Laboratory 7 - BSDS500 Benchmark

Guillaume Jeanneret
Universidad de los Andes
Cra 1 #18a-12, Bogotá, Colombia
g.jeanneret10@uniandes.edu.co

Jorge Madrid
Universidad de los Andes
Cra 1 #18a-12, Bogotá, Colombia
ja.madrid2714@uniandes.edu.co

Abstract

The database BSDS500 is a tool for researcher in the fields of image segmentation and boundary detections. In this laboratory, this database is used to check the algorithm created in the past one. The results shows that the K-Means *algorithm gives a ODS=0.36, a OIS=0.40 and a area of 0.23. Watershed produced an error so the result of the precision-recall curve hasn't been achieved.*

1. Introduction

The Berkeley Data Set and Image Segmentation Resources (BSDS) is an image database composed of 300 (BSDS300) or 500 (BSDS500) images. Each picture has the original image, the ground-truth and the benchmark. This dataset has the goal of giving the researchers in the image segmentation and boundary detection field an empirical basis [1]. Furthermore, the BSDS500 provides all investigating groups the same workspace, equal possibilities and the perfect way to compare their the algorithms.

The Berkeley dataset does not provide only the pictures. It also provides a way to compare the result of the method using the Precision-Recall curve (PRc)[1]. The ideal result of this curve is a point in the coordinates (1,1). For a precision of 100%, a recall of 100% is achieved. This means that no false negatives and no false positives are detected.

Because the annotations are made by a group of people, a perfect segmentation can't be achieved. Then, the goal is to accomplish a curve that compares to the human performance.

2. Materials and Methods

2.1. Database

The database used in this laboratory is the BSDS500, composed of 500 images, partitioned in 3 categories: train, validation and test. The train category has 200 pictures, the validation folder has 100, and the test has 200.

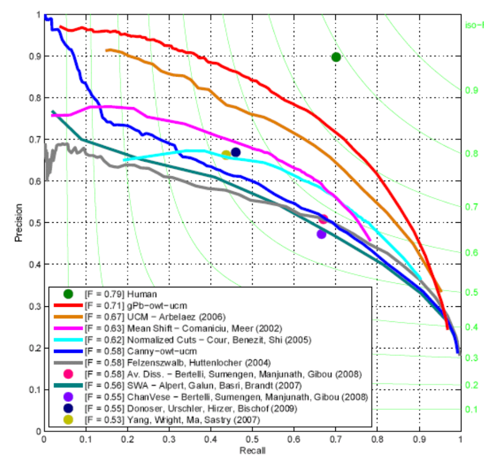


Figure 1: Actual PRc evaluation of segmentations algorithms done in BSDS300 [2].

2.2. Methods

The laboratory was done using the code of our last assignment [3].

```
segmentByClustering(rgbImage,...
        featureSpace,method,clusters);
```

"*segmentByClustering* is a Matlab function which implements image segmentation according to certain criteria given by the user as parameters. For instance, *rgbImage* is the color image, *featureSpace* is the 'rgb', 'lab' or 'hsv' color space with or without '+xy', meaning the incorporation of the spatial components for classification, *method* -which is self explanatory- with possibles values of 'k-means', 'gmm', 'hierarchical' and 'watershed' and, finally, the number of clusters" [3]. Thus, the function performs various forms of segmentation according to different classification criteria, and can be easily called upon in a script to segment a number of images automatically.

As noted in [3], the methods with the best performances were k-means and watershed. Futhermore, the

GMM method was used to segment the database. Then, these methods were used to classify using the following *k*'s:

```
k=round(2.5:2.5:30);
```

2.3. Evaluation

The evaluation of our segmentation methods were done using the PRc curve shown in figure 1. This was accomplished using the *.m files* provided by the database in the section *bench_fast*.

3. Results

Some examples of the segmentations are as follow. Figure 2 exemplifies the qualitative performance of the segmentation algorithms on an image from the train section of the dataset. Note that using watersheds results in an oversegmentation of the image, as in figure 2c. This may be due to the lack of direct control over the number of regions into one segments the image. Segmentation by *k-means* results in a segmentation which is closer to human ground truth, which is evident from figures 2b and 2d. An important feature to notice about our segmentations is the fact that wide background areas are commonly fragmented into subregions of similar size. A positive result is achieved with an almost complete continuous segmentation of the starfish. Bear in mind that figure 2 belongs to the train category, which means that, at least in principle, classification should be optimized for it.

Figure 3 exemplifies the results of the segmentation on an image from the test category, for which the classifiers were not optimized. Again, watershed results in oversegmentation, as in subfigure 3c, while the use of *k-means* yielded results closer to human annotations, as shown in subfigures 3b and 3d, at least qualitatively. *K-means* still has difficulties recognizing as different entities the persons sitting together, and may divide into different groups patches of rock in the snow that belong to the background.

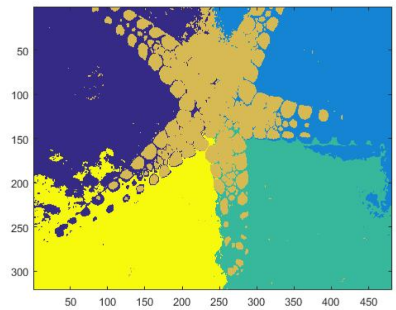
It's important to clarify that GMM method was used to produce a result in the BSDS500, it was our backup plan. Because of some error (we actually don't know what happened), the process didn't converge. It reached at least 34 hours of processing and stop. Therefore, the algorithm didn't gave us a result.

The results of the segmentation using K-means and *RGB+xy* are given in figure 4 and tables tables 1 and 2. The results of the segmentation using K-means and *RGB* are given in figure 5 and tables tables 1 and 2.

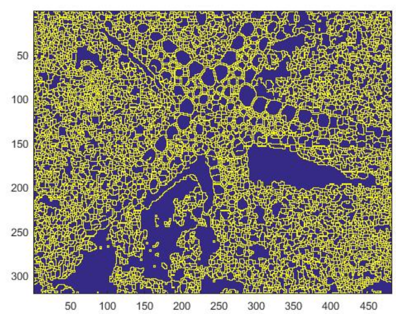
4. Discussion

The methods used in this laboratory and the results shown in figure 4 and tables 1 and 2 evidence that the implementation of this methods gives a generalized segmentation. This means that the results given by algorithm are

(a) Original image without any segmentation



(b) Segmentation using *k-means*.

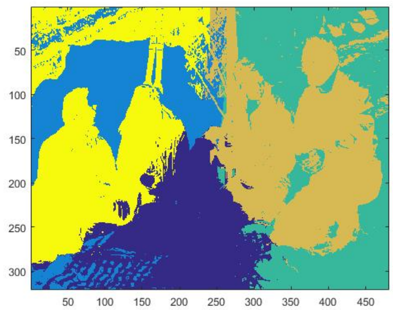


(c) Segmentation using watersheds.

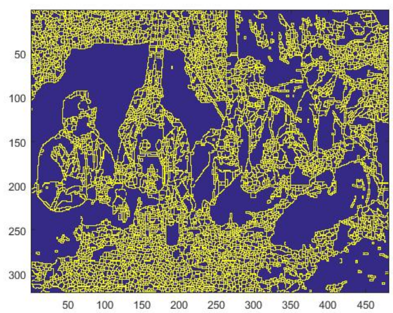
(d) Segmentation from human ground truth.

Figure 2: Segmentation of an image from the train section of the dataset.

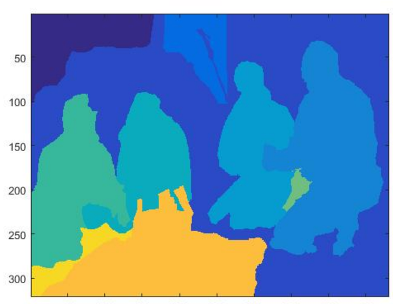
(a) Original image without any segmentation



(b) Segmentation using *k-means*.



(c) Segmentation using watersheds.



(d) Segmentation from human ground truth.

Figure 3: Segmentation of an image from the train section of the dataset.

consistent for a small change in the value of clusters *k*.

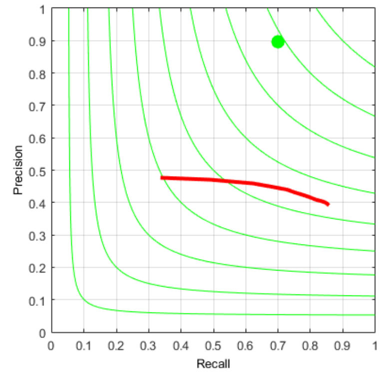


Figure 4: PRc - Result from classification using k-means (RGB+xy) and the test database.

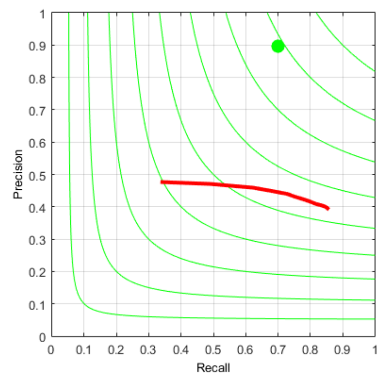


Figure 5: PRc - Result from classification using k-means (RGB) and the test database.

Database	ODS	OIS
train	0.38	0.42
val	0.37	0.41
test	0.36	0.40

Table 1: ODS/OIS resulting table from ground-truth covering using k-means segmentation.

Database	Area
train	0.23
val	0.22
test	0.23

Table 2: Area beneath the PRc from the partition of the database using k-means segmentation.

The segmentation method of watershed has turned out into oversegmentation and more limited results. Even

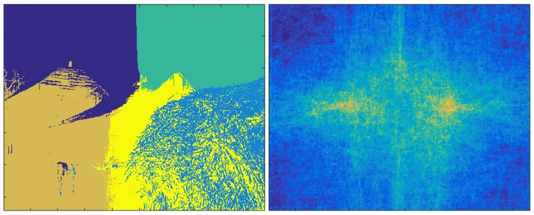


Figure 6: *K-means* favors the segmentation into regions of roughly the same size. **Left** An exemplar segmentation in which the image is divided into regions of the similar size (regardless of the continuity of the structure). **Right** Average of the magnitude of the gradient of the segmentations in the test database for a fixed *k*. Note the strong vertical and the somewhat diffuse horizontal lines indicating the preference for segmentations into quadrants.

though this contradicts our previous results, we acknowledge that improving the performance of watersheds is difficult if one wants to avoid the need for user-given hyperparameters such as minima or constrast differences.

K-means, although outperforming watersheds, do face limitations as well. The incorporation of spatial information into the similarity definition of the classifier has resulted in a bias for segmenting regions of similar size and shape. This appeared in the background of subfigure 2b, but is better exemplified by figure 6. This figures adds another example of division into regions of similar size, regardless of the continuity of the original objects in the image. Plus, the average of the gradient over all 200 segmentations in the test set demonstrates a strongly favored division into quadrant-like regions.

To perform the algorithm, each coordinate -for the case of *xy*- should take less force. This means that multiplying the coordinates by some scalar C, $0 < C < 1$, so when the distances between vectors is calculated, the space information will not be as important as before. Also, including some more information like texture will be profitable, giving them some regulation constants like C.

Furthermore, filtering each segmented picture with a filter. This filter will replace the central pixel with the category i with higher frequency. This method could prevent that a pixel withing a window of a different classification will be part of another structure. The problem is that this filter can delete small objects if the window size isn't choose appropriately.

5. Conclusion

- Getting a similar result in the ODS and OIS table suggest that the chosen method of segmentation gives a more general result and a similar results given a number of clusters fix.
- Even for a Unsupervised Clustering method, the results given by K-means are great because there's no training in the algorithm
- Getting a backup plan it's important, but getting partial result between the algorithm running will prevent that, after an error is produced, not all the data generated is will be lost.
- Beating Pablo is very difficult -he defeated Malik in his own game-, more advanced methods are needed and more information for a better segmentation.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011.

[2] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265.

[3] J. Madrid and G. Jeanneret. Laboratory 6 - superpixels. =https://github.com/guillaumejs2403/Vision17/tree/master/Lab6-Superpixels, 2017.