EC 504 Project Report

Image segmentation using max-flow min-cut algorithm

Zhenghao Sun szh1007@bu.edu

Shu Yang conlany@bu.edu

Xingyu Chen chxy517@bu.edu

Xingjian Zhang zhangzxj@bu.edu

Figure 1. Example result of image segmentation

1. Task

The task of this project is to develop an efficient and robust image segmentation method that can separate the foreground and background of the image. Image segmentation is a crucial task in computer vision and has numerous applications, for example, object recognition, image editing, and computer-aided diagnosis. Our project focuses on implementing the max-flow min-cut algorithm, and we will also compare other popular machine learning algorithms such as the K-means algorithm and the Gaussian Mixture Model, also known as GMM, with the Expectation Maximization (EM) algorithm.

The task consists of the following sub-tasks:

1. Developing a user-friendly interface to upload input images and enable the user to manually select the source and sink nodes for the max-flow min-cut algorithm, ensuring an efficient and accurate segmentation process.
2. Using the K-means method to segment images and evaluating its benefits and drawbacks with regard to the clustering parameters and classification areas it produces.
3. Using the Expectation-Maximization (EM) algorithm and the Gaussian Mixture Model (GMM) as an alternate method for segmenting images, exploiting their probabilistic character to get around some of the drawbacks of the K-means algorithm.
4. Implementing a click-based interaction technique for users to manually choose the source and sink nodes, emphasizing the significance of these decisions for the segmentation's success and their bearing on the effectiveness and precision of the algorithm.
5. Analyzing the segmentation boundaries' correctness, contrasting the performance of the GMM and max-flow min-cut algorithms on the same input picture, and debating which approach to use depending on the particular job and properties of the data.
6. Putting the project's draft code and documentation in a publicly available GitHub repository to ensure the findings' openness and repeatability.

2. Related Work

In this section, we review the existing literature and techniques related to image segmentation and provide a background for our project. Recently, researchers have done a lot of research on image segmentation, and they have developed various methods to solve these problems. For our project, we focus on two specific methods: Gaussian Mixture Models (GMM) with Expectation-Maximization (EM) algorithm and Graph Cut Algorithm.

1. Gaussian mixture model and expectation maximization algorithm:
GMM is a probabilistic model that has been widely used for image segmentation tasks [1]. It

represents a mixture of multiple Gaussian distributions to approximate complex and overlapping data distributions. The EM algorithm iteratively refines model parameters with the aim of maximizing the likelihood of the observed data. Several studies have demonstrated the effectiveness of GMM-based methods in image segmentation [2, 3]. However, some limitations include the need for initialization, sensitivity to the number of components, and potential problems with local optima.

2. Graph-cut algorithm:
The Graph-cut algorithm has become popular in the computer vision community due to its effectiveness in various image segmentation tasks [4, 5]. These algorithms aim to partition a graph into disjoint subsets by minimizing a specific energy function, which usually includes terms related to data fitting and spatial smoothness. Recent advances in graph cutting methods address various challenges such as computational efficiency, handling of multi-label problems, and incorporation of high-level information [6, 7].

3. Comparing and mixing methods:
Several studies compared the performance of GMM and graph cut methods on image segmentation [8, 9]. These comparisons show that each method has its advantages and disadvantages, depending on the nature of the data and the specific segmentation task. Some researchers have proposed hybrid methods that combine the advantages of GMM and graph cutting techniques, resulting in improved segmentation results [10, 11].

In summary, the existing literature highlights the potential of GMMs and graph cut algorithms for image segmentation tasks. Our project aims to implement, compare and analyze these methods, taking into account their strengths and limitations. In doing so, we seek to contribute to ongoing research in image segmentation and provide insights into the applicability of these methods to various real-world scenarios.

3. Approach

Many methods have been innovated to solve this problem. For example, we can use Max-flow algorithms like Edmonds-Karp, and Ford-Fulkerson, and we can also use some machine learning algorithms like The k-means algorithm and EM algorithm for the Gaussian mixture model(GMM). In this part, we will introduce two kinds of Machine learning algorithms and one max-flow algorithm.

3.1 K-means algorithm

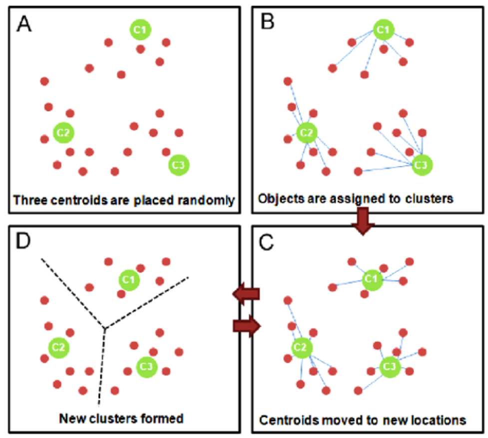


Figure 2. schematic representation of algorithm steps

K-means is a commonly used clustering algorithm based on Euclidean distance, which assumes that the closer two objects are, the more similar they are. In our case, we will select two points manually for the foreground and background. This means we will rely on the distance of each pixel point to the current centroid to determine which pixels belong to the foreground and which pixels belong to the background.

The operating order of the K-means algorithm is shown in Figure 2. We can summarize the K-means algorithm into the following steps using this flowchart:

1. Select k samples as the initial cluster centers (usually randomly selected k samples).

2. For each sample in the dataset, calculate its distance to the k cluster centers and assign it to the class corresponding to the cluster center with the minimum distance.

3. For each class, recalculate its cluster center, i.e., the centroid of all samples belonging to that class (average of each feature).

4. Steps 2 and 3 are repeated until some termination condition is met. The usual termination condition is when the maximum number of iterations is reached or when the clustering centers no longer change significantly. In each iteration, each sample is reassigned to a new category, so the final clustering results may vary.

The K-means algorithm not only has simple implementation steps and low complexity but also performs well in clustering scenarios. However, the algorithm also has obvious limitations, such as it is very sensitive to how the clustering centers are initialized, and different initialization methods may yield different results; because K-means uses a hard judgment method, it is not possible to assign samples to more than one category, and thus is not suitable for multiple classification tasks; it is not suitable for over-discrete classification, sample category imbalance, and non-convex classification.

Considering the many limitations of the K-means algorithm, we introduce an alternative probability-based soft-judgment algorithm, which will be discussed in the next section.

3.2 Gaussian Mixture Model & EM algorithm

In this section, we will introduce Gaussian mixture models and analyze how it works to solve image segmentation problems.

The Gaussian mixture model (GMM) is a linear combination of multiple Gaussian distribution functions. Theoretically, GMMs can be adapted to any type of distribution and are commonly used to solve situations where data in the same set contains several different distributions. This method utilized a Gaussian distribution as a model and is trained using the Expectation-Maximization (EM) algorithm.

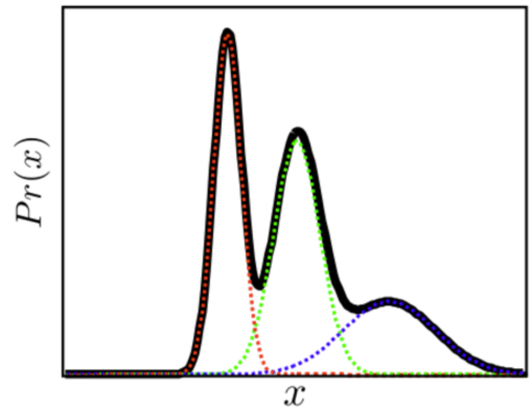


Figure 3. GMM representation graph

In our project, we will use the grayscale range of each pixel as the X-axis and normalize the number of pixels for each grayscale value to obtain the probability density function (PDF) value as the Y-axis. This will be used to build the model.

So it is easy to see that GMM is a probability-based algorithm, while K-means is a distance-based algorithm. Due to the difference in classification criteria, the shape of the classification region in GMM will not be as restricted as in K-means.

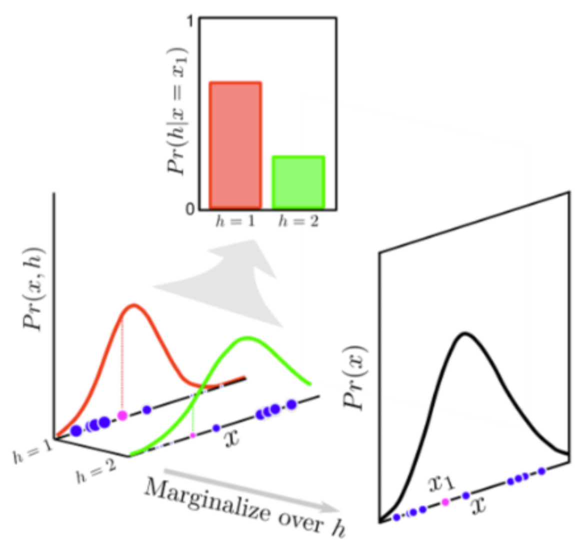


Figure 4. Principles of EM algorithm

Again, after building the model, we need to use the algorithm to update our model to achieve the best classification results. Here, we will use the EM algorithm to iterate until convergence. The EM algorithm can be decomposed into two parts.

1. E-step: For each pixel point, calculate the probability generated by each component in the mixture model. (Each component represents an independent Gaussian distribution).
2. M-step: Adjust the model parameters to maximize the probability of the model generating these parameters.

According to the above steps, the core of the EM algorithm is to determine for each pixel which class it is most likely to be generated by, based on the previously obtained probability density function. Then, by continuously adjusting the parameters of the Gaussian distribution, we find the distribution that maximizes the likelihood. Once this is done, we can easily segment the image's foreground and background.

3.3 Click interaction

As the very first step of our implementation, we built an user interface to take in the input image file and select the source and sink node. Providing an interface enables the users to select any images they want to do the segmentation. Also, the interface can show the coordinates of the mouse cursor. The reason why we let clients manually select the source node and sink node is that this will greatly increase the efficiency and accuracy of our algorithm. Choice of source node and sink node has a huge effect on the success of the segmentation. They have to be one from the foreground and another from the background. Say that two nodes are both from the foreground, they will have similar RGB value, and the flow between these two nodes will be quite big. According to the max-flow algorithm, it will be hard to separate them into two sets, thus the segmentation will fail.

Another point is that the selection of nodes can be done by program automatically. We can take a 10*10 pixel matrix from the image. With the 100 sample pixels, we pick the 2 nodes with the smallest and greatest RGB value and set them as source and sink nodes. But this will slow down the algorithm and also has the possibility of failing. So we would like to leave the job of selecting nodes to users.

3.4 Graph cut algorithm

The graph cut algorithm for image segmentation consists of two main parts, the first part is to construct a graph based on color similarity and the rate of change of gray values, and the second part is to use the min cut algorithm to obtain a minimum cut, which is also the boundary line separating the object from the background.

We begin with the construction of the first part of the graph. The final aim of the graph cut is to obtain a cut that divides the image into independent regions that are not connected to each other, and to make the remaining (uncut) weights maximized. This means, in a more concrete sense, that the internal colors of the cut are closest to each other and do not change much. The energy function is expressed as follows:

$$\begin{aligned} |C| &= \sum_{p \notin \mathcal{O} \cup \mathcal{B}} \lambda \cdot R_p(A_p(C)) \; + \\ & \qquad \sum_{\{p,q\} \in \mathcal{N}} B_{\{p,q\}} \cdot \delta(A_p(C), A_q(C)) \\ &= E(A(C)) - \sum_{p \in \mathcal{O}} \lambda R_p(\text{“obj”}) - \sum_{p \in \mathcal{B}} \lambda R_p(\text{“bkg”}). \end{aligned}$$

The energy function is divided into two components, the first of which is the similarity weight between all selected and unselected points and is expressed as follows:

$$\begin{aligned} R_p(\text{“obj”}) &= -\ln \Pr(I_p|\mathcal{O}) \\ R_p(\text{“bkg”}) &= -\ln \Pr(I_p|\mathcal{B}). \end{aligned}$$

In this paper probabilistic similarity is calculated using the GMM algorithm. O denotes the pixel points that are artificially labeled as object and B denotes the pixel points that are considered to be labeled as background.

The other component of the energy function represents the difference in gray value between all pixel points and their neighbors and is calculated as follows:

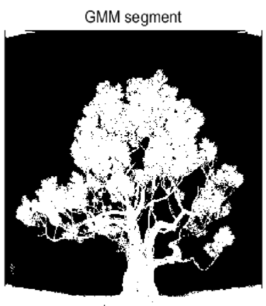
$$B_{\{p,q\}} \propto exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}.$$

The group of weights takes into account the differences in color and distance between pixel points. In the energy function, the importance attached to the two sets of parameters is balanced by introducing a parameter λ.

4. Results

Comparison between GMM and graph cut

We evaluate our solution by putting the same graph into both GMM and graph-cut algorithms and compare the segmentation result. The two output images are shown below.



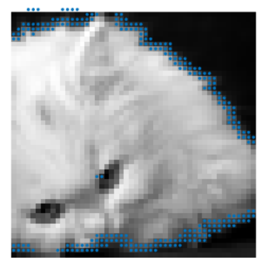
GMM result

graph cut result

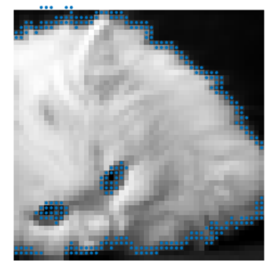
We can see that GMM has a more accurate edge around the main foreground(tree), but it accidentally attributes the top-left most region and top-right most region to the foreground. In comparison, Graph-cut does have a relatively more blurred border, but it correctly distinguishes the target object. The reason for this discrepancy is that GMM counts the weight difference (RGB value in this case) of disconnected nodes as a key factor in segmentation. With similar weights, nodes will have greater possibility of being classified as the same cluster. After going through E-M iteration, these nodes will finally be set as foreground. However, in the case of Graph-cut, we only take the weight difference of neighbor nodes into consideration. Since the top-left region and top-right region are separated from the center region by pixels from the background, it is unlikely that the max-flow algorithm will classify these regions the same as the center.

Does this mean that Graph-cut is a better solution for image segmentation? Actually the choice between GMM and Graph-cut depends on the specific task and the characteristics of the data. If the data has well-defined boundaries between different clusters and the goal is to segment the image into distinct regions, then graph cut may be the better choice. On the other hand, if the data has complex and overlapping clusters, then GMM may be a better choice for clustering and density estimation.

Graph cut using different parameter



λ = 0.05



λ = 3

By adjusting the parameter λ, we have chosen in the first image to focus more on grouping similar pixels in a larger area, while in the second image we have chosen to think more in terms of the color of the pixel points themselves. Thus the first image classifies the eye area of the cat as an object, while the second image classifies the cat's eyes as a background.

5. Preliminary Code

Include a link to your project's github repository containing your current code and brief README. It should be publicly accessible.

Github: https://github.com/Chenxy517/EC504_ImageSegmentation

References

[1] Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2, 246–252.
[2] Rother, C., Kolmogorov, V., & Blake, A. (2004). "GrabCut": Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics, 23(3), 309-314.
[3] Zhang, D., & Chen, S. (2004). A novel kernelized fuzzy C-means algorithm with application in medical image segmentation. Artificial Intelligence in Medicine, 32(1), 37–50.
[4] Boykov, Y., & Jolly, M. P. (2001). Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. Proceedings of the IEEE International Conference on Computer Vision, 1, 105–112.

[5] Kolmogorov, V., & Zabih, R. (2004). What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(2), 147-159.
[6] Ishikawa, H. (2003). Exact optimization for Markov random fields with convex priors. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(10), 1333–1336.
[7] Kohli, P., & Torr, P. H. S. (2007). Robust higher-order potentials for enforcing label consistency. International Journal of Computer Vision, 82(3), 302–324.
[8] Li, M., & Zhang, K. (2010). A comparative study of GMM and graph cuts for image segmentation. International Journal of Innovative Computing, Information and Control, 6(3), 1111-1120.
[9] Wang, J., Li, J., & Chan, K. L. (2011). A comparative study of Gaussian Mixture Model and Graph Cut method for object extraction from complex background. In Proceedings of the International Conference on Digital Image Computing Techniques and Applications (DICTA), 148-153.
[10] Zhu, M., Chen, W., & Hao, Y. (2015). Image segmentation based on a hybrid region and edge weighting approach. Information Sciences, 316, 136-150.
[11] Zhang, Y., Wu, L., & Wang, S. (2017). A new image segmentation method based on Gaussian Mixture Model and Graph Cuts. In Proceedings of the International Conference on Computer and Information Technology (CIT), 5-9.