Comparative Study of Medical Image Segmentation Algorithms

Atakan Topcu, *Bilkent University* Ege Kor, *Bilkent University*

***Abstract*—With the advance of deep learning, many segmentation tasks are done solely by training a network for a specific dataset. Nonetheless, such methods are hindered by their training datasets distribution. Furthermore, most deep learning models must be tailored specifically for each task. Classical works, on the other hand, are generally not hindered by such problems. Here, we implement four classic segmentation tasks: K-means clustering, Mean-shift clustering, GrabCut, and Conditional Random Field (CRF).**

***Index Terms*—Computer Vision, Segmentation, Medical Imaging, Graph-Cut, Random Fields, Clustering**

I. Introduction

IMAGE segmentation is crucial in clinics where the segmentation of a medical image provides useful information, such as the 3D structure of a tissue or automation of targeting malignant tissues such as tumors. Its popularity among clinics is even more relevant after the rise of deep learning techniques. Nonetheless, classic methods are still prevalent and can be utilized when deep learning models fail. The classical models are especially more powerful when there is not enough training data or the training dataset has a specific distribution which can lead to bias in the model. Furthermore, some models have to be tailored specifically for a particular task. Thus, this limits their usability in clinical settings. This project implements K-means clustering, Mean-shift clustering, GrabCut, and Conditional Random Field (CRF) algorithms are implemented for tumor segmentation tasks. To see their success in the segmentation task, Accuracy, Precision, Recall and F1 scores are calculated and comparatively evaluated. Further information is in the methods section.

II. Methods

Here, methods used for K-means clustering, Mean-shift clustering, GrabCut, and CRF are further discussed.

A. K-means Clustering

Each pixel in the image has a feature that helps us to detect those points. They can be pixel position, the pixel value if it is a gray image. The pixel value has 3 red, green, and blue features if it is an RGB image. These features are extracted from the image using simple algorithms and some built-in functions; edges, gradients, and gradient direction information are added to the feature space. We prepared a map from a gray image to a 7-dimensional feature space using those features [1]. In K-mean segmentation, the number of clusters is given as input with the image. In our algorithm, we determine initial cluster means by using two different initializations, uniform and random. Then, using the L1 norm, determine the new cluster mean through some iterations. When the change of meaning is under the threshold, the iterations finish, and the last means is determined as the cluster's mean. The advantage of this algorithm is that the segmentation is so fast that the duration is not exceeding about 1 second. The drawback of the method is that the cluster number must be given. So, the results can be under or over-segmented.

B. Mean-shift Clustering

In Mean Shift clustering, the number of clusters is not given as input. Each pixel in the image is assigned to the nearer peak (mode) point. The peak or mode point is determined based on the density map of feature space of the image. The number of peak points determines the number of clusters. In this method, the bandwidth is given as input for the threshold value to separate the peak points. So, mean shift algorithm is sensitive to bandwidth and the duration of the method is longer compared to k - means algorithm. However, some preprocessing steps can decrease the duration. All in all, the advantage of the method is that the cluster number is not given as input. The drawback of the method is the sensitivity to bandwidth choice and long duration [1]. To solve the long duration problem, we implemented 2 preprocessing steps. At the end, we just used the 30% of that data. In other words, we removed the unnecessary background information and transformed the image to 1D vector. The implementation of the main idea is inspired by an open-source[1].

C. GrabCut

GrabCut algorithm was designed by Carsten Rother, Vladimir Kolmogorov & Andrew Blake from Microsoft Research Cambridge, UK [2]. The whole pipeline has been implemented in Python, and the main process is given in Fig. 1.

First, segmenting an image means attributing a label to each pixel. In the case of GrabCut and other foreground extraction algorithms, each pixel is labeled as being in the foreground or the background of the image [3]. The user draws a rectangle around the region of interest and defines the region outside the rectangle as "Sure Background". Then, the Gaussian mixture

A. Topcu and E. Kor are with the Department of Electrical and Electronics Engineering, Bilkent University, Turkey,

[1] https://www.mathworks.com/matlabcentral/fileexchange/10161-mean-shift-clustering

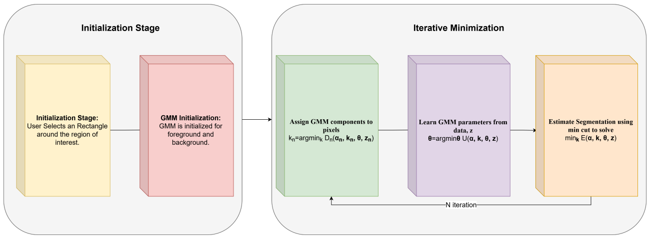
Initialization Stage
Initialization Stage: User Selects an Rectangle around the region of interest.
GMM Initialization: GMM is initialized for foreground and background.
Iterative Minimization
Assign GMM components to pixels
k_n=argmin$_k$ $D_n(\alpha_n, k_n, \theta, z_n)$
Learn GMM parameters from data, z
θ=argminθ $U(\alpha, k, \theta, z)$
Estimate Segmentation using min cut to solve
min$_k$ $E(\alpha, k, \theta, z)$
N iteration

Fig. 1. General Framework for GrabCut.

model (GMM) has been implemented for the background and foreground. Then, the parameters of GMM is being learned from finding the argument which minimizes the U Term shown in equation 1.

To obtain the segmentation, GrabCut takes advantage of the graph-like structure of an image. Each pixel has the following:

- One "n-link" to each of its 4 direct neighbors
- Two "t-link" to the source and sink nodes of the graph, representing respectively the image foreground and background.

The calculation of the minimal cost is done through "Gibbs" energy of the form:

$$\boldsymbol{E(\alpha, \theta, z)} = U(\boldsymbol{\alpha, \theta, z}) + V(\boldsymbol{\alpha, z}) \tag{1}$$

where the term U evaluates the fit of the opacity distribution α and V is the smoothness term. θ represents the parameters of GMMs, and z is the array of image pixels.

$$U(\boldsymbol{\alpha, \theta, z}) = \sum_n -log(h(z_n; \alpha_n)) \tag{2}$$

$h(z_n; \alpha_n)$ represents the histograms of grey values, $\theta = \{h(z_n; \alpha_n), \alpha = 0, 1\}$ one for background and one for foreground.

$$V(\boldsymbol{\alpha, z}) = 50 \sum_{(m,n) \in \boldsymbol{C}} \|\mathbf{m} - \mathbf{n}\|^{-1} [\alpha_n \neq \alpha_m] e^{(-\beta(z_m - z_n)^2)} \tag{3}$$

$\boldsymbol{C}$ represents the set of pairs of neighboring pixels. It has been shown that $\beta > 0$ [4]. From the original paper of [Boykov and Jolly 2001], β calculated by $\beta = \left(2\mathbb{E}\left[(z_m - z_n)^2\right]\right)^{-1}$ [4]. The implementation of the main idea is inspired by an open-source[2].

Once the iterative loop finishes, labels around the segmentation border are refined and classified as sure background, probable background, probable foreground, and sure foreground.

D. Conditional Random Field

Conditional random fields (CRFs) are undirected probabilistic graphical models. They are, by definition, discriminative, that is, they model a distribution $P(Y|X)$. Similar to the GrabCut algorithm, CRF can be considered in the domain of vertex and edges, $G = (V, E)$ in which each vertex $v \in \boldsymbol{V}$ is assigned to variable $Y_i \in \boldsymbol{Y}$. Each $Y_i \in \boldsymbol{Y}$ is independent of all its non-neighbors, given its neighbors and $\boldsymbol{X}$. The connectivity of G implies certain constraints on possible sets of cliques C_G. To get an actual distribution, a set of factors are required. Given the set of factors $\boldsymbol{\Phi}$ and the corresponding set of cliques C_G, we can write out the Gibbs distribution given in equation 4 [5].

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \frac{exp(-\sum_{c \in C_G} \phi_c(\boldsymbol{Y_c}|\boldsymbol{X}))}{Z(\boldsymbol{X})} \tag{4}$$

In equation 4, $\boldsymbol{X}$ represents the pixels of our image. The output $\boldsymbol{Y}$ is the segmentation of the image, and the edges in the graph represent the neighbors of a pixel. Generally, the more edges we add, the more complex and powerful the model will become. At the start of the project, fully-connected dense CRFs have been considered [6]. However, due to its complexity and the computation cost, we have decided to go with 8 neighborhood CRFs (Up-left, Up, Up-right, Left, Right, Down-left, Down, and Down-right). This allows for capturing long-distance relations between the pixels while also reducing the computation cost.

Next, we must define the cliques and factors defining the distribution. Similar to the GrabCut algorithm, factors that define the distribution can be considered in two parts, namely, unary potentials and pairwise potentials, $\phi_c = \phi_{c,unary} + \phi_{c,pairwise}$.

We assume that the unary potentials, ϕ_{unary}, are given by manual labeling. For the pairwise potentials, $\phi_{pairwise}$, we assume a sum of weighted Gaussians [6].

$$\phi_{pairwise,i,j}(Y_i, Y_j) = \mu(Y_i, Y_j) \sum_{k=1}^{K} w_k g_k(\boldsymbol{f_i}, \boldsymbol{f_j}) \tag{5}$$

Where w_k are weights, $\mu(Y_i, Y_j)$ is a compatibility function called Potts model, which models the relation between different labels by $\mu(Y_i, Y_j) = 1$ if i=j and 0 otherwise. Also, g_k are Gaussians defined by.

$$g_k(\boldsymbol{f_i}, \boldsymbol{f_j}) = exp(\frac{-1}{2}(\boldsymbol{f_i} - \boldsymbol{f_j})^T \boldsymbol{\Sigma_k}^{-1}(\boldsymbol{f_i} - \boldsymbol{f_j})) \tag{6}$$

$\boldsymbol{f_i}$ is a 5-dimensional feature vector of the ith pixel. It comprises the x and y coordinates of the ith pixel and its r,g, and b values. Since our images are in greyscale, $\boldsymbol{f_i}$ collapses to a 3-dimensional feature vector. In the end, we put these two potentials to the ϕ_c in equation 4. Based on this, we can state the actual image segmentation problem as MAP inference in this distribution. $Y^* = argmaxP(\boldsymbol{Y}|\boldsymbol{X})$. We use the approximate inference method called Mean Field Approximation to make the MAP inference feasible [5],[6]. The main process to solve $Y^* = argmaxP(\boldsymbol{Y}|\boldsymbol{X})$ using Mean Field Approximation is given in Fig. 2.

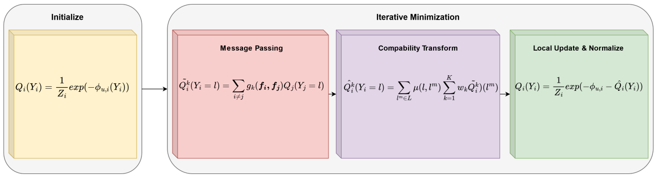
Initialize
$Q_i(Y_i) = \frac{1}{Z_i} exp(-\phi_{u,i}(Y_i))$
Iterative Minimization
Message Passing
$\tilde{Q}_i^k(Y_i = l) = \sum_{i \neq j} g_k(\boldsymbol{f_i}, \boldsymbol{f_j}) Q_j(Y_j = l)$
Compability Transform
$\hat{Q}_i^k(Y_i = l) = \sum_{l^m \in L} \mu(l, l^m) \sum_{k=1}^{K} w_k \tilde{Q}_i^k)(l^m)$
Local Update & Normalize
$Q_i(Y_i) = \frac{1}{Z_i} exp(-\phi_{u,i} - \hat{Q}_i(Y_i))$

Fig. 2. General Framework for CRF Mean Field Approximation.

Though this methodology is for Dense fully-connected CRF, we have applied it for our 8 neighborhood CRF [5]. For

[2] https://github.com/MoetaYuko/GrabCut

understanding the framework of Mean Field Approximation, the ”tutorial.pdf” given in a GitHub repository has been used [6].

III. Results

In this section, we represent both algorithms’ initial results. For testing, the dataset from Medical Segmentation Decathlon [7]. In literature, most researchers have focused on metrics for quantifying model accuracy. Thus, the following metrics are used [8].

$$Accuracy = \frac{TP+TN}{P+N}, \quad Precision = \frac{TP}{TP+FP}$$
$$F1 = \frac{2TP}{2TP+FN+FP}, \quad Recall = \frac{TP}{TP+FN} \tag{7}$$

Here, TP means True Positive, TN means True Negative, FP means False Positive, and FN means False Negative. Table I and Table II are given to evaluate these quantitative metrics.

A. K-means Clustering

After implementing the K-means algorithm, the model is tested on the MRI brain tumor dataset. The results can be seen in Fig. 3, Fig. 4, and 5. The right-most image for figures represents the Ground Truth.

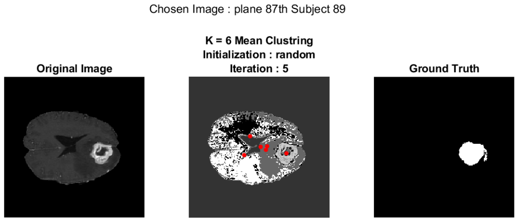

Fig. 3. K - Means Algorithm Implemented on subject 89, 87th slice.

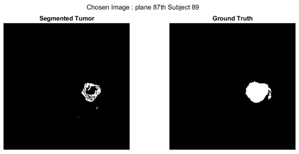

Fig. 4. K - Means Result Implemented on subject 89, 87th slice.

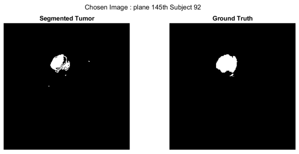

Fig. 5. K - Means Result Implemented on subject 92, 145th slice.

In this method, we created 5 dimensional feature space at first based on the pixel position, value, gradient, and edges. Then, we wrote k-mean algorithm and compared the results with the built-in function of k-means segmentation. The results mostly matched. As seen from the figures, we clustered the tumors from images successfully.

B. Mean-shift Clustering

After implementing the Mean-shift algorithm, the model is tested on the MRI brain tumor dataset. The results can be seen in Fig. 6, 7, and 8. The right-most image for figures represents the Ground Truth.

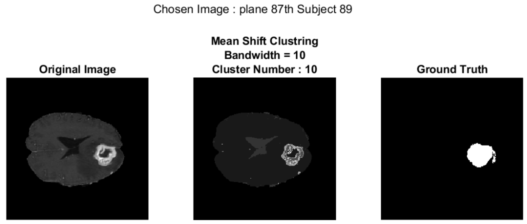

Fig. 6. Mean Shift Algorithm Implemented on subject 89, 87th slice.

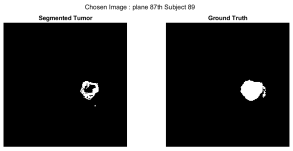

Fig. 7. Mean Shift Result Implemented on subject 89, 87th slice.

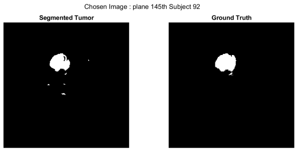

Fig. 8. Mean Shift Result Implemented on subject 92, 145th slice.

Although our expectations about mean-shift is that the duration is long, we concluded that after 2 preprocessing steps (specific for our dataset), the duration decreased under 2 seconds. As seen from the figures, the method successfully clustered the tumor regions. We also realized the sensitivity of the bandwidth so, we took bandwidth around 10-12 to get better results. When bandwidth is high, we observed that the cluster number is low and the image is smooth. When the bandwidth is low, the image was overclustered and looked like noisy. All in all, with appropriate bandwidth range, we clustered tumors correctly.

C. GrabCut

After implementing the GrabCut algorithm, the model is tested on the MRI brain tumor dataset. The results can be seen in Fig. 9 and Fig. 10 for qualitative examination. The right-most image for both figures represents the Ground Truth.

It can be seen in Fig. 9 that in situations with a more complex structure, GrabCut fails to segment the fine details. In Fig. 9, this corresponds to the inner region of the segmentation where GrabCut fails and contradicts the ground truth. In Fig. 10, this corresponds to the outer edges where there are small tumor regions. These might have resulted due to the fewer features available for grayscale images. However,

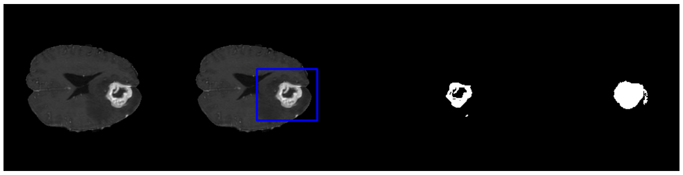

Fig. 9. GrabCut Algorithm Implemented on subject 89, 87th slice.

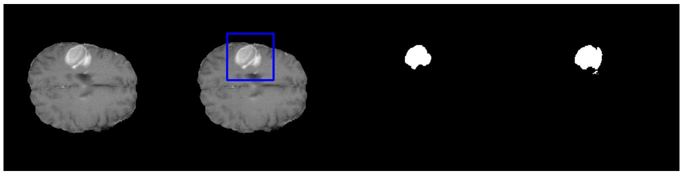

Fig. 10. GrabCut Algorithm Implemented on subject 92, 145th slice.

unlike CRF, which will be discussed next, GrabCut doesn’t need finely detailed reference segmentation. It only requires a rectangle to start segmentation. Thus, there is a trade-off between high accuracy and costly segmentation reference. Furthermore, speed-wise, GrabCut is significantly faster than CRF.

D. Conditional Random Field

After implementing the CRF algorithm, the model is tested on the MRI brain tumor dataset. The results can be seen in Fig. 11 and Fig. 12 for qualitative examination. The right-most image for both figures represents the Ground Truth. It can be seen that the algorithm is able to extract the tumourous site better than the GrabCut algorithm. However, it takes much more time to compute the segmentation, and CRF requires a much more detailed initial segmentation than the GrabCut method. Though, these limitations can be overcome using efficient message-passing algorithms for faster computation and using neural networks for forming initial segmentation [6].

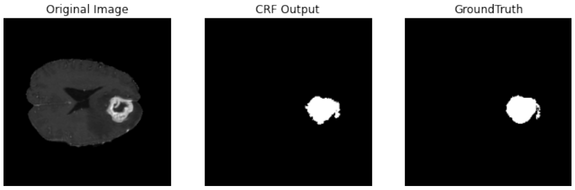

Fig. 11. CRF Algorithm Implemented on subject 89, 87th slice.

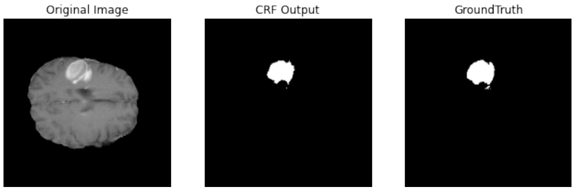

Fig. 12. CRF Algorithm Implemented on subject 92, 145th slice.

TABLE I
COMPARISON OF EACH METHOD FOR SUBJECT 89, 87TH SLICE

	Accuracy	Precision	Recall	F1
K-means Clustering	0.9851	0.6883	0.9868	0.8109
Mean-shift Clustering	0.9867	0.7222	0.9893	0.8349
GrabCut	0.9936	0.9808	0.7481	0.8488
Conditional Random Field	0.9964	0.9976	0.9987	0.9981

TABLE II
COMPARISON OF EACH METHOD FOR SUBJECT 92, 145TH SLICE

	Accuracy	Precision	Recall	F1
K-means Clustering	0.9939	0.8396	0.9929	0.9099
Mean-shift Clustering	0.9965	0.9262	0.9747	0.9498
GrabCut	0.9967	0.9847	0.8382	0.9056
Conditional Random Field	0.9972	0.9974	0.9998	0.9986

IV. DISCUSSION & CONCLUSION

It can be seen that GrabCut is able to keep up with CRF even though it is less complex and faster to compute. However, in some cases, it cannot capture fine details as much as the CRF algorithm. Nonetheless, considering its speed and optimal performance, it can be considered a better alternative to CRF. As for the K-means and Mean-shift clustering algorithms, even though they were able to perform relatively well, they are not performing as well as the GrabCut or CRF algorithms. One of the downsides of these clustering methods is that it is constrained by their cluster number. Furthermore, unlike CRF and GrabCut, clustering methods don’t have an inherent region of interest. This means even if similar features are apart, they can still be grouped together. In the end, CRF and GrabCut models seem to be the most optimal classical methods, while the clustering methods can achieve high performance while being relatively less complex.

REFERENCES

[1] S. K. Nayar, “Image segmentation.” [Online]. Available: https://www.youtube.com/playlist?list=PL2zRqk16wsdop2EatuowXBX5C-r2FdyNt

[2] C. Rother, V. Kolmogorov, and A. Blake, “”grabcut”,” *ACM SIGGRAPH 2004 Papers on - SIGGRAPH ’04*, 2004.

[3] “Grabcut for automatic image segmentation [opencv tutorial].” [Online]. Available: https://www.sicara.fr/blog-technique/grabcut-for-automatic-image-segmentation-opencv-tutorial

[4] Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images,” *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001.*

[5] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected crfs with gaussian edge potentials,” 12 2011.

[6] L. Bruns, “Fast and accurate image segmentation using fully connected conditional random fields,” https://github.com/roym899/crf_segmentation_tutorial, 2021.

[7] A. L. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. van Ginneken, A. Kopp-Schneider, B. A. Landman, G. Litjens, B. Menze, O. Ronneberger, R. M. Summers, P. Bilic, P. F. Christ, R. K. G. Do, M. Gollub, J. Golia-Pernicka, S. H. Heckers, W. R. Jarnagin, M. K. McHugo, S. Napel, E. Vorontsov, L. Maier-Hein, and M. J. Cardoso, “A large annotated medical image dataset for the development and evaluation of segmentation algorithms,” 2019.

[8] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image segmentation using deep learning: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–1, 2021.