Lung Segmentation Pipeline for CT Images

Leo Ramos
School of Mathematical and Computational Sciences,
Yachay Scientific Computing Group
Yachay Tech University
Urcuquí, Ecuador
leo.ramos@yachaytech.edu.ec

Israel Pineda
College of Science and Engineering
Universidad San Francisco de Quito
Quito, Ecuador
ipineda@usfq.edu.ec

2022 IEEE Sixth Ecuador Technical Chapters Meeting (ETCM) | 978-1-6654-8744-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ETCM56276.2022.9935736

***Abstract*—Segmentation is one of the fundamental tasks in biomedical image processing. Adequate image segmentation and computer-aided diagnosis systems are excellent allies for healthcare professionals. There are multiple methods for image segmentation using image processing techniques that are still being used and developed. These have advantages over machine learning models and deliver reliable and fast results as training data for their operation do not limit them. This work proposes a 3-step semi-automatic pipeline for lung computed tomography image segmentation. It starts with preprocessing, in which the input image is enhanced; then, the image is segmented using the region growing technique, and finally, the segmentation mask is enhanced by applying a hole-filling process. The experimental results of the pipeline provided a Dice Coefficient of 0.9633 and an Intersection over Union of 0.9341 on average.**

***Keywords*—image segmentation, region growing, medical imaging, image pipeline.**

I. Introduction

An image is a visual representation of a real or imaginary object and a valuable tool for storing [1] and transferring information. Because of this, techniques are necessary to process them. These days, image processing techniques are in high demand in various areas such as medical imaging [2], remote sensing, pattern recognition, multimedia computing, and more [3], [4]. For all these applications, techniques are essential to automate the analysis and interpretation of images [3].

Often, not all parts of an image are helpful or necessary, but some specific area with some specific feature; this is why, in many processes, segmenting the image is the first step [1]. Image segmentation is one of the fundamental tasks in digital image analysis [5] and a critical component in many complex processes [6]. Segmentation consists of partitioning an image into disjoint regions so that each of them corresponds to a specific object [3], [7]. Each region has similar characteristics or properties that allow them to be identified and differentiated from others [5]. Once identified, regions can be analyzed, measured, or classified [8].

Image segmentation applications vary, including object detection, image classification [9], self-driving cars, and medical imaging. In the latter field, it is an important tool. Image segmentation is essential in the medical imaging process since it extracts a region of interest (ROI) [10] to analyze structures, tissues, organs, image measurements [2], and detect pathologies [1], [11]. In this way, it is possible to extract more and better information and have better arguments for diagnosis, and medical treatment [11]. Some specific applications include pixel classification of an image into anatomical regions, such as bones and muscles [2], tumor detection and volumetry, such as cancer, tissue deformities, surgical planning, classification of blood cells, and more.

Segmentation is a crucial task in the field of medical image processing. Sometimes, segmentation can be complex since the process can be challenged by numerous aspects such as poor image quality, noise, and occlusion of the figure to be segmented, among others [2]. Poor segmentation can lead to various problems, from incorrect measurement to misdiagnosis. No single technique works perfectly in all scenarios; most of the time, one technique works better than another depending on the context of the problem. However, an algorithm or technique must provide the highest level of accuracy possible in its field of action.

In this work, we propose a pipeline for the segmentation of lung Computed Tomography (CT) images, which uses input enhancement processes before being segmented by the region growing technique. Then, the segmented image is processed one last time before being delivered as the final result. Our results show high accuracy values in the experiments; we discuss and analyze them.

II. Related Work

Image processing methods are still being studied and developed despite being considered classical. They perform correctly and have some advantages over current deep learning methods [12].

Xuan *et al.* [13] presented a method that combines region growing and edge detection for magnetic resonance imaging (MR) brain image segmentation. They started from an image preprocessing in which edge detection is applied using the Canny edge detector and edge-preserving smoothing implemented by an adaptive smoothing filter. Subsequently, the region growing algorithm was applied followed by a region merging method. Next, false edges produced by the segmentation are removed. Their results were visually compelling and demonstrated that integrating the region growing method and edge processing produces reliable results.

978-1-6654-8744-3/22/$31.0 ©2022 IEEE

A few years later, in 2006, Ng *et al.* [14] proposed a methodology incorporating k-means and an improved watershed segmentation algorithm for medical image segmentation. Given some limitations of the watershed segmentation method, such as sensitivity to false edges, the idea of the authors was to use the k-means method to perform an initial image segmentation before applying the main segmentation algorithm. After applying k-means to the image, the Sobel filter was applied for edge detection, followed by an automatic thresholding process to create a binary mask. After this, the watershed segmentation method is applied, and then a merging process to obtain the resulting image. They applied it to 50 magnetic resonance (MR) head images to test their method. As a result, they found that their methodology produces segmentation maps with 92% fewer partitions than the classical watershed segmentation method.

In 2010 Zhan and Cheng [15] proposed a segmentation approach that combines the watershed algorithm with graph theory. The method starts by calculating the gradient of the image and then reconstructing it using a mixed opening and closing morphological reconstruction operation. Then an active floating-point image is introduced as the reference image of the watershed transformation. Finally, an algorithm based on graph theory, Grab Cut, is used for fine segmentation. CT images of human pelvic cavity bone and human finger were used for testing. The results showed that the proposed method excludes the false contours of over-segmentation and improves segmentation quality while maintaining all possible image information.

In 2011, Mesanovic *et al.* [16] proposed a lung segmentation technique to segment lung parenchyma from CT images of the lung accurately. The method consists of six stages. First, histogram thresholding is used to find the binary mask of the image. Then, the region growing algorithm is applied classically. Subsequently, edge detection using the Sobel filter is applied to create a mask for lung extraction. Then, morphological filling processes are applied. This image is multiplied by the original image to obtain the result. Lastly, the lung region is extracted. They used 50 CT images for testing and found adequate results for this first dataset. However, they then tested using a new dataset, and the method was not entirely accurate. The latter occurred in images where the lesion is touching the pleura.

III. Segmentation Methods

There are several methods of image segmentation [5], [17]. These can be classified into three general groups: edge-based, region-based, and clustering-based segmentation algorithms [5], [9], [17]. This paper focuses on region-based segmentation, specifically, the region growing technique.

A. Edge-based segmentation

In this category, images are divided according to discontinuities or edges [4]. Edges generally correspond to those points in an image where the gray level changes drastically [5]. These changes or discontinuities usually occur at object boundaries. Therefore, pixels that are part of edges can be detected and used to delimit objects [5].

Edge detection techniques are usually classified as sequential and parallel [7]. In the sequential method, whether a point or set of points belongs to the edge depends on previously examined points; on the other hand, in the parallel method, the decision does not depend on whether other previous points are on an edge [9], [11]. Some edge-based techniques are the Sobel operator, Laplace operator, and Canny edge detector.

B. Region-based segmentation

The region-based method partitions an image by looking for similarities, such as pixel intensity value, gray level, color, between adjacent pixels and grouping them into unique regions [5]. Each region is assigned a unique label for representing the output image [1], [11], [17]. Some techniques of this group are region growing, region merging, and splitting and merging techniques [8].

Region Growing Technique is an iterative procedure that consists of grouping pixels or subregions into larger regions [3], [9] based on predefined criteria for growth. Each region starts from seed points, and these are compared with their four or eight neighboring pixels [18]. Those with similar properties [3], [8] are assigned to the region and become the new seed pixels. Gray level intensity is usually the most common parameter for comparing pixel similarity and is applied through a thresholding process [11]. A gray-level thresholding operation can be described as

$$S(x,y) = \begin{cases} R & \text{if } I(x,y) \leq \mathrm{T} \\ B & \text{if } I(x,y) > \mathrm{T} \end{cases} \tag{1}$$

where $I(x,y)$ is the original image, T is the tolerance or threshold, $S(x,y)$ is the segmented image, R is the region of interest, and B corresponds to the background. Then, pixels at or above the tolerance are assigned to the region. All pixels below the tolerance are assigned to the background.

The selection of one or more seeds can often be based on the nature of the problem and the type of image data available [6]. The comparison process is repeated iteratively, and the whole procedure ends when all pixels have been processed, obtaining defined and delimited regions [3]. Fig. 1 describes how this technique works.

C. Clustering-based segmentation

Clustering is a widely used technique for image segmentation [19]. It is an unsupervised machine learning technique that consists of grouping data [19], [20]. For this, a similarity criterion is defined between pixels, and then similar pixels are grouped to form clusters [19], [21]. This type of technique is widely used for color image segmentation [21], [22]. One of the best-known algorithms of this group of techniques is K-means clustering [22].

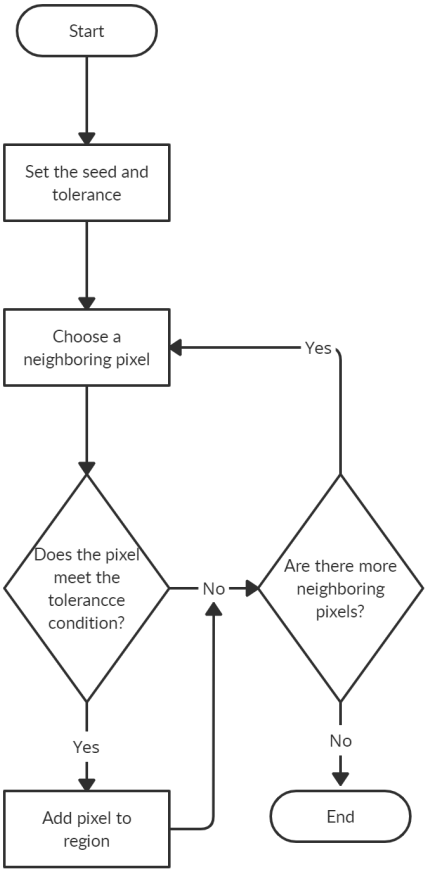


Fig. 1: Schematic illustration of region growing technique

IV. Methodology

A. Dataset description

The "Finding and Measuring Lungs in CT Data"[1] is the dataset used for this work. This dataset is uploaded and available for free on Kaggle. The dataset focuses on segmentation structures, specifically lungs, for subsequent measurement.

The dataset contains 2D and 3D CT images of the lung along with their respective manually segmented masks. In addition, it has a spreadsheet in which there are measurements performed on the 2D images. There are 267 2D images and four 3D images. 2D images are available in TIF format and 3D images are available in NII format. The whole dataset is 663 MB.

B. Proposed pipeline

Once the image is in grayscale, we continue with our three-step procedure.

1) Input preprocessing: First, a histogram equalization process is performed on the input to improve its contrast. Then, a median filter is applied to smooth the image and eliminate noise. This filter is applied because noise can often affect and even prevent the correct segmentation of an image. Finally, edge detection reduces the possibility of considering pixels that do not correspond to the region of interest.

2) Segmentation: The segmentation method executes after the input preprocessing. This pipeline uses the region growing technique, which uses a set of seeds to segment an object in an image. The technique analyzes neighboring pixels and adds those that meet a tolerance condition to the region. The process ends when all pixels have been analyzed and results in the segmented object.

3) Output enhancement: Once the image has been segmented, a final process is performed to improve it. We used the morphological operation of closing. The morphological closing performs a dilation process followed by an erosion process that fills in any possible holes found in the image. The image resulting from this process is the final result.

Fig. 2 shows this pipeline of our proposal.

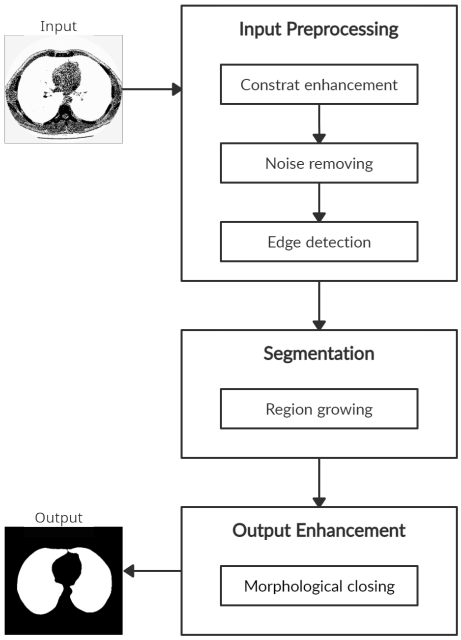


Fig. 2: Proposed pipeline for lung segmentation using CT images

C. Evaluation metrics

1) Dice coefficient: The dice coefficient (DC), also called dice score, is a statistic to compare the similarity of two samples [23], [24]. This metric is widely used in medical imaging for the validation of image segmentation and evaluates the usefulness of this to highlight isolated abnormalities such as tumors [25].

For the calculation of the DC, we use the true segmentation mask, also called ground-truth, and the image resulting from our segmentation process. Then, we take the number of pixels at the intersection of the two images, multiply this value by two and divide it for the total pixels in both images.

Equation 2 shows the definition of DC.

[1]https://www.kaggle.com/kmader/finding-lungs-in-ct-data

TABLE I: Hardware and software specifications

Device	Acer Nitro NP515-51 Spin
CPU	Intel Core i7-8550U 1.80GHz
GPU	NVIDIA GeForce GTX 1050 4 GB
RAM	8 Gb DDR3
Storage	500 GB SSD
OS	Windows 11 Home
Programming language	Python 3.8.2
Libraries	NumPy, OpenCV, Matplotlib
IDE	Visual Studio Code

$$DC = \frac{2|A \cap B|}{|A| + |B|} \tag{2}$$

where A is the ground-truth, and B is the image obtained by some segmentation technique.

The DC ranges from 0 to 1; the closer the DC value is to 1, the better the segmentation achieved [23]–[25]. A DC of 0.7 is considered an excellent match between the segmentation result and the expected or ideal segmentation [23].

2) Intersection over union: Intersection over Union (IoU) is a metric to quantify the percentage of overlap between the ground-truth and the prediction result [26], [27]. Mathematically, it is defined as the intersection of the ground-truth and the predicted segmentation divided by their union [24], [26], [27], as shown in Equation 3.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

where A is the ground-truth and B is the prediction.

Like DC, IoU also ranges from 0 to 1, with 0 meaning no overlap and 1 meaning perfect overlap [27].

D. Implementation details

1) Implementation environment: The pipeline was implemented in a desktop environment using Python as the programming language and the libraries Numpy, OpenCV and Matplotlib. All hardware and software specifications can be seen in Table I.

2) Code features: The pipeline follows a modular structure; i.e., the entire program is made up of or divided into subprograms or modules. The nature of the dataset suggests that what is expected is a binary mask; therefore, the program receives as input the CT image and returns a binary image in which the areas of interest are white while everything else is black. The image to be segmented is read in grayscale.

a) Input preprocessing: This component comprises the histogram equalization, the median filter, and the edge detection; each was programmed separately to be used as modules. It is worth mentioning that the histogram equalization and the median filter functions were programmed following the classical paradigms, while edge detection was performed using the well-known thresholding procedure. The median filter uses a 3×3 kernel.

b) Segmentation: This component comprises the region growing technique as such. It takes as input the image after being preprocessed, the tolerance, and the seed or group of seeds that the user must previously set. The seed should be placed within the region the user expects to segment. The function is iterative since it goes through the pixels of the image matrix to compare them with their neighbors. The comparison is performed using a 8-connected neighborhood; each pixel is compared with its eight neighboring pixels.

c) Output enhancement: This component is the final part of the pipeline. It comprises the morphological closure operation, which is performed using a function provided by OpenCV. This process uses a 3x3 kernel and four iterations.

d) Complementary functions and additional details: To make it easier for the user to set the seeds, a function was programmed that opens the image to be segmented and allows the user to click on the place where he/she wants to place the seeds. In addition, there is a function to visualize the results that shows the user a window with four images; from left to right: the original image, the original image with the seeds, the ground-truth, and the pipeline result. In addition, function automatically calculates and displays the DC, the IoU, and the execution time of the pipeline.

It should be mentioned that the user must set the path of the image to be segmented, and the tolerance must be set beforehand.

V. Results and Discussion

The experiments were performed by taking all 2D images from the dataset, i.e., 267 CT images. The task, as expected, consists of segmenting the lungs from the CT images, resulting in a binary mask. The seeds were placed manually, making sure that they were in the area of interest, and the tolerance was established after a previous test that allowed choosing a fixed number for it.

Visually, the pipeline delivered better results on images whose border did not show significant alterations nor too much noise in the area of interest. The pipeline correctly segmented large regions, as shown in Fig. 3d, and also small regions, as in the case of Fig. 5d. It also had no problems with images with rounded or sharp areas, as shown in Fig. 4d.

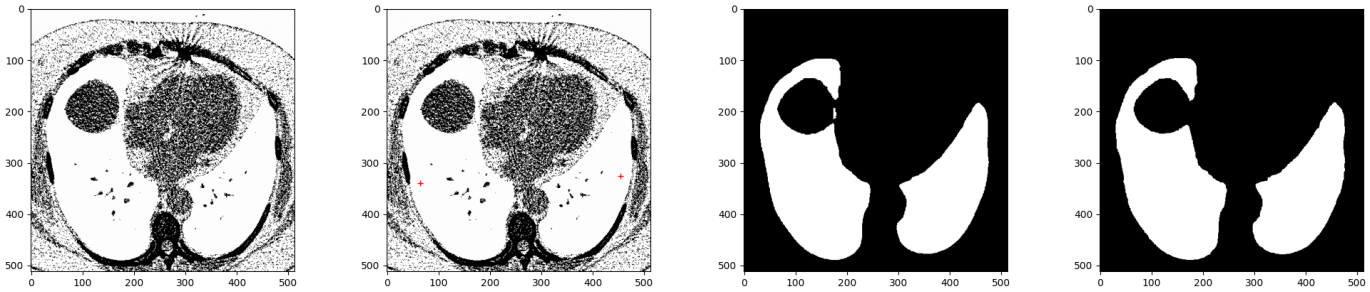
The lowest quality results occurred in images that presented high perturbations at the edges of the regions of interest, as shown in Fig 6d. Excessive noise was also a limiting factor in some cases. Although most images were correctly smoothed and filled by the pipeline, areas with too much noise caused the resulting masks to have gaps.

Numerically, the mean value of DC was 0.9633, while for IoU, it was 0.9341. As for the execution time, none of the images took more than 2.0788 sec to be segmented; on average, the execution took 1.2921 sec.

Table II shows the above numerical results.

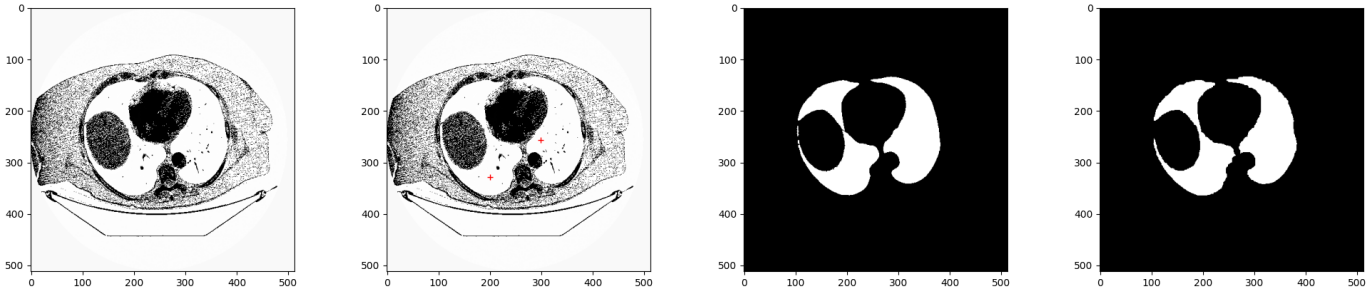
VI. Future Work

We want to explore different improvements for this pipeline as future work. First, this pipeline needs to be tested with other



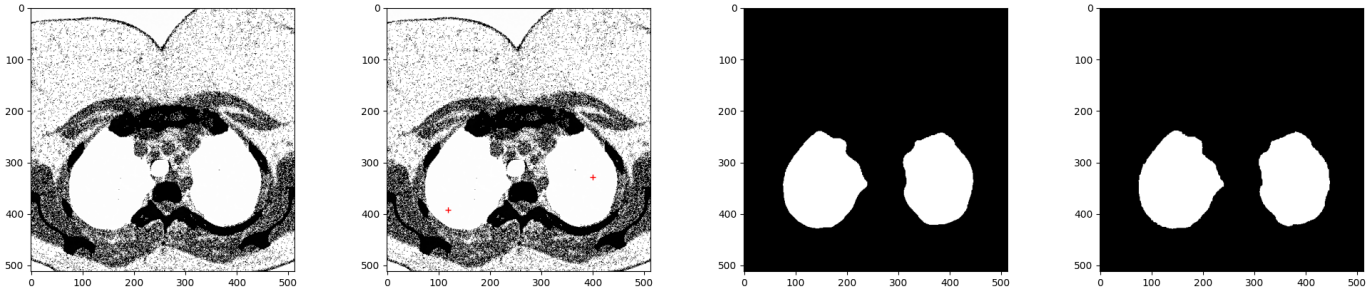
(a) Test Image (b) Seeded Image (c) Ground-Truth (d) Prediction

Fig. 3: Comparison of the prediction generated by the pipeline with respect to the reference element (ground-truth), large and concave lungs.



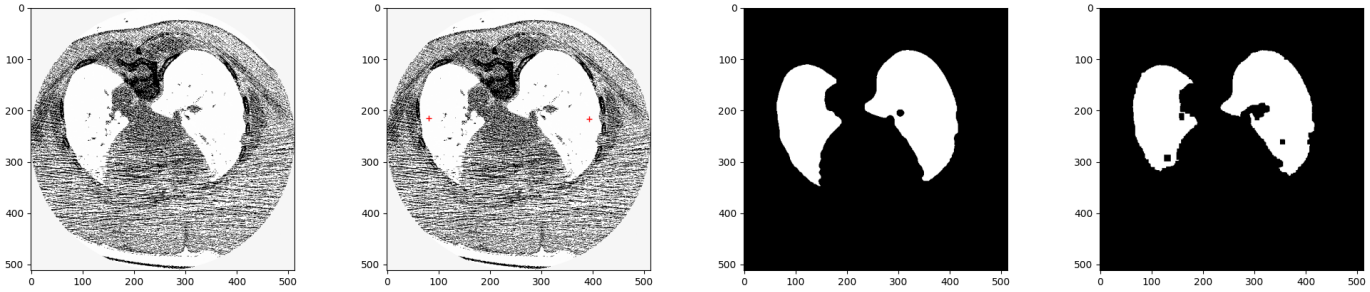
(a) Test Image (b) Seeded Image (c) Ground-Truth (d) Prediction

Fig. 4: Comparison of the prediction generated by the pipeline with respect to the reference element (ground-truth), concave and sharp lungs.



(a) Test Image (b) Seeded Image (c) Ground-Truth (d) Prediction

Fig. 5: Comparison of the prediction generated by the pipeline with respect to the reference element (ground-truth), small lungs.



(a) Test Image (b) Seeded Image (c) Ground-Truth (d) Prediction

Fig. 6: Comparison of the prediction generated by the pipeline with respect to the reference element (ground-truth), worst case.

TABLE II: Pipeline performance metrics results after 267 tests.

	DC	IoU	Runtime (sec)
Min.	0.7977	0.6262	0.8794
Max.	0.9932	0.9865	2.0788
Avg.	0.9633	0.9341	1.2921

datasets and organs. In this way, we could see whether or not the pipeline can be generalized to other contexts.

Also, it is possible to combine this pipeline with additional techniques to automate its operation. For example, apply machine learning or deep learning techniques to automatically set the seed and tolerance. Likewise, the program could be modified to segment color images using conditions that group pixels of the same color.

Additionally, it is possible to do a deeper analysis to find alternatives to improve some pipeline stages. Further reducing the noise and improving the sensitivity of the contour detection would be an alternative to improve the quality of the prediction. Interesting applications could be derived using our work as a pre-processing stage. For example, automatically labeling the image with a diagnosis could be achieved using techniques similar to the ones proposed by Castro *et al.* [28].

VII. Conclusions

This work proposes an image processing pipeline for lung segmentation using CT images. This pipeline uses the region growing technique for segmentation and enhances it with two additional processes. The first one is preprocessing, which increases contrast, reduces noise, and highlights the contours of the input image. The second is applied after segmentation and fills any remaining holes improving the final result.

The proposed pipeline takes advantage of image processing techniques to deliver an accurate mask in a timely manner. Our experiments show an average DC of 0.9633, an average IoU of 0.9341, and an average execution time of 1.2921 sec.

A current limitation of our proposal is that the tolerance and the seed must be set and adjusted manually. However, this is understandable since this is not a deep learning model but a pipeline using classical image processing techniques. However, this limitation is manageable, and we ought to explore better ways to fine-tune this number. The advantage of such a solution is that it is simple to implement in code and works deterministically using few computational resources.

References

[1] T. W. Ryan, "Image Segmentation Algorithms," in *Architectures and Algorithms for Digital Image Processing II*, vol. 0534. SPIE, jul 1985, p. 172.

[2] J. Rogowska, "Overview and Fundamentals of Medical Image Segmentation," in *Handbook of Medical Imaging*. Elsevier, 2000, pp. 69–85.

[3] T. Acharya and A. K. Ray, *Image processing: principles and applications*. John Wiley & Sons, 2005.

[4] A. Eleftheriadis and A. Jacquin, "Image and Video Segmentation," vol. 7, pp. 1–68, jan 1999.

[5] Q. Wu and K. Castleman, "Image segmentation," *Microscope image processing. Academic, Burlington*, pp. 159–194, 2008.

[6] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 4th ed. Pearson, 2018.

[7] H. J. He, C. Zheng, and D. W. Sun, "Image Segmentation Techniques," in *Computer Vision Technology for Food Quality Evaluation: Second Edition*. Academic Press, jan 2016, pp. 45–63.

[8] N. Ikonomakis, K. N. Plataniotis, M. Zervakis, and A. N. Venetsanopoulos, "Region growing and region merging image segmentation," in *International Conference on Digital Signal Processing, DSP*, vol. 1. IEEE, 1997, pp. 299–301.

[9] K. S. Fu and J. K. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, no. 1, pp. 3–16, jan 1981.

[10] J. Li, M. Erdt, F. Janoos, T.-c. Chang, and J. Egger, "Medical image segmentation in oral-maxillofacial surgery," in *Computer-Aided Oral and Maxillofacial Surgery*. Elsevier, 2021, pp. 1–27.

[11] S. Yuheng and Y. Hao, "Image Segmentation Algorithms Overview," jul 2017.

[12] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, "Deep Learning vs. Traditional Computer Vision," in *Advances in Intelligent Systems and Computing*, vol. 943, 2020, pp. 128–144.

[13] J. Xuan, T. Adali, and Y. Wang, "Segmentation of magnetic resonance brain image: integrating region growing and edge detection," in *IEEE International Conference on Image Processing*, vol. 3. IEEE, 1995, pp. 544–547.

[14] H. P. Ng, S. H. Ong, K. W. Foong, P. S. Goh, and W. L. Nowinski, "Medical image segmentation using k-means clustering and improved watershed algorithm," in *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, vol. 2006, 2006, pp. 61–65.

[15] Y. Zhang and X. Cheng, "Medical image segmentation based on watershed and graph theory," in *Proceedings - 2010 3rd International Congress on Image and Signal Processing, CISP 2010*, vol. 3, 2010, pp. 1419–1422.

[16] N. Mesanovic, M. Grgic, H. Huseinagic, M. Males, E. Skejić, and S. Muamer, "Automatic CT Image Segmentation of the Lungs with Region Growing Algorithm," Tech. Rep., 2011.

[17] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 1, pp. 100–132, jan 1985.

[18] M. Mancas, B. Gosselin, and B. Macq, "Segmentation using a region-growing thresholding," in *Image Processing: Algorithms and Systems IV*, vol. 5672. SPIE, mar 2005, p. 388.

[19] G. B. Coleman and H. C. Andrews, "Image Segmentation by Clustering," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 773–785, 1979.

[20] S. Chebbout and H. F. Merouani, "Comparative study of clustering based colour image segmentation techniques," in *8th International Conference on Signal Image Technology and Internet Based Systems, SITIS 2012r*, 2012, pp. 839–844.

[21] L. P. Maguluri, K. Rajapanthula, and P. N. Srinivasu, "A Comparative Analysis of Clustering based Segmentation Algorithms in Microarray Images," *International Journal of Emerging Science and Engineering (IJESE)*, no. 1, p. 27, 2013.

[22] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, feb 2003.

[23] A. P. Zijdenbos, B. M. Dawant, R. A. Margolin, and A. C. Palmer, "Morphometric Analysis of White Matter Lesions in MR Images: Method and Validation," *IEEE Transactions on Medical Imaging*, vol. 13, no. 4, pp. 716–724, 1994.

[24] J. Yang, Y. He, J. Caspersen, and T. Jones, "A discrepancy measure for segmentation evaluation from the perspective of object recognition," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 101, pp. 186–192, 2015.

[25] B. Guindon and Y. Zhang, "Application of the Dice Coefficient to Accuracy Assessment of Object-Based Image Classification," *Canadian Journal of Remote Sensing*, vol. 43, no. 1, pp. 48–61, jan 2017.

[26] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June. IEEE Computer Society, feb 2019, pp. 658–666.

[27] R. Padilla, S. L. Netto, and E. A. Da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," in *International Conference on Systems, Signals, and Image Processing*, vol. 2020-July. IEEE Computer Society, jul 2020, pp. 237–242.

[28] R. Castro, I. Pineda, W. Lim, and M. E. Morocho-Cayamcela, "Deep learning approaches based on transformer architectures for image captioning tasks," *IEEE Access*, vol. 10, pp. 33 679–33 694, 2022.