Credits and original work

December 21, 2020

In this document the ideas, techniques, algorithms and code, which were used for this work are listed. Some of these are original and some built upon existing work as a foundation. Most of the plots and figures were created with the help of the Matplotlib library [1].

- Preprocessing: for this part, several techniques and code were used from two separate solutions to the preprocessing part of the Lung Nodule Analysis 2016 (LUNA16) challenge [2] [3].

    - Loading the DICOM files, resampling and normalising the final results were performed using code from [2].

    - Segmentation of the lungs' area was performed using and modifying the code from [3], with the substituiton of the binary closure step with the convex hull, giving better results in areas where there are juxtapleaural nodules.

    - Importing the annotations and cropping the nodules was done with original code.

- Fast Marching Method implementation: for this part most of the code was written from the ground up, except for the use of some functions from Python image processing libraries. Several techniques were designed taking inspiration from already existing work.

    - For the FMM preparation, the smoothing and gradient magnitude were performed with functions from the SimpleITK library [4]. The seed grid generation method draws heavily from [5], where an automatic initialisation for the seeds is proposed based on statistical analysis. A similar method was designed that makes use of the image gradient and uses a kindred shifting technique, the implementation was done with original code. The padding scheme which improves performance for juxtapleureal nodules was an original idea.

    - FMM was implemented by writing the code completely from the beginning, by following the algorithm as described by J.A. Sethian

in [6]. For the addition of region assigning to multiple fronts the idea from [7] was implemented, where a new region growing method is proposed for segmenting images using a multi-label fast marching algorithm.

– The scikit-learn Python library [8] library was used to execute K-means, and cluster merging was carried with an often used criteria for region growing, for which the recursive algorithm was implemented from the ground up. The trackbars were added using functions from the matplotlib library.

- Evaluation: for this part most of the code was written from the ground up, except for the use of some basic functions from Python image processing libraries, and for the Matlab [9] implementation of active contours. Several techniques were designed taking inspiration from already existing concepts.

– For the comparison method the *activecontours* function was used from Matlab as is, without making any significant modifications.

– For the subjective evaluation the questionnaires were written from scratch keeping in mind that the radiologists task and the scoring system should be well defined. To extract the data from the completed forms the PyPDF2 Python toolkit [10] was used.

– For the objective evaluation the phantom dataset was generated by using image processing tools from GIMP. Implementing the evaluation metrics was done writing original code by following the definitions of both IoU and long-axis/short-axis diameter measure.

References

[1] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.

[2] Guido Zuidhof. *Full Preprocessing Tutorial*, 2016).

[3] Arnav Jain. *Candidate Generation and LUNA16 preprocessing*, 2017).

[4] Ziv Yaniv, Bradley Lowekamp, Hans Johnson, and Richard Beare. Simpleitk image-analysis notebooks: a collaborative environment for education and reproducible research. *Journal of Digital Imaging*, 31, 11 2017.

[5] J. Hearn. Competitive medical image segmentation with the fast marching method. 2008.

[6] J A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[7] Eftichis Sifakis, Christophe Garcia, and Georgios Tziritas. Bayesian level sets for image segmentation. 13(1):44–64, March 2002.

[8] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[9] MATLAB. *version 9.8 (R2020a)*. The MathWorks Inc., Natick, Massachusetts, 2020.

[10] Inc Phaseit. *PyPDF2 - The PdfFileReader Class*, 2016.