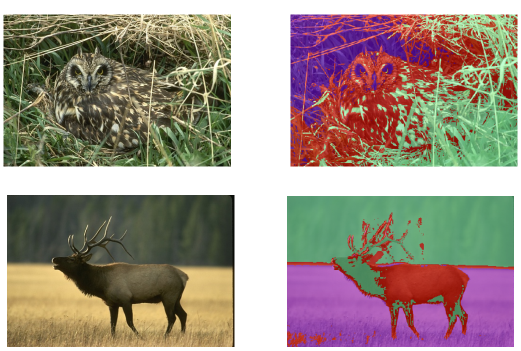

Figure 4: Two images (left) and their best segmentation according to the previously defined metric (right)

4. Conclusions

As it was shown, the proposed metric is not very good for two big reasons: In first place, the fact of it needing to have both histograms with the same amount of bins loses a lot of information. In second place, it doesn't consider the spatial information at all. This means the metric only looks at the size of the regions in both groundtruth and segmentation but not if they correspond spatially. For this to be solved it would be better to evaluate using spatial intersection between the masks of segmentation and groundtruth and calculating the Jaccard index [8]. You can see Watershed works really well for color spaces without spatial information and when you add that information it becomes the worst method in almost every case. Additionally, Hierarchical gives a bad result when it doesn't have spatial information as expected because it clusters using distance. When we add the spatial information it is almost never the worst algorithm compared to color spaces without spatial information where it was recurrently the worst. We can also notice that the basic clustering algorithms (Kmeans and GMM) give a good result in a very good part of the cases. In conclusion, Kmeans and GMM are very good unsupervised segmentation methods that don't require a lot of processing and give consistent results.

In terms of effectiveness of this approach of segmentation it can be said that it leaves a lot to be said. Most of the segmentations are not even close to the annotations made by humans. On the other hand, it is a very efficient way of doing a segmentation since it doesn't require labels to train and it does it really quickly. That unsupervised training comes at the cost of effectiveness since it has been shown that supervised training with neuronal networks are the best approach right now, like Mask R-CNN [3]. The only method that is really slow and resource expensive is the hierarchical clustering because it does a bottom-up aggregation and consumes a lot of memory in the process. The image had to be downscaled to half the original size for it to be possible to run. The other methods ask more of CPU rather than RAM. Another constrain these methods have is the fact of not converging to an absolute minima but instead converge into local minima. This means we can never be sure if we found the optimal answer to the clustering problem.

References

[1] S. Beucher and C. Lantujoul. Use of watersheds in contour detection. volume 132, 01 1979.

[2] M. Celenk. A color clustering technique for image segmentation. *Comput. Vision Graph. Image Process.*, 52(2):145–170, Sept. 1990.

[3] R. Hu, P. Dollár, K. He, T. Darrell, and R. B. Girshick. Learning to segment every thing. *CoRR*, abs/1711.10370, 2017.

[4] A. A. Kaur. Image segmentation using watershed transform. *International Journal of Soft Computing and Engineering (IJSCE)*, 03 2014.

[5] D. Kilitcioglu. Hierarchical clustering and its applications. `https://towardsdatascience.com/hierarchical-clustering-and-its-applications-41c1ad4441a6`, 2018. [Online; accessed 07-03-2019].

[6] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using brightness and texture. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1279–1286. MIT Press, 2003.

[7] I. Sobel and G. Feldman. A 33 isotropic gradient operator for image processing. *Pattern Classification and Scene Analysis*, pages 271–272, 01 1973.

[8] K. Tecnomo. Jaccard's coefficient. `https://people.revoledu.com/kardi/tutorial/Similarity/Jaccard.html`, 2015. [Online; accessed 07-03-2019].

[9] A. Trevino. Introduction to k-means clustering. `https://www.datascience.com/blog/k-means-clustering`, 2016. [Online; accessed 07-03-2019].

[10] J. VanderPlas. In depth: Gaussian mixture models. `https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html`, 2016. [Online; accessed 07-03-2019].