Off-the-shelf 3D Lung Segmentation in CT using Generalized Histogram Thresholding

Marcelo A. F. de Toledo[1], Marina F. S. Rebelo[1], José E. Krieger[2,] Marco A. Gutierrez[1],

[1]Laboratório de Informática Biomédica – InCor HCFM (USP)
[2]Laboratório de Genética e Cardiologia Molecular – InCor HCFM (USP)
Av. Dr. Enéas de Carvalho Aguiar 44, 05403-900, São Paulo – SP – Brasil

{marcelo.arruda, marina.rebelo, j.krieger}@hc.fm.usp.br,
marco.gutierrez@incor.usp.br

Abstract. *Computerized Tomography is very important for lung disease diagnostics, including computer assisted methods. Lung segmentation is usually a first step in further sophisticated methods of diagnosis. If in one hand, deep learning methods have state-of-the-art performance, they aren't as simple to apply compared to classical methods, sometimes requiring extra data and training. We designed a method specific for lung segmentation based on histogram thresholding. We observed that, in our proposed method, by changing from Otsu to the more recently developed GHT we got a significant improvement in segmentation, jumping from 77% to 91% average dice (from 90% to 95% median dice, respectively), approaching deep learning methods (UNet) results (94% average and 97% median dice). Even though our proposed method runs on CPU, it's still 2.6 times faster than UNet on GPU. Moreover, our proposed method is off-the-shelf, requiring no training or parameter calibration, being suitable as pre-processing for more sophisticated methods that aim specific diagnoses.*

1. Introduction

Computerized Tomography (CT) scan is a non-invasive method very often used in the diagnose of several diseases such as cancer (Rajagopalan & Babu, 2020), nodules detection (Chilakala, 2020), diffuse pulmonary and pleural effusion (Zhou et al., 2016) and many other lungs diseases. Beyond its historical importance, CT scans will be highly relevant in the follow up of COVID-19 cases for sequel monitoring (McCarthy, 2001; Pu et al., 2020; Sara Naybandi Atashi, Zeinab Naderpour, & Saeedi, 2021). For a great set of computer aided diagnostic (CAD) methods, this includes the localization or segmentation of the lungs in CT volumes (Gordaliza et al., 2018; Huidrom, Chanu, & Singh, 2018; Liu, Zhao, Xie, & Pang, 2020; Pu et al., 2020).

Nowadays, deep neural network methods represent the state of the art in such operations (Hofmanninger et al., 2020; Liu & Pang, 2020). However, their application is computationally intensive, requiring GPUs, narrowing its deployment viability in real clinical scenarios. Moreover, some deep learning methods have an unknown quality reproducibility, and sometimes drop in performance in generalizations to new datasets. These concerns are usually addressed by applying transfer learning or fine tuning. However, these procedures require the availability of extra data and specialists to label them, which is a sensitive demand for clinical images where labeled data is scarce. Moreover, they introduce an extra step, requiring technical computer knowledge to apply such methods. Considering that lung segmentation is rarely an end in itself, being often a preprocessing step for more sophisticated methods for diagnose (Huidrom et al., 2018), it is convenient to have off-the-shelf methods, that don't require extra data or computational steps, while also don't compromise the quality of the subsequent methods employed in diagnostic pipelines.

1.1 Histogram Thresholding Methods:

Among the classical segmentation methods, there is a plethora of which apply thresholding on voxels or pixels intensity, at least in some step. Even nowadays, many approaches rely on the well-established Otsu's method (Otsu, 1979), or variations of it (Akter, Moni, Islam, Quinn, & Kamal, 2020; Im et al., 2018; Kumar, Pant, & Ray, 2012; Than, Noor, Rijal, Yunus, & Kassim, 2014; Zhou et al., 2016). Even though there are other classical algorithms for threshold determination, such as Minimum Error (Kittler & Illingworth, 1986) and percentile based (p-Tile)(Doyle, 1962), they are not as commonly used or based on parameters setting, requiring adjustment for different datasets. Otsu's method works by calculating a threshold value for which the sum of the internal variance to each of the two classes is minimal. This will also correspond to maximal interclass variance.

Recently, a more general approach for histogram thresholding (Barron, 2020) has been developed, for which Otsu's, Minimum Error and P-Tile algorithm are special cases. In essence, it consists of approaching the histogram threshold calculation as a mixture of two gaussians in one dimension, one for each of the binary classes. As it doesn't rely on any other method for fitting the Gaussian mixture, such as Expectation-Maximization, it uses priors that have four parameters, set by the algorithm itself. See *appendix* for a more in depth explanation of Otsu's and GHT.

2. Objectives

We aimed to compare lung segmentation performance in CT scans between a classical approach using Otsu and the more recent GHT, and a deep learning approach (UNet). To meet this objective, we designed an off-the-shelf method for 3D lungs segmentation that doesn't require training or parameter tuning. It includes pre and post-processing steps, enabling it to deal with histograms distortions, clutter removal and inclusion of inner lungs blood vessels in the final mask. We compared this same approach using Otsu's and GHT to an UNet. We compared the three methods in segmentation performance, both in terms of Dice coefficient and processing time.

3. Materials and methods

3. 1 Data

For the experiments, we used the LCTSC dataset (Yang et al., 2017). This dataset has 60 CT exams and its respective segmentation masks, with right and left lungs, heart, esophagus and spinal cord manually annotated by specialists. Each exam has each of its axial slices saved as a Dicom image. The segmentation masks of each exam are all, both organ and slices, in one RTSTRUCT Dicom file.

3.2 Proposed method using GHT and Otsu

The proposed method, described below, involves steps of pre-processing, histogram thresholding and post processing, specifically designed for lung segmentation in CT scans (figure 2). For comparing Otsu and GHT all steps were kept the same, changing only the threshold calculation method. As the proposed methodology doesn't require learning or parameter adjustments, all the exams were used for evaluation. We analyzed each exam individually, and calculated the whole volume Dice coefficient (3D).

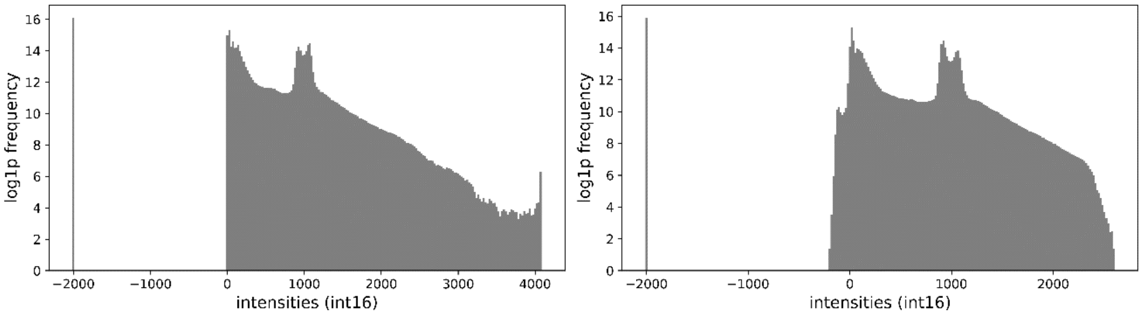

Figure 1. Log histograms of CT voxels intensities. Left: a standard case, background voxels marked as -2000, and the exam data starts at 0. Right: an exam whose histogram is distorted, starting before 0 and truncated before 3000. The proposed approach works on both cases.

Regarding CT histograms, in some cases, exams might have been submitted to histograms transformations and other unknown alterations, having elongated or truncated tails (figure 1, right). These are not obstacles for the proposed method's pre-processing. Also, exams use either the value –2000 or –1000 for marking voxels outside the CT chamber, making a background mask. This mask is used in post-processing step.

Pre-processing

1) Hounsfield scale: we subtracted 1000 from the volume intensities so they would be in the standard Hounsfield scale (from –1000 to +3000).

2) Hounsfield window: we applied a window centered in –400, with a total amplitude of 1000, which corresponds to lung tissue with an ample margin for handling more general cases than the dataset at hand (Zhou et al., 2016). Doing this, we kept only the values between –900 and +100.

Histogram calculation and thresholding

3) Histogram: we calculated a 256 bins histogram using only the voxels values selected by the window.

4) Threshold calculation: according to the GHT or Otsu's method, depending which one was being tested.

5) Thresholding: The calculated threshold was applied to the volume obtained after step 1.

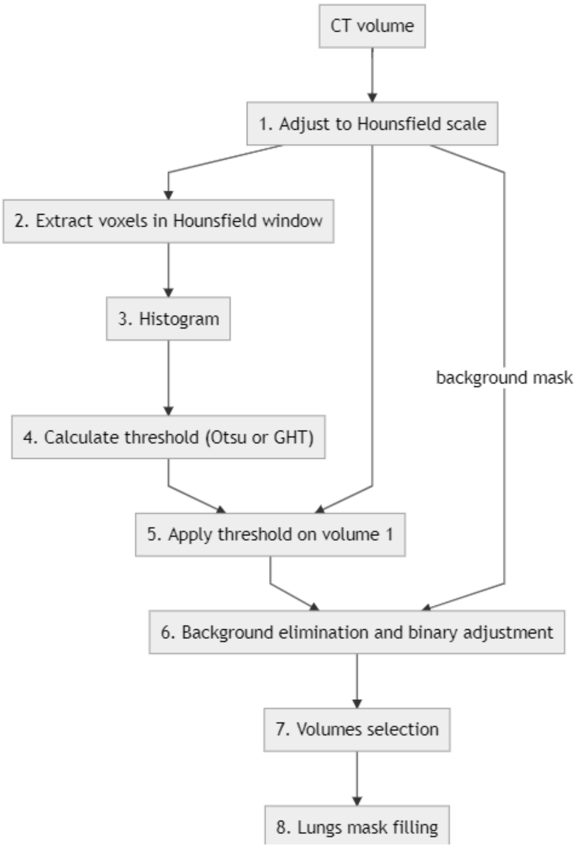
CT volume
1. Adjust to Hounsfield scale
2. Extract voxels in Hounsfield window
3. Histogram
4. Calculate threshold (Otsu or GHT)
background mask
5. Apply threshold on volume 1
6. Background elimination and binary adjustment
7. Volumes selection
8. Lungs mask filling

Figure 2. The proposed method, including pre-processing (1 and 2), thresholding (3 to 5) and post-processing (6 to 8) steps.

Post-processing

6) Background elimination (i) and binarization adjustment (ii): in the thresholded volume both the background and the lungs are black, and the rest of the body is white (figure 3). Given that only the lungs must be white, a simple inversion doesn't work. So, first, (i) we use a background mask (pixels originally marked as external to the chamber) to turn the outside values to white. Then, (ii) we invert the binarization slice by slice (2D). This must be done in 2D because if the thresholded values mistakenly include the bronchi and aerial pathways, it merges lungs to background, and then, the whole segmented volume would be erased in this step.

7) Volume selection: we kept only the three biggest 3D connected components. This step must be processed in 3D because the lower axial slices usually have fragmented parts of the mask, disabling a slicewise approach. It's only viable in 3D, as these fragments are axially connected. Keeping only the three biggest volumes is conservative given that it is not uncommon that the two lungs masks merge at some point, which also happens in the labels.

8) Mask filling: the remaining volumes masks are filled to include blood vessels that are usually not present in the masks.

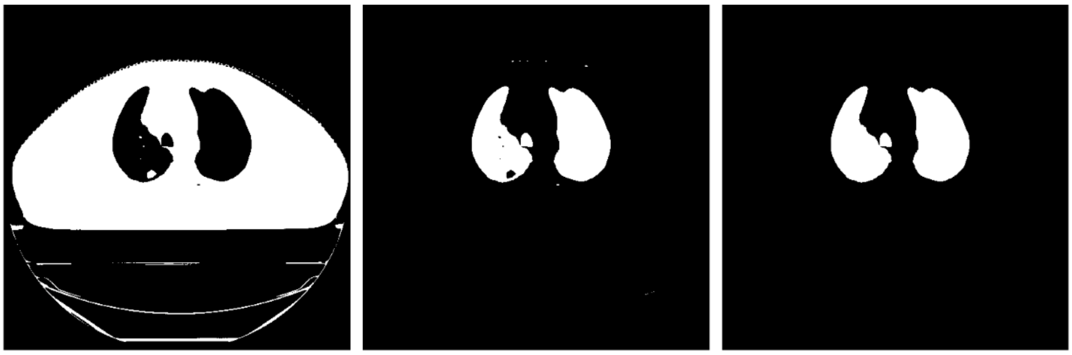

Figure 3. Example of post-processing. Left: thresholded slice, right before step 6 (left). Center: after step 6 binary adjustment and background exclusion. Right: after step 8, when the segmentation mask is filled. Note that in this case, the trachea lumen was wrongly included in the segmentation mask (see discussion).

3.3 UNet

We used a standard 2D UNet (Ronneberger, Fischer, & Brox, 2015)(python package segmentation models (Yakubovskiy, 2019)). We used 5 folds, each run one net instance is created, trained in four folds and tested in the remaining, alternating the folds used as test among runs. Folds were specified such that a given exam would be entirely in one fold or another, separating exams from training and testing. This way, the whole dataset was evaluated in test, and the performance in all the exams could be compared to the other methods.

For the UNet, each axial slice of each exam was resized to 256 by 256 pixels, and intensities were individually standardized. The UNet was trained for 40 epochs using batches of 64 slices shuffled between exams, using Adam optimizer (learning rate:10^{-3}), using binary cross-entropy loss, and the weights corresponding to the best observed loss were saved and used for testing. Tests were performed in each exam individually, sequentially evaluating all of its slices and calculating the Dice coefficient for the whole exam (3D).

3. 4 Equipment

We ran the proposed method on an Intel ® Core i7-8700 CPU, 3.20GHz, 16GB RAM. The UNet was trained and tested using a NVIDIA Tesla V100 16GB card.

3. 5 Statistical analyses

For the statistical analyses we used R (3.6.1). We used mixed model ANOVA on ranked data for comparing the Dice coefficient and processing time across the different methods, specifying exam as random factor, using the *nlme* package (Pinheiro, Bates, DebRoy, Sarkar, & R Core Team, 2013). This approach was adopted as data wasn't normally distributed and this is the closest nonparametric counterpart to repeated-measures ANOVA (Zimmerman & Zumbo, 1993). For multiple comparisons we used the Bonferroni correction method, using the *lsmeans* package(Lenth, 2016). For checking data normality, we used Shapiro-Wilk test.

4. Results

Considering that all methods were evaluated in the whole exam volume (3D), the proposed method on GHT had a Dice coefficient performance intermediary to the other two, the UNet being the top result. It's valid to point out that the proposed method using GHT had both its average and median Dice performance closer to the UNet results than to Otsu's. As it can be observed in figure 4, although all the three methods showed median Dice above 90%, its respective average performances were lower than its medians, due to the number of cases in which each method had a lower performance.

Table 1. Each method's Dice coefficient performance

Method	Dice coef. (avg. ± std.)	Dice coef. (median)
Proposed (GHT)	0.9135 ± 0.1061	0.9497
Proposed (Otsu)	0.7709 ± 0.2286	0.9085
UNet	0.9445 ± 0.1041	0.9712

The methods had significantly different Dice coefficient performances (mixed model ANOVA on ranked data, F: 68.5815, d. f..: 2, 118, p-value < 10^{-4}). Using the Bonferroni approach for multiple comparisons we observed statistically significant differences between every pair (GHT – Otsu: t ratio: 4.151, p-value: 0.0002; GHT – UNet: t ratio: -7.409; p-value < 10^{-4}; Otsu – UNet: t ratio: -11.560; p-value < 10^{-4}; d. f.: 118 for each of the tests). We adopted the ranking approach as the normality hypothesis for the distribution of Dice difference values was rejected (Shapiro Wilk, value: 0.7839, p-value: 5.0175 x 10^{-15}).

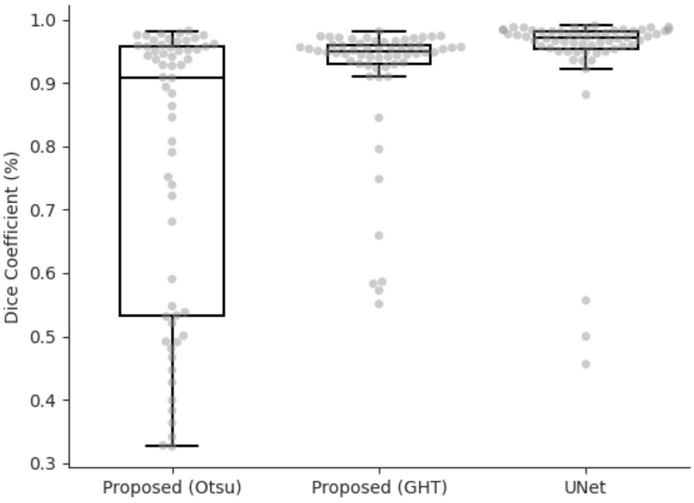

Figure 4. Dice coefficient performance of each method.

We also evaluated the processing time taken for whole volume segmentation in each one of the methods. In the proposed method, using both Otsu and GHT algorithms, the reported time includes the pre and post-processing steps, histogram calculation, thresholding and Dice calculation. As for the UNet, the reported time includes image resizing and standardization, inference, and Dice calculation. It doesn't include the extra time taken loading the UNet on the GPU, which is done only once for each fold. Even so, we measured the loading time, which was on average 1.17s.

Table 2. Each method's time performance

Method	Time (s) (avg. ± std.)	Time (s) (median)
Proposed (GHT)	1.8778 ± 0.6004	1.7099
Proposed (Otsu)	0.6607 ± 0.2802	0.5531
UNet	4.4086 ± 0.8372	4.4489

The tested methods show significantly different processing times (mixed model ANOVA on ranked data, F: 1646.023; d. f.: 2, 118; p-valor < 10^{-4}). The multiple comparison using Bonferroni also pointed statistically significant differences between all the methods (GHT – Otsu: t ratio: 27.886, p-value: < 10^{-4}; GHT – UNet: t ratio: -29.483, p-value: < 10^{-4}; Otsu - UNet t ratio: -57.369, p-valor: < 10^{-4}; d. f.: 118 for each of the tests). The time difference between methods also wasn't normally distributed (Shapiro-Wilk, value: 0.9718, p-value: 0.0010).

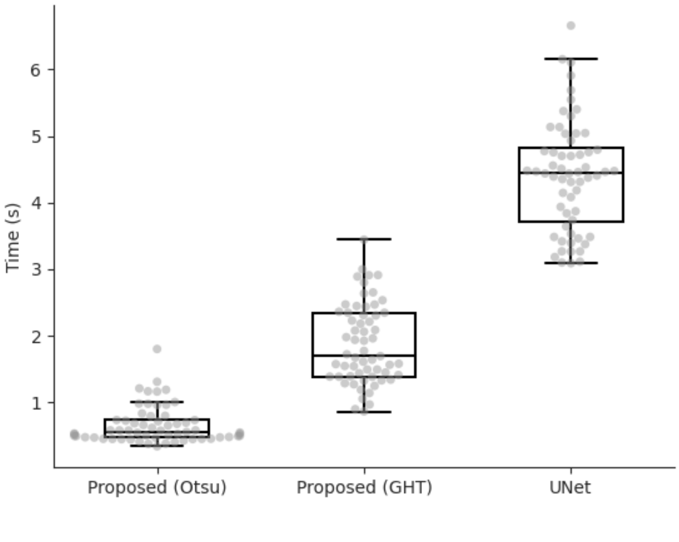

Figure 5. Each method's time performance distributions.

The proposed method is simpler that the UNet; as such, even running on CPU, it was 2.6 times faster. In this regard, one of the differences in the proposed method is that it doesn't require reshaping the images, which consumes more time than calculating the histogram. Also, the UNet inference takes longer than the simple application of the threshold.

5. Discussion

The proposed segmentation method showed a significant improvement just by changing from Otsu's to GHT, performing better than other classical methods in literature (Than et al., 2014; Zhou et al., 2016), although tested in different datasets, reaching an average and median Dice coefficient closer to the UNet's and other network (Hofmanninger et al., 2020) performance, while being 2.6 times faster. Thus, it seems to be efficient while performing well enough, being an adequate choice for a pre-processing step in a more sophisticated method, specific to some further diagnose.

The dispersions observed in the Dice values points that the methods differ in two aspects. Firstly, even though all the medians are above 90%, they differ statistically, showing a baseline difference between the methods. Secondly, the dispersions show in how many exams each method had a much lower Dice. For example, even though the UNet concentrates most results above 90%, it had a much lower quality results in three cases. In comparison, the proposed method using GHT had a lower performance in eight cases, but none was as low as any of the three lower UNet performances. This might indicate some generalization ability difference among the methods, where the proposed method using GHT is more general, however less precise more often. Compared to the proposed method using Otsu, it shows a significant improvement, given that it has fewer intermediate values and no values under 55% of Dice.

Usually, segmentation is not an end in itself (Akter et al., 2020; Huidrom et al., 2018; Liu et al., 2020; Pu et al., 2020). Often, it is a step for obtaining the slices containing lung parts, or to calculating lungs bounding box. It is common for deep learning methods for lungs clinical diagnostics to work on segmentation results. Sometimes the segmentation is also used for exclusion, as is the case when the focus of the method is the heart, and exclusion of the lungs volume might favor heart detection or segmentation, which might have been harder at start.

The proposed method using GHT has a performance close to deep learning methods (Hofmanninger et al., 2020), but has some qualities that might be advantageous in specific cases, it's off-the-shelf, fully automatic, non-supervised, is fast as it is a one-shot method, doesn't require further fine-tuning or transfer learning steps in order to be applied to new data, and thus, doesn't require extra labeled data for parameter fitting or learning. Even though this might easily be incorporated if necessary. Moreover, the proposed method is highly reproducible, being simple to apply not requiring specialized hardware. Although, an implementation of the proposed method using GHT in GPU is possible, using any of the well-known frameworks such as TensorFlow or pyTorch, what might make it faster. Given such considerations and its performance, it's an adequate pre-processing method that might serve well other, more sophisticated, diagnostic analysis.

Some further improvements for the proposed method using GHT regarding lungs segmentation quality is possible. Considering that some of the observed current errors are the inclusion of bronchia and trachea (as seen in figure 3), a simple improvement could be the adoption of a smaller Hounsfield windows, as the one currently used is very broad and there might be some margin for a cut off without compromising generality. Another possibility, as these errors might deform the segmentation mask, might be the inclusion of an active shape model (Chen, Udupa, Bagci, Zhuge, & Yao, 2012) or active appearance ("Automatic Lung Field Segmentation in X-ray Radiographs Using Statistical Shape and Appearance Models," 2016) in the volume's post-processing step. However, this would also require validation step and extra data.

6. Acknowledgments

This work was supported by Foxconn Brazil and Zerbini Foundation as part of the research project "Machine Learning in Cardiovascular Medicine".

7. References

Akter, O., Moni, M. A., Islam, M. M., Quinn, J. M. W., & Kamal, A. H. M. (2020). Lung cancer detection using enhanced segmentation accuracy. *Applied Intelligence*, (October). https://doi.org/10.1007/s10489-020-02046-y

Automatic Lung Field Segmentation in X-ray Radiographs Using Statistical Shape and Appearance Models. (2016). *Journal of Medical Imaging and Health Informatics*, *6*(2).

Barron, J. T. (2020). *A Generalization of Otsu's Method and Minimum Error Thresholding*. Retrieved from http://arxiv.org/abs/2007.07350

Chen, X., Udupa, J. K., Bagci, U., Zhuge, Y., & Yao, J. (2012). Medical image segmentation by combining graph cuts and oriented active appearance models. *IEEE Transactions on Image Processing*, *21*(4), 2035–2046. https://doi.org/10.1109/TIP.2012.2186306

Chilakala, L. R. (2020). *Optimal deep belief network with opposition-based hybrid grasshopper and honeybee optimization algorithm for lung cancer classification : A DBNGHHB approach*. (April), 1–20. https://doi.org/10.1002/ima.22515

Doyle, W. (1962). Operations Useful for Similarity-Invariant Pattern Recognition. *Journal of the ACM*, *9*(2), 259–267. https://doi.org/10.1145/321119.321123

Gordaliza, P. M., Muñoz-Barrutia, A., Abella, M., Desco, M., Sharpe, S., & Vaquero, J. J. (2018). Unsupervised CT Lung Image Segmentation of a Mycobacterium Tuberculosis Infection Model. *Scientific Reports*, *8*(1), 1–10. https://doi.org/10.1038/s41598-018-28100-x

Hofmanninger, J., Prayer, F., Pan, J., Röhrich, S., Prosch, H., & Langs, G. (2020). Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem. *European Radiology Experimental*, *4*(1), 50. https://doi.org/10.1186/s41747-020-00173-2

Huidrom, R., Chanu, Y. J., & Singh, K. M. (2018). Automated lung segmentation on computed tomography image for the diagnosis of lung cancer. *Computacion y*

Sistemas, *22*(3), 907–915. https://doi.org/10.13053/CyS-22-3-2526

Im, H.-J., Solaiyappan, M., Lee, I., Bradshaw, T., Daw, N. C., Navid, F., … Cho, S. Y. (2018). Multi-level otsu method to define metabolic tumor volume in positron emission tomography. *American Journal of Nuclear Medicine and Molecular Imaging*, *8*(6), 373–386. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/30697457%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC6334209

Kittler, J., & Illingworth, J. (1986). Minimum error thresholding. *Pattern Recognition*, *19*(1), 41–47. https://doi.org/10.1016/0031-3203(86)90030-0

Kumar, S., Pant, M., & Ray, A. K. (2012). Segmentation of CT Lung Images Based on 2D Otsu Optimized by Differential Evolution. In Deep, K and Nagar, A and Pant, M and Bansal, JC (Ed.), *PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON SOFT COMPUTING FOR PROBLEM SOLVING (SOCPROS 2011), VOL 2* (pp. 891–902). HEIDELBERGER PLATZ 3, D-14197 BERLIN, GERMANY: SPRINGER-VERLAG BERLIN.

Lenth, R. V. (2016). Least-Squares Means: The {R} Package {lsmeans}. *Journal of Statistical Software*, *69*(1), 1–33. https://doi.org/10.18637/jss.v069.i01

Liu, C., & Pang, M. (2020). Extracting Lungs from CT Images via Deep Convolutional Neural Network Based Segmentation and Two-Pass Contour Refinement. *Journal of Digital Imaging*, *33*(6), 1465–1478. https://doi.org/10.1007/s10278-020-00388-0

Liu, C., Zhao, R., Xie, W., & Pang, M. (2020). Pathological lung segmentation based on random forest combined with deep model and multi-scale superpixels. *Neural Processing Letters*, *52*(2), 1631–1649. https://doi.org/10.1007/s11063-020-10330-8

McCarthy, D. (2001). Chest CT in “Post” COVID-19: What the Radiologist Must Know. *Progress in Transplantation*, *11*(3), 162–162. https://doi.org/10.7182/prtr.11.3.hh66651262116783

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transaction on Systems, Man and Cybernetics*, *smc-9*(1), 62–66.

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., & R Core Team. (2013). *nlme: Linear and Nonlinear Mixed Effects Models*.

Pu, J., Leader, J. K., Bandos, A., Ke, S., Wang, J., Shi, J., … Jin, C. (2020). Automated quantification of COVID-19 severity and progression using chest CT images. *European Radiology*. https://doi.org/10.1007/s00330-020-07156-2

Rajagopalan, K., & Babu, S. (2020). The detection of lung cancer using massive artificial neural network based on soft tissue technique. *BMC Medical Informatics and Decision Making*, *20*(1), 1–13. https://doi.org/10.1186/s12911-020-01220-z

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9351*, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

Sara Naybandi Atashi, Zeinab Naderpour, & Saeedi, K. G. M. (2021). An Eight-week

Follow-up Study in Patients With COVID-19 Respiratory Failure: Delayed Recovery or Lung Sequel. *Case Reports in Clinical Practice*, *5*(Covid-19), 153–157.

Than, J. C. M., Noor, N. M., Rijal, O. M., Yunus, A., & Kassim, R. M. (2014). Lung segmentation for HRCT thorax images using radon transform and accumulating pixel width. *IEEE TENSYMP 2014 - 2014 IEEE Region 10 Symposium*, 157–161. https://doi.org/10.1109/tenconspring.2014.6863016

Yakubovskiy, P. (2019). Segmentation Models. *GitHub Repository*. GitHub.

Yang, J., Sharp, G., Veeraraghavan, H., van Elmpt, W., Dekker, A., Lustberg, T., & Gooding, M. (2017). Data from lung CT segmentation challenge. *The Cancer Imaging Archive*.

Zhou, H., Goldgof, D. B., Hawkins, S., Wei, L., Liu, Y., Creighton, D., … Nahavandi, S. (2016). A Robust Approach for Automated Lung Segmentation in Thoracic CT. *Proceedings - 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, 2267–2272. https://doi.org/10.1109/SMC.2015.396

Zimmerman, D. W., & Zumbo, B. D. (1993). Relative power of the wilcoxon test, the friedman test, and repeated-measures ANOVA on ranks. *Journal of Experimental Education*, *62*(1), 75–86. https://doi.org/10.1080/00220973.1993.9943832

8. *Appendix*

The formulas were adapted from (Barron, 2020) to use the same notation for both methods.

Otsu's method

In essence, it calculates the threshold value t for which the total intraclass variation is minimal, which will correspond to maximal interclass variance. It corresponds to the K means solution when k is two.

$$w_0(t) = \sum_{i=0}^{t-1} n_i \qquad \mu_0(t) = \frac{\sum_{i=0}^{t-1} n_i x_i}{\omega_0(t)}$$

$$w_1(t) = \sum_{i=t}^{L-1} n_i \qquad \mu_1(t) = \frac{\sum_{i=t}^{L-1} n_i x_i}{\omega_1(t)}$$

$$\sigma_b^2(t) = w_0(t) w_1(t) [\mu_0(t) - \mu_1(t)]^2$$

Above, n_i is the bin's height, x_i is the bin's intensity value, σ_{b}^2 is the total inter-class variance, given by classes probabilities w_0 and w_1 and respective classes means μ_0 and μ_1, all evaluated at the threshold value t which maximizes σ_b^2. L corresponds to the number of bins in the histogram.

Generalized Histogram Thresholding (GHT)

GHT approaches threshold calculation as a mixture of two gaussians. Each fitted gaussian has a mean, a standard deviation and a weight. An inverted rescaled chi-square distribution as a prior for the variances (σ_k), and a beta distribution as a prior for the weights (π_k).

$$p\left(I_{x,y} \middle| z_{x,y} = k\right) = \pi_k \, \mathcal{N}\left(I_{x,y} \middle| \mu_k, \sigma_k\right) \; \chi^2_{SI}(\sigma_k | \pi_k \nu, \rho^2) \; Beta(\pi_k | \kappa, \omega)$$

$$\pi_0(t) = w_0(t) / \|n\|_1 \qquad \pi_1(t) = w_1(t) / \|n\|_1 = 1 - \pi_0(t)$$

$$d_0(t) = \sum_{i=0}^{t-1} n_i x_i^2 - w_0(t)\left(\mu_0(t)\right)^2 = \sigma_0^2 \qquad d_1(t) = \sum_{i=t}^{L-1} n_i x_i^2 - w_1(t)\left(\mu_1(t)\right)^2 = \sigma_1^2$$

$$v_0(t) = \frac{\pi_0(t)\, \nu\, \tau^2 + d_0(t)}{\pi_0(t)\, \nu + w_0(t)} \qquad v_1(t) = \frac{\pi_1(t)\, \nu\, \tau^2 + d_1(t)}{\pi_1(t)\, \nu + w_1(t)}$$

$$f_0(t) = -\frac{d_0(t)}{v_0(t)} - w_0(t) \, log \, v_0(t) \; + 2[w_0(t) + \kappa\omega(t)] \, log \, w_0(t)$$

$$f_1(t) = -\frac{d_1(t)}{v_1(t)} - w_1(t) \, log \, v_1(t) \; + 2[w_1(t) + \kappa(1 - \omega(t))] \, log \, w_1(t)$$

$$GHT(x, \mathrm{n}, \nu, \tau, \kappa, \omega) = x_{argmax(t)}\left(f_0(t) + f_1(t)\right)$$

The method sets the extra parameters as $\nu = \mathrm{sum(n)}, \tau = \sqrt{1/12}, \kappa = \mathrm{sum(n)}$ and $\omega = 0.5$.