Berkeley Segmentation Data Set and Benchmarks 500 (BSDS500) Laboratory

Diego Martínez
Universidad de Los Andes
Cra 1N 18A-12 Bogota (Colombia)
da.martinez33@uniandes.edu.co

Felipe Torres
Universidad de Los Andes
Cra 1N 18A-12 Bogota (Colombia)
f.torres11@uniandes.edu.co

Abstract

Research in Computer Vision has three main areas of interest and one of them is the segmentation of images. The segmentation problem has been tried to solve via different approaches but still the most commonly used are the implementation of clusters to re-group pixels that contain about similar information, however this is not the only way that this is done but the most recurrent. Among the clustering methods actually known, K-Means and Gaussian Mixture Models are the ones that stand out from the others due to their accuracy and performance, the last being a little bit faded due to their slow time to process an image. In this article a matlab function was developed to segment images according to the clustering approach, said function allows the user to select whether if the segmentation is going to be done by K-Means, GMM or the watershed transform, and the color space over which clustering is going to be taken into account. Then said methodology was tested on Berkeley's BSDS 500 and the results described the better performance for GMM at HSV+XY color space.

Key words: Clustering, K-Means, GMM, Watershed

1. Introduction

In Computer Vision the design of algorithms and methodologies to classify and detect objects inside images needs a huge amount of data to verify the accuracy and performance of said methods, yet; before the people working in the field became aware of the importance of sharing (algorithms and data), most experiments and results could not be replicated easily and research in the subject was getting stuck due to the little (not to say null) capacity to compare algorithms and methods. With the arrival of internet and *The internet of things* people started to become aware of the advantages that could be achieved by sharing frameworks and data to perform experiments; on segmentation and regrouping the release of **BSDS300** and later on **BSDS500** (Berkeley Segmentation Data Set) by UC Berkeley, pushed forward the methods and state of the art on said topics. These databases contain 300 and 500 images respectively, yet; the importance of these relys on the human made annotations to every single one of the images, thus providing the researchers a capability to compare the obtained results with the human made segmentation.

BSDS500 was first released with Arbelaez, Maire, Fowlkes and Malik's Contour Detection and Hierarchical Segmentation [2], this method provided a high performance contour detector, a method to transform any contour signal into a hierarchy of regions and finally extensive quantitative evaluation helped with the release of the dataset.

Traditionally, segmentation in computer vision has been regarded as a topic to be treated by clustering data, however several methodologies exist for clustering, yet; the most important ones and the ones used in this work are K-Means and Mixture of Gaussian Models. In addition to these methods, research has showed that aided with the values of intensity given for each image, a hierarchich segmentation can be done by the implementation of the watershed transform in cooperation with the imposition of regional minimums in the image. An example of these hierarchichal segmentations is presented in the *Contour Detection and Hierarchical Segmentation* by the authors mentioned before[2]. However, these methods are expensive and slow and nowadays with the revitalization of Neural Networks starting with AlexNet by Alex Krizhevsky [6], several segmentation strategies have been purposed much more efficient and accurate, such as R-CNN [5] and Faster R-CNN[4].

2. Methodology

A matlab function named ***segment_by_clustering()*** was developed to segment a desired image. This function allows the user to have different feature spaces such as 'rgb','hsv','lab' and the combination of said spaces with the information of spatial localization for each pixel (such as 'rgb+xy'). The clustering methods vary from K-Means, Watershed and GMM (Gaussian Mixtures Model). Finally the last input of this function answers to the desired number of clusters that the user may want to have, briefly we are

going to present in detail the parameters of this function.

2.1. Segmentation Methods

Segmentation Methods avialable to implement in this function are: K-Means, GMM, Watershed

- K-Means. K-Means is an iterative method that assigns n-obsertvations to exactly one of the K-clusters defined by centroids, where K is the number of clusters desired before the initialization of the algorithm. It follows the next steps:
  1) Choose k initial cluster centers.
  2) Compute point-to cluster centroid distance for all observations to each and every centroid.
  3) Each observation is assigned to the cluster with the closest centroid.
  4) Compute the average observations in each cluster to obtain K-new centroid locations.
  5) Repeat iterations 2 to 4. In computational terms this is done until cluster assignements don't change or a certain number of iterations is achieved
  K-Means clustering method is also known as Lloyd's algorithm, also this is used as the main tool to cluster data, however the application of this method requires a lot of computer memory making it expensive to implement.[7]

- GMM. A Gaussian Mixture Models (GMM) are composed of K-Multivariate normal density components, each one of these components has a d-dimensional mean; a d-by-d covariance matrix and a mixing proportion; mixing proportion j determines the pfraction of the population composed by the component j, $j = 1,...k$. As the function is developed on MATLAB, to fix a GMM the following algorithm needs to be followed:
  1) Extract colour and coordinate matrixes and turn them into vectors, then concatenate them into a new Matrix.
  2) Fit the GMM model by applying MATLAB function fitgmdist with input parameters the matrix from before and the K number of clusters.
  3) Once the GMM model is out as the output of fitgmdist, use cluster function; using the GMM model and the features Matrix.
  4) Reshape.
  GMM models present the same issues as K-Means, the output of this algorithm will get a good segmentation of the image, however it is really slow and sometimes it can throw errors because it is not easy to find the covariance in less than 100 iterations (These iterations are part of the fitgmdist function).[8]

- Watershed. A watershed is a line that determines where from two basins a water drop will go. In image processing, the intensity of a grayscale can be seen as a topographic map, in which; the higher the intensity, the higher a peak will go and in regions where the intensity is low a basin will be created. To imagine how the transform works we think as if the topographic map is being flooded, and everytime two basins are about to be joint together we can draw a line to separate them, that is the watershed.[9] However the simple version of this transform creates a distorted image in which the number of superpixels is way too big to even be considered as useful, therefore; we implement the marker controlled one. In this one, we select a K (height/value of intensity) which will be the threshold for the minimum values of intensity, and every value below it will be considered as K, thus creating a cleaner partition map.

Two color spaces were selected to run in the dataset, 'HSV+XY' and 'RGB'. HSV+XY provides a linear descriptor for colour and the XY coordinates aid with the spatial positioning of items inside the image. RGB colorspace is the image without any modification, we set RGB because we wanted to compare how an uniform colorspace (HSV) with coordinates would create differences from the segmentation with the image.

Also we selected GMM and K-Means. These two methods can be **really** slow, however they are more accurate than watershed transform, more importantly the clusters can be directly controlled with the input, yet with watershed transform it is uncertain at what level the image will be segmented with the K selected.

2.2. Test Metodology

2.2.1 Dataset

Tha daset was composed of 500 images from Berkeley Segmentation Dataset. All images had 481x321 pixels represented in RGB colorspace and saved in .jpg format [1]. The 500 images are organized or divided in three different subsets in order to developt the test methodology and obtain optimazed parameters that give best segmentation performance:

- Train dataset: Which describes 200 images from original 500 images that are used in parameter tunning and selection of parameters that maximize the algorthm performance
- Validation dataset: Which describes 100 images from original 500 images, that is useful to optimize and improve algorithm performance
- Test dataset: Which is composed of 200 images and is useful to test and evaluate robustness of optimized

algorith if it is evaluated in images totally different to Train and Validation images

Each image from the dataset must have a groundtruth or image annotations. For the Berkeley Segmentation Dataset it was cosntructed with human hand made anotattions for segmentation from five different people and taking the averege of them[1].

To test the function, it was run on the Dataset and 5 segmentations per Image were obtained. Said segmentations were stored in a cell array named 'segs' in a .mat file wich then would be read by the function to test the benchmark.

To get the desired segmentations four experiments were purposed, yet they all had in common the same parameter, K number of clusters would vary from 5 to 9.

- First Experiment: The first experiment consisted on implementing K-Means on every Image using the feature space HSV+XY.
- Second Experiment: The second experiment consisted on implementing GMM on every Image, using the feature space HSV+XY.
- Third Experiment: The third experiment consisted on implementing K-Means on every Image using the feature space RGB.
- Fourth Experiment: The fourth experiment consisted on implementing GMM on every Image, using the feature space RGB.

One may wonder why marker controlled watershed transform was not implemented. To adress that issue we go back to the outputs of the function, it is true that watershed transform is **way** faster than the implementation of K-Means or GMM, however the output will not always have the K number of clusters that were desired and the image may consist mainly of superpixels or background.

After implementation of the function over all the images, there was necesary the use of already implemented methodologies of evaluation. In this case, we utilized benchmarks funcions created by Malik and its colleagues at Berkeley Segmentation Data Set and Benchmarks 500 (BSDS500). Benchmark unction ***boundary_bench()*** allow us to obtain precision of implementation through Precision-Recall curves, besides other performance classification algorithms like best Precision and Recall values, Rand indez and Variance information. Worth to explain that precision-recall curves gives proportion meassures correctly labeled in a segementation. Results of this curves are shown at Figures 6, 7 and 8. Benchmark were applied to segmentation results of each combination of color space and segmentation method mentioned previously at the experiments.

3. Results

3.1. Implementation of the Function

The function implementation is fairly easy to do, you just need to bear in mind that you must input the correct name for each clustering method (however it may not affect if you write K-MEANS or k-means, capital letters won't affect the performance). To begin with we can briefly check the documentation to have a clear point of view for the function. To

```
>> help segment_by_clustering
       FUNCTION SEGMENT_BY_CLUSTERING
       OUT = segment_by_clustering(RGB_IMAGE,FEATURE_SPACE,...
       CLUSTERING_METHOD,NUMBER_OF_CLUSTERS)
       RGB_IMAGE. Rgb_image is the input image, it represents what you
       would want to segment.
       FEATURE_SPACE.Feature_Space sets the space of representation for the
       segmentation, it can either be RGB,L*AB,HSV; as well RGB+XY,L*AB+XY
       and HSV+XY. XY means that the input will be considering X and Y
       distances in the image, meaning localization.
       CLUSTERING_METHOD. Clustering method adjusts the way the clusters
       will be selected. It can either be 'K-Means','GMM','Watershed'. If
       watershed is selected, the number of clusters selecetd as an input
       will need to be a value from the range [0,255].
       NUMBER_OF_CLUSTERS. Number of Clusters adjusts the 'K' for each
       clustering method that requires it, if no argument for
       Number_of_clusters is input, the default will be 3.
       You may want to set the dynamic range of the image in imshow as [].
       Then I highly suggest you to use colormap(colorcube) to see the
       output.
```

Figure 1. Help Command in matlab containing the documentation for the function

continue as we know the syntaxis needed in order to have a well functioning image we present the results obtained of our experiments in just one single image and one of the five K variations used, this K variation will be K = 7.

3.2. First Experiment

With the first experiment we wanted to check how the feature space affects the output in K-Means, the output is shown below:

Figure 2. Comparisson between using K-Means with feature spaces HSV+XY and RGB

3.3. Second Experiment

With this experiment we wanted to check again how the feature space affects the output but using GMM.

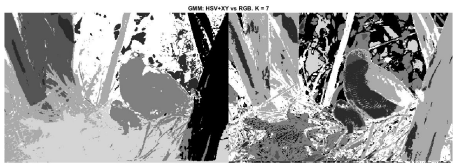

Figure 3. Comparisson between using GMM with feature spaces HSV+XY and RGB

3.4. Third Experiment

Now after checking how different feature spaces affect the same method, we checked how different clustering methods affect the output using the same feature space.

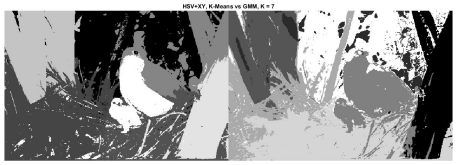

Figure 4. Comparisson between using GMM and K-Means with feature space HSV+XY

Finally we check the last experiment

3.5. Fourth Experiment

At last we want to check how the output varies from K-Means to GMM using the image without any extra feature spaces.

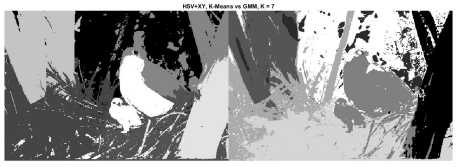

Figure 5. Comparisson between using GMM and K-Means with feature space RGB

On the other hand, as can be seen in Precession-Recall curves Figures 6, 7, 8 and 9 bencharks were implemented correctly for first, second, third and fourth experiments, respectively.

4. Discussion

Firstly, visual results of segmentation of Figures 2, 3 and 4 gives us an idea of which of segmentation methodology had the best performance. Segmentation results for K-means describes and over fitting of clasification of pixels at every image. An example of this is the Figure 4, at which segmentation by K-means divedes the bird in to different objects, yet it represent a unique object at the image. On the other hand, GMM gives a better segmentation regarding it don't separated in 2 different clusters pixels of the same object. This could be the result of non-homogeneous and soft-clustering that represents GMM when implemented at

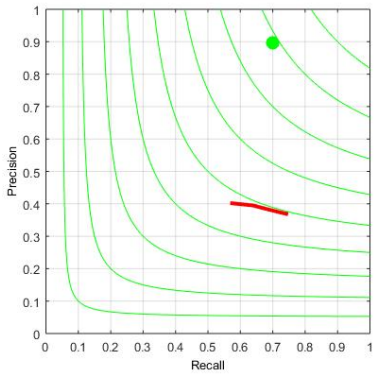

Figure 6. Precision-Recall curve of HSV+XY and K-means

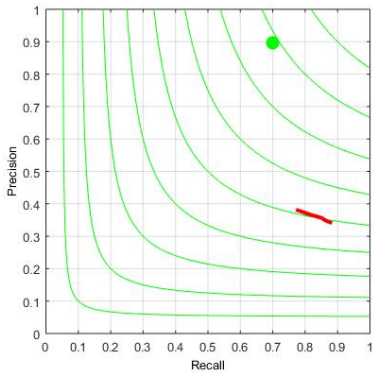

Figure 7. Precision-Recall curve of HSV+XY and GMM

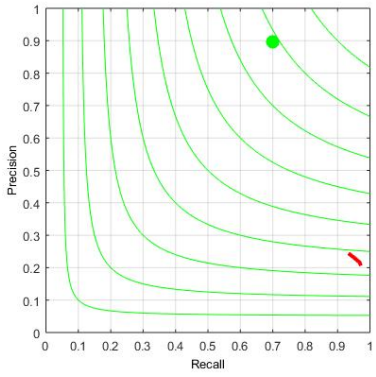

Figure 8. Precision-Recall curve of RGB and K-means

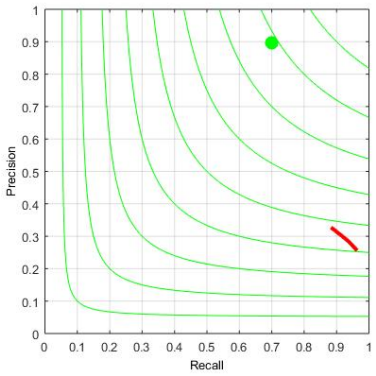

Figure 9. Precision-Recall curve of RGB and GMM

a segmentation problem.

However, after anilize Precision-Recal curves that result from benchmark carried through, differences between the two segmentation methods are not too significatives. Actually, the real difference excel from the two types of color spaces. As seen in segmentation example images of Figures 2 and 3, the most effective color space over which segmentation should be done is HSV+XY over RGB. This is explain because, a more larger and richer descriptor of the image is bieng used. HSV+XY give more information than RGB as to Hue, Saturation and Brightness, which is give more data than just color or intensities, and also adds spatial information with X and Y coordinates.

Finally, the best method of segmentation,from the methods implemented in the present research, is GMM clustering over a HSV+XY color space. At the end this method can be limited at its performance of precision and recall, which are minimal comparing to other methods like global contours detector and UCM [**?**][2]. Many artifacts ar added to the segmentation related to overclustering of the image at which pixels from the same object are assigned as part of two different objects.

5. How to Improve

There are several different methods to segment Images that do not require the same computational power as GMM and K-Means, however these two methods can be improved by switching certain parameters.

In order to improve performance in time we can lesser the number of clusters that will be required, say from 2 to 7. Another way to improve this is by restraining the number of iterations of the method, it may not converge at covariances or centroids but the speed would be faster than it is achieved at the moment.

Now if we want to improve the accuracy and precision of the methods we can look at state of the art segmentation methods. One way to approach the segmentation method can be MCG (Multiscale Combinational Grouping). This method relies on selecting regions and forming masks; once the masks are formed a hierarchichal segmentation is done by using different scales; after the segmentation is complete, the output images are resized and realigned to have about the same size of the original and the answers with the most frequency will have an edge with a higher intensity. [3]. Another approach can be the implementation of convolutional neural networks (CNN). By just mentioning CNN we have a high variability of methods possible to implement the segmentation, therefore we will be now talking about Faster R-CNN. Faster R-CNN is a method designed by Ross Girshick, that using selective search method[10] to obtain region purposals, runs a CNN pre-trained in ImageNet and fine tunes it in Pascal VOC challenge[5]. This method is highly slow, therefore Girshick a year after it's publication purposed the faster version of it, being Faster R-CNN. Faster R-CNN runs without the need to use external methods to obtain region purposals (using sliding window) and Bounding Box Regresion, then in difference to R-CNN, it does not use a SVM vector as the last layer of the ConvNet but it switches to a soft max layer at the end of the CNN instead of the SVM implemented before by Girshick[4]

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation.

[2] 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.

[3] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Computer Vision and Pattern Recognition*, 2014.

[4] R. Girshick. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[7] S. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inform. Theory*, 28(2):129–137, 1982.

[8] G. McLachlan and D. Peel. Finite mixture models. *Wiley Series in Probability and Statistics*, 2000.

[9] F. Meyer. Topographic distance and watershed lines. *Signal Processing*, 38(1):113–125, 1994.

[10] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013.