An efficient automatic segmentation of left ventricle by hybrid approach

1st M. Arsalan KHAWAJA
Condorcet university center
University of Burgundy
Le-Creusot, France
Muhammad-Arsalan_Khawaja@etu.u-bourgogne.fr

2nd Thien Bao Bui
Condorcet university center
University of Burgundy
Le-Creusot, France
Thien-Bao_Bui@etu.u-bourgogne.fr

***Abstract*—Left Ventricle is an integral part of the heart. It pumps blood into the body. Cardiovascular diseases are quite common in modern world. The health of the heart can be studied using left ventricle. Most of the heart diseases can be diagnosed by retrieving important information from systole and diastole operations of heart via effective medical imaging analysis of Left Ventricle (LV). The segmentation of Left Ventricle plays an important role in retrieving the ejection fraction (a parameter which indicates the heart's health). Since Magnetic Resonance Images (MRI) are noisy and not really perfect, scientific community is always curious to find more efficient, reliable, robust and more accurate segmentation methods to calculate ejection fraction. In this research project, we use a novel hybrid approach for segmentation. K-means, along with border following algorithm is used for segmentation. The results are good and discussed in the last section.**

***Index Terms*—K-means, Python, Border Following Algorithm, Left Ventricle, Adaptive Smoothing, Clustering, Hough Transform.**

I. Introduction

The segmentation of left ventricle is of utmost importance when it come to diagnosis of heart diseases. It takes too much of a time for manual segmentation and is a painstaking task. This segmentation establishes and provides us with important knowledge of critical parameters which can help us calculate certain ratios. These ratios help doctors to predict the condition of heart and in some cases diagnose a particular disease. In short, an efficient segmentation of heart ends up saving patient life.

Unfortunately doctors are too busy to manually segment each of the medical image and it also wastes a lot of their time. They are more needed to patients. There is an inevitable demand for some automatic segmentation of left ventricle. A lot of work has been done in this field, but because medical images are not really the most perfect images, segmentation algorithms are destined to struggle with accuracy.

We went through the algorithms and decided to use Hybrid Approach for our segmentation. In hybrid approach, we intend to use more than one approaches / algorithms to work simultaneously on the segmentation of the image. In this project we used mainly primarily K-Means algorithm inspired by [6]. Alongside, a border following algorithm inspired from [8] was used to enhance segmentation results. The preprocessing was done with adaptive smoothing algorithm. Finally we optimized the performance of our algorithm by repeated experiments of using the algorithm with different datasets and setting the parameters to optimal value.

II. Anatomy of Heart

The heart is made up of four chambers: two upper chambers known as the left atrium and right atrium and two lower chambers called the left and right ventricles. These are shown in following figure .

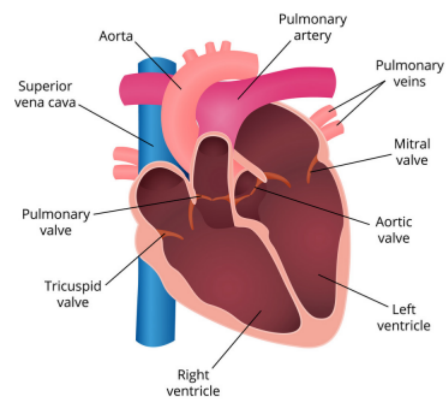

Fig. 1. The left ventricle is longer and more conical in shape. In short axis, the left ventricle appears circular in shape. The image is taken from [1]

The left ventricle is considered an important part of the cardiovascular system. It is thought off as a pump that supplies blood to the body. The mass of the left ventricle, as estimated by magnetic resonance imaging, averages $143g \pm 38.4g$, with a range of 87–$224g$ [4].

Another important thing to discuss are the two phases of the cardiac cycle. They are called Systole and Diastole. They occur as the heart beats, pumping blood through a system of blood vessels that carry blood to every part of the body. Systole occurs when the heart contracts to pump blood out, and diastole occurs when the heart relaxes after contraction.

A. Ejection Fraction

Ejection fraction (EF) refers to how well your left ventricle (or right ventricle) pumps blood with each heart beat [6]. The ejection fraction can be calculated as follows:

$$\mathrm{EF}=\frac{V_{\text{endo}}\left(t_{\mathrm{D}}\right)-V_{\text{endo}}(t\mathrm{s})}{V_{\text{endo}}\left(t_{\mathrm{D}}\right)} \tag{1}$$

Here note that V_{endo} is the volume of the inner walls of the heart, $V_{endo}(t_D) = max_t[V_{endo}(t)]$ is the end-diastolic volume and$V_{endo}(t_S) = min_t[V_{endo}(t)]$ is the end-systolic volume.

III. Methodology

The first task was to identify the tasks in the project. After we were done with identifying the tasks, we came to their implementation on python. We studied and learned to use the needed libraries in python. The libraries have been given in Appendix A. The following flowchart gives a very articulate work flow 2 demonstration.

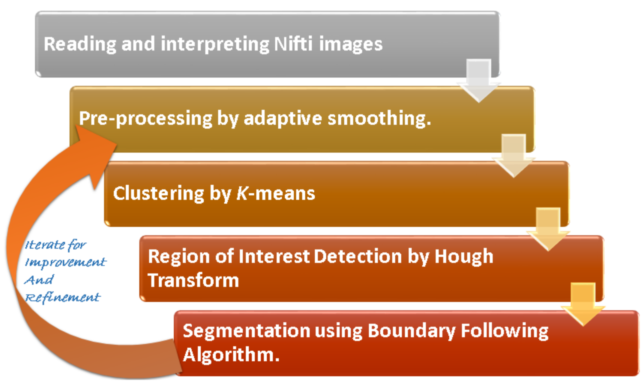

Fig. 2. The flowchart demonstrates work flow. Necessary iterations were done to find the optimal combination of all algorithm parameters with the goal of achieving the highest accuracy possible.

The following section will summarize the theory of our chosen papers.

A. Adaptive non linear smoothing

The smoothing technique that is used in the pre-processing stage is adapted from [3]. A smoothing filter will be applied on every images before clustering. By using this edge-preserving filter, noise is removed while maintaining the important edge information.

The idea is to adapt pixel intensities to the local attributes of an image on the basis of discontinuity measures. Note that in this algorithm there are two kind of discontinuities: Local and Contextual discontinuity. The local discontinuities indicate the local detailed structure while the contextual discontinuities represent the important features (4). This novel approach joins these both discontinuities to preserve edges and at same time smooths image.

In order to measure local discontinuities, four detectors for pixel (x, y) along four directions is defined. These four directions are vertical (V), horizontal (H), diagonal (D), and counter-diagonal (C), respectively.

$$\begin{aligned} E_{H_{xy}} &= |I_{x+1,y} - I_{x-1,y}| \\ E_{V_{xy}} &= |I_{x,y+1} - I_{x,y-1}| \\ E_{C_{xy}} &= |I_{x+1,y+1} - I_{x-1,y-1}| \\ E_{D_{xy}} &= |I_{x+1,y-1} - I_{x-1,y+1}| \end{aligned} \tag{2}$$

Here $I_{(x,y)}$ is the intensity of pixel (x, y). In order to illustrate the idea, we constructed this figure 3.

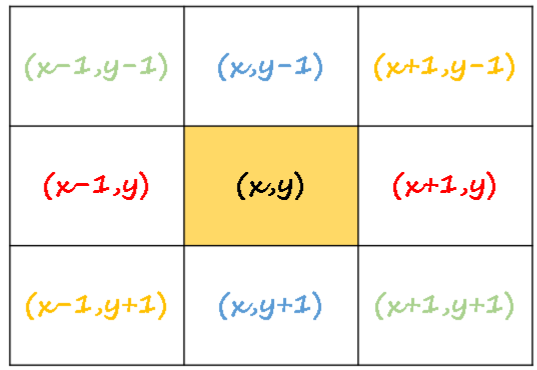

Fig. 3. The pixel (x, y) is our pixel of operation.

Then the local discontinuity is obtained by calculating the differences around the neighbouring pixels, as shown in this equation below:

$$E_{xy} = \frac{E_{H_{xy}} + E_{V_{xy}} + E_{C_{xy}} + E_{D_{xy}}}{4} \tag{3}$$

In order to detect contextual discontinuities, we use spatial variance to make a measure. First, a contextual neighborhood $N_{xy}(R)$ associated with pixel $(x; y)$ is defined as:

$$N_{xy}(R) = \{(i,j)|x - R \le i \le x + R, y - R \le j \le y + R\} \tag{4}$$

Note that, here $R(R > 1)$ is a parameter that determines the size of this contextual neighborhood. The contextual neighborhood is defined here without explicitly counting image boundaries. The parameter R describes a spatial scale or a resolution that critically determines results of the contextual discontinuity measure. We calculate the mean of all the pixels in this kernel as:

$$\mu_{xy}(R) = \frac{\sum_{(i,j)\in N_{xy}(R)} I_{i,j}}{|N_{xy}(R)|} \tag{5}$$

and the spatial variance, $\sigma^2_{xy}(R)$, can be calculated as:

$$\begin{aligned} \sigma^2_{xy}(R) &= \frac{\sum_{(i,j)\in N_{xy}(R)} \left(I_{i,j} - \mu_{ij}(R)\right)^2}{|N_{xy}(R)|} \\ &= \frac{\sum_{(i,j)\in N_{xy}(R)} I^2_{i,j}}{|N_{xy}(R)|} - \left(\frac{\sum_{(i,j)\in N_{xy}(R)} I_{i,j}}{|N_{xy}(R)|}\right)^2 \end{aligned} \tag{6}$$

The normalized $\tilde{\sigma}^2_{xy}(R)$ can be written as follow:

$$\tilde{\sigma}^2_{xy}(R) = \frac{\sigma^2_{xy}(R) - \sigma^2_{\min}(R)}{\sigma^2_{\max}(R) - \sigma^2_{\min}(R)} \tag{7}$$

where $\sigma^2_{\max}(R)$ and $\sigma^2_{\min}(R)$ are the maximal and minimal spatial variance across the entire image, respectively. Intuitively, $\tilde{\sigma}^2_{xy}(R)$ reflects the relative degree of the contextual discontinuities for pixel $(x; y)$.

The local attributes of a pixel with a high contextual discontinuity should be preserved and those of a pixel with a low contextual discontinuity should be smoothed toward homogeneity. However, both noise and trivial features irrelevant to a given problem lead to the complexity for visual information processing. In order to reduce their influence in the contextual discontinuity estimation, we introduce a transformation into $\tilde{\sigma}^2_{xy}(R)$:

$$\Phi\left(\tilde{\sigma}^2_{xy}(R), \theta_\sigma\right) = \begin{cases} 0 & \tilde{\sigma}^2_{xy}(R) < \theta_\sigma \\ \tilde{\sigma}^2_{xy}(R) & \tilde{\sigma}^2_{xy}(R) \geq \theta_\sigma \end{cases} \tag{8}$$

Here $\theta_\sigma\,(0 \leq \theta_\sigma \leq 1)$ is a threshold. For a given θ_σ, $\Phi\left(\tilde{\sigma}^2_{\pi v}(R), \theta_\sigma\right)$ gives us lead to contextual discontinuity map.In summary, pixels of same area or region are supposed to have low contextual discontinuities except the case, for those pixels located near boundaries that have high contextual discontinuities. Unlike local discontinuity measure, contextual discontinuity measure is relatively insensitive to local intensity changes.

Based on the local and contextual discontinuity, an adaptive algorithm scheme is introduced:

$$I^{(t+1)}_{xy} = I^{(t)}_{xy} + \eta_{xy} \frac{\Sigma_{(i,j)\in N_{xy}(1)/[(x,y)]}\eta_{ij} y^{(t)}_{ij}\left(I^{(t)}_{i,j} - I^{(t)}_{x,y}\right)}{\Sigma_{(i,j)\in N_{xy}(1)/[(x,y)\}}\eta_{ij}\gamma^{(t)}_{ij}} \tag{9}$$

where

$$\eta_{ij} = \exp\left(-\alpha\Phi\left(\tilde{\sigma}^2_{ij}(R), \theta_\sigma\right)\right)$$
$$\gamma^{(t)}_{ij} = \exp\left(-E^{(t)}_{ij}/S\right)$$

The main goal of smoothing is to alleviate the complexity for subsequent processes in early vision. With this preprocessing, the results of our MRI image are shown in figure 4 (image taken from patient 01, 2^{nd} ED slice).

In short, we summarize the definition and the effect of each hyper-parameter of this algorithm in the table below.

The smoothing and the edge-preserving degree can be configured by following these details.

B. K-means Clustering

The smoothed images were then clustered using K-means algorithm proposed by Duda and Hart [5], [7]. This algorithm has four steps as shown in figure 5 to search for the image clusters.

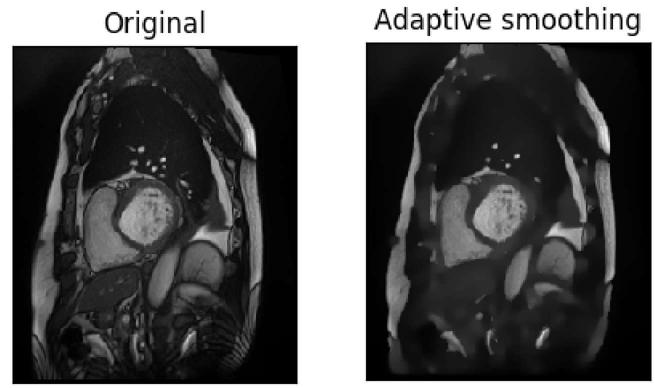

Fig. 4. On the left, is the original image. On the right is the smoothed image obtained by setting the hyper-parameters: $\theta = 0.1$, $\alpha = 15$, $S = 6$ after 5 iterations with a 3×3 kernel. We can clearly notice that the algorithm has blurred the details of object, but the edge is still preserved. This will help us in getting great accuracy for segmentation.

Parameter	Value set	**Indication**
θ	Large Value	Tend to extract only major regions.
	Small Value	Tend to preserve details
	if $\tilde{\sigma}^2_{xy}(R) < \theta_\sigma$	The current pixel is noise
	if $\tilde{\sigma}^2_{xy}(R) > \theta_\sigma$	The current pixel is a feature
α	Small value	Fast smoothing and discontinuity reduction
	Large value	Slow smoothing and preserving important features
S	Small value	Leads to better preservation of details
	Large value	All discontinuities disappear

TABLE I
ANALYSIS OF CHANGING SMOOTHING PARAMETERS

The key idea of the K-means algorithm is to divide M points in N dimensions into K clusters, such that the within-cluster sum of squares is minimized. K-means Clustering is categorized as unsupervised learning, which means that it does not need any intervention by human in our project and leads way to automatic segmentation of left ventricle. The figure (6) shows results after applying K-means clustering.

C. Automatic localization and segmentation of Left Ventricle

The input of this stage is a clustered image using K-means algorithm (6). We need to recognize the left ventricular before segmenting it from the image. At this stage, we adapt a method from other paper. The original approach is somehow impossible to us due to the lack of explanation. For the automatic localization task, we use the Hough Gradient method [2], while the segmentation task is completed by implementing the border following algorithm [8].

The Hough Gradient method is a combination of Canny Edge detector and the Hough Circle transform technique. It is used to determine the circular shape in an image. In order to automatically localize the left ventricular, we crop the region of interest, which is defined as the space ranging from the first quarter to the third quarter of the image width and height, based on the shape descriptor (the left ventricular is not in the

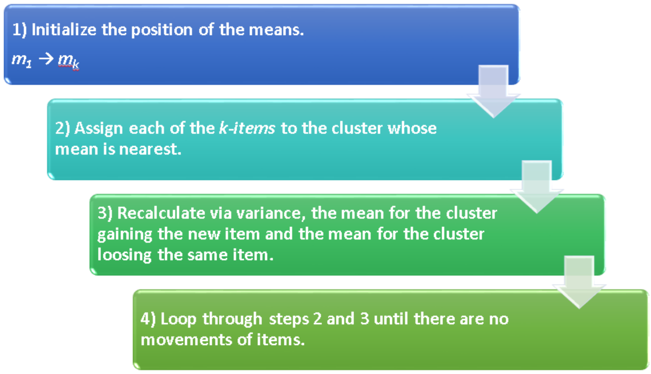

Fig. 5. The figure demonstrates the process of K-Means algorithm step by step. The algorithm is quite simple and is widely used in Machine learning, Economics, Data science and many other fields.

Fig. 6. Clustering of the original image as shown in figure 4. The image has been clustered using K means algorithm with $k = 3$, where k is the number of clusters.

peripheral of the image). Then the center of left ventricular will be determined by this method.

As shown in the figure 6, the left ventricular is on the same cluster with other shapes. Since the left ventricular belongs to a cluster with other shapes, an effective method to segment it is required. The border following algorithm is chosen for this task. Border following is one of the most important and popular technique in segmentation of binary images. It derives a sequence of the coordinates or the chain codes from the border between a connected component of 1-pixels[1] (1-component) and a connected component of 0-pixels[2] (background or hole).

This algorithm is illustrated in figure 7. It sees the component as 1 and background as 0. When it moves through the pixel grid passing each pixel watching out their intensity (0 or 1). As soon as it observes a jump from 0-Pixel to 1-Pixel, it memorizes the pixel location and value. After it memorizes[3] all the jumps i.e switching from 0-Pixel to 1-Pixel and vice versa. It categorizes the the border in number of objects depending whether the pixel locations are connected to each other or they are sparsely located.

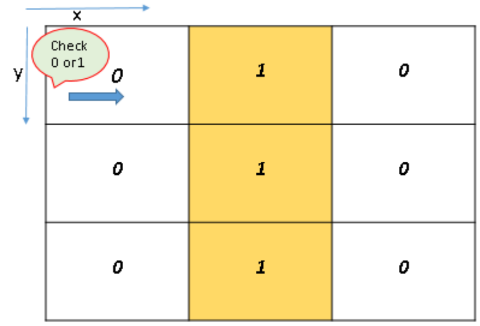

Fig. 7. A 3×3 image demonstrates the border following algorithm. The yellow pixels are some object border. The algorithm moves through the pixel grid looking for the jump in pixel intensities. It memorizes that jump and categorizes them to respective objects according to their spacial location.

All the contours obtained from this algorithm will be compared with the calculated center from Hough Gradient method. The closest contour will be considered as the left ventricular.

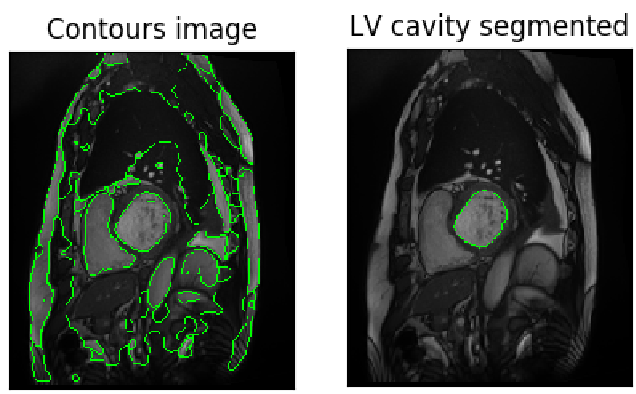

Fig. 8. The left image is the original image with the contours detected. Combining with the left ventricular center that we obtained from the algorithm, we can extract the left ventricular cavity, as shown in the right image - final segmented image. The left ventricle is indicated by green contour. The original image is shown in figure 4.

IV. Implementation and Results

A. Dataset

The dataset is provided by Dijon University Hospital Center. It comprises of 100 Magnetic Resonance examinations, dividing into 5 classes: Normal, Systolic heart failure with infarction, Dilated cardiomyopathy, Hypertrophic cardiomyopathy, Abnormal right ventricle. All images are short axis slices, in Nifti format. The ground truth label field is label as 0,1,2,3 representing pixels located in the background, right ventricular cavity, myocardium and left ventricular cavity. Therefore, preprocessing the ground truth label is necessary since our task concerns only the left

[1]Pixels with densities 1 are called the 1-pixel
[2]Pixels with densities 0 are called the 0-pixel
[3]By memorize we mean it saves the pixel location in a vector

ventricular cavity. The number of images of each examination is various. In this paper, we use only the images from the end diastolic and end systolic slices of each patient. The bar graph following is for illustration.

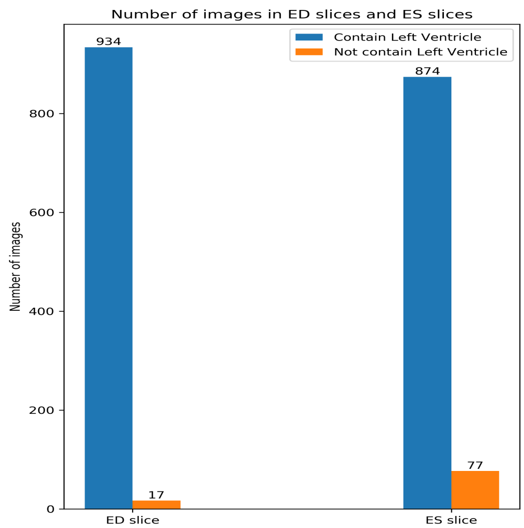

Fig. 9. Total number of images in ED and ES slices.

B. Implementation details

In this section we will discuss the implementation of algorithm. Python programming language was used for the implementation. The methodology from figure 2 was observed to complete the project. We faced challenges in almost every stage but our resilience and motivation kept us going. The following subsections take us step by step through the ladder of project.

To begin with, fine tuning the smoothing algorithm is considered as the most time-consuming task of this paper, since we need to find a group of hyper-parameters that works for nearly 2000 different images. It is ideal when this algorithm helps highlighting the difference between the left ventricular and its surrounding pixels. In order to achieve it, we also apply the contrast stretching technique on the smoothed image. The following figure illustrates the difference between a good and a bad smoothing effect (image taken from the patient 20, 1^{st} ED slice).

For K-means clustering, we set the number of cluster $k = 3$. It seems to be the optimal choice in our case. In this dataset, it is recognized that the pixel intensity of the left ventricular is not homogeneous for a larger number of images. Therefore, when we set the value of k greater than 3, the left ventricle will be separated. It is shown in this figure below (image taken from patient 25, 6^{th} ED slice):

In addition, there are a lot of images in which the pixels surrounding and inside left ventricle are similar. If we set $k = 2$, the left ventricle could be enlarged. It also affects the accuracy of our algorithm.

For the implementation of border following algorithm, it is necessary to mention that this algorithm can work only on

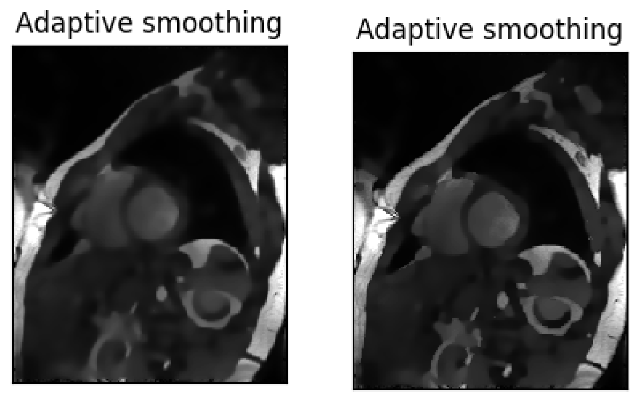

Fig. 10. The left image is a bad smoothing image since it cannot highlight the difference between the left ventricular cavity and its surrounding pixels, on the contrary to the right image which is a good smoothing image. It affects badly the result of the segmentation since the K-means algorithm depends heavily on the pixel intensities. Indeed, the left image gives 0% as the result for Dice metric, while the right image gives 94%.

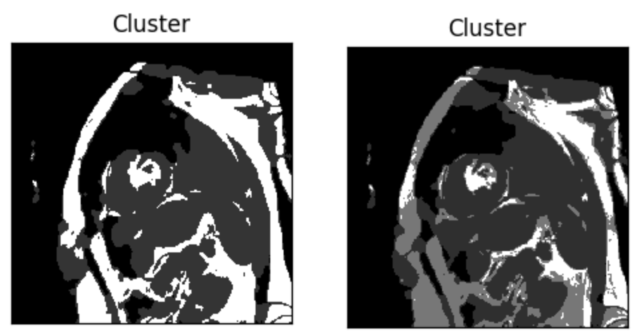

Fig. 11. The right image is the result with $k = 4$. The left ventricle is clearly separated, on the contrary to the left image with $k = 3$.

binary image. If we set $k = 2$, no action need to take. In case we set $k > 2$, we need to process the clustered image before passing it to this function. It can be done by separating each cluster from each other. The number of binary images therefore will be equal to the number of cluster. Then we apply border algorithm on each image and get the contour that satisfy our condition.

By testing various set of hyper-parameters following the rules mentioned in table I, we come up with this set:

Parameter	Value
θ	0.1
α	15
S	6
Kernel size	5×5
Number of iteration	5
Number of cluster	3

TABLE II
HYPER-PARAMETER SETTINGS FOR THE APPLICATION

C. Results and discussion

To measure the accuracy of our segmentation algorithm, we choose several metrics namely mean Intersection Over Union (mIOU), Dice Coefficient Index and Root mean square error (RMSE). The segmented image will be converted to a binary mask (an image of 0 and 1 values) before comparing with the ground truth image. The result is shown in the following table:

mIOU	Dice Coefficient	RMSE
81.2%	70.5%	0.05

TABLE III
ALGORITHM ACCURACY

During the ACDC challenge 2017, 9 out of the 10 cardiac segmentation methods were based on deep learning. In particular,the 8 best-ranked methods were all deep learning ones, with the accuracy (DICE metric) always above 90%. It is hard to find the non deep learning methods to make a comparison with our method. On the other hand, our original paper [6] use root mean square error as the metric. They achieve the error result 4.765 while our method got 0.06. However, the comparison with the original paper is also impossible since our implementation is not fully based on this paper, not to mention the difference in 2 datasets. We also tried to find their dataset to perform our algorithm on it but they do not mention it in their paper.

The accuracy is not very good. There are two main reasons. Firstly, our main method is K-means clustering, which perform clustering based on the pixel intensities. In this dataset, the relation between the pixel intensities of the left ventricle and their surroundings are diverged. Many cases show that the left ventricle is quite different with its surrounding, while the others cases does not. This algorithm gives poor results on many cases where the left ventricle are nearly immersed in the background, such as the dark blood images, the apex image of each slice, which is known as the time when the left ventricle is at its smallest size, etc. The left ventricle in these cases is either not detected by the algorithm, or is wrongly detected because it is enlarged with other objects around it. The morphological closing technique is applied to ensure the limitation of this problem. Indeed, the accuracy is improved after we apply it. Secondly, it seems to be difficult to find the hyper-parameters, for the smoothing algorithm as well as the number of cluster k, that works on every images since this is an unsupervised learning technique. However, to be fair this is a paper from 2006. At that time, it is still considered one of the best method for left ventricular segmentation.

D. Tools designed

In order to help normal users who are not familiar with coding, this tool is created with a very friendly GUI allowing them to perform instance segmentation on image in a very simple way. We created this tool using PyQt5 - a free software developed by the British firm Riverbank Computing. Here is a screenshot of our application. The details of how to use it is mentioned below.

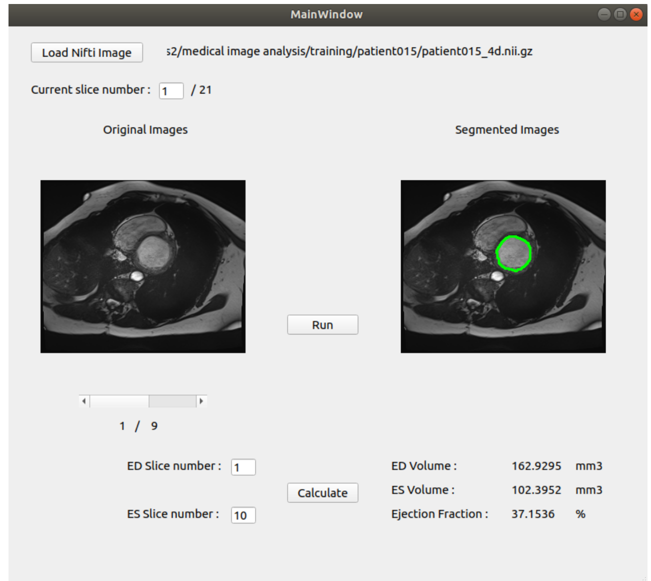

Fig. 12. The graphical user interface of the application.

- Load Nifti image: The button used to browse to a 4D nifti image. Our application works only on this type of file.
- After loading the nifti file, its information regarding number of slices and number of images of each slices as well as the first 2D image of the current slice are displayed on screen. For example, the current nifti file is from patient015. There are total 21 slices in this file. The first image of the first slice is displayed automatically on the screen. The image will be rotated and flipped before displaying to give the best orientation of the heart. Below the image is a horizontal bar help switching between images.
- Run: This button is used to perform segmentation on the current image. Our application allows performing segmentation only on one image at a time. Users need to change the image on the left before continuing segmenting.
- Calculate: This button is used to perform calculation the End Diastolic and End Systolic volume as well as the ejection fraction. It will take 2 inputs: the ED and ES slice number on the left. Users need to specify them before click on calculate button.

V. CONCLUSION

During this project, inspired by [6], a combination of smoothing algorithm [3], K-means algorithm [7], border following algorithm [8] is implemented. The accuracy, as discussed in the previous section, is not very impressive relative to the state of the art Deep Learning technique. In the future, further works need to be done such as improving the accuracy by changing the method to automatically localize the left ventricle, adding more function to the graphical user interface that allows to perform segmentation on every images of one slice at a time, etc. Via this project, we have discovered new image processing techniques like the smoothing and the border following algorithm. They are pretty useful techniques that we will apply them in our next project. This hybrid approach of using multiple algorithm helped us achieve better results.

References

[1] Vector stock: Anatomy of the human heart vector image.

[2] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[3] Ke Chen. A feature preserving adaptive smoothing method for early vision. *National Laboratory of Machine Perception, Peking University, China, Technical report*, 1999.

[4] Jessica S Coogan, Jay D Humphrey, and C Alberto Figueroa. Computational simulations of hemodynamic changes within thoracic, coronary, and cerebral arteries following early wall remodeling in response to distal aortic coarctation. *Biomechanics and modeling in mechanobiology*, 12(1):79–93, 2013.

[5] Hartigan. A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[6] Michael Lynch, Ovidiu Ghita, and Paul F Whelan. Automatic segmentation of the left ventricle cavity and myocardium in mri data. *Computers in biology and medicine*, 36(4):389–407, 2006.

[7] Duda Richard. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[8] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.

Appendix A : Experiment settings

- Python 3.6
- Nibabel
- Numpy
- OpenCV
- PyQt5
- Google Colab