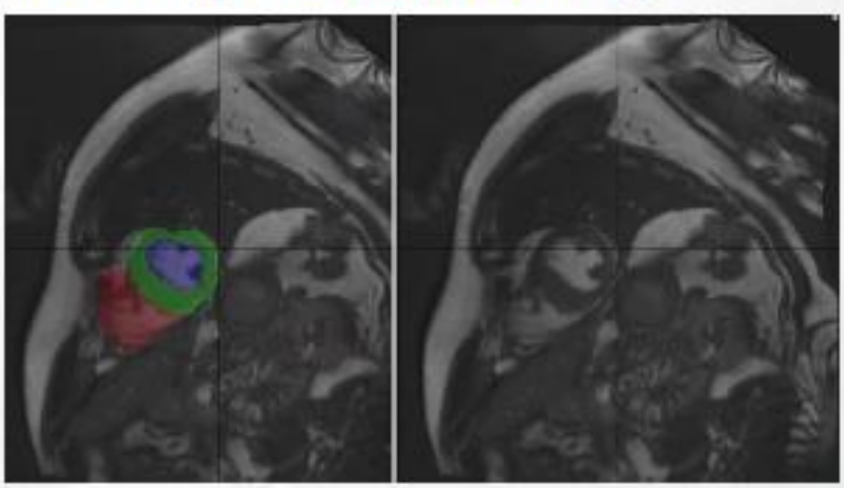
Azadeh Hadadi, Adeyemi Abdulnafiu Olalekan

Automated Cardiac Diagnosis Challenge (ACDC): Left Ventricle Segmentation using K-means Clustering and Deep Learning



22-May-2020

UBFC
UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

LE CREUSOT

2020 MSc. COMPUTER VISION

Reporter (name/surname):
Azadeh Hadadi, Adeyemi Abdulnafiu Olalekan

Module (category):
Medical imaging

Report Tile:
Automated Cardiac Diagnosis Challenge (ACDC)
Sub-title:
Left Ventricle Segmentation using K-means Clustering and Deep Learning

Professor/lecturer/instructor:
Professor Alain Lalande

Date:
22-May-2020

At:
Centre université Condorcet, Le Creusot, France

1. TABLE OF CONTENTS

A) TABLE OF FIGURES

B) LIST OF TABLES

2. ABSTRACT

This research focuses on the automatic cardiac diagnostic challenge (ACDC) and more specifically on left ventricle segmentation in MRI images. Two fully automatic segmentation methods, i.e., k-means and deep learning, will be presented, discussed in detail. Then, implementation of each method will be described and both methods will be unified in a GUI tool for better usability. The tool is a user interface application which developed in Python and wrapped under MATLAB to give better access to the developer as well. The objective of the tool is to segment the left ventricle during diastole and systole (only the endocardium) in MRI images and calculation of ventricle cavity volume. The methods were tested on MRI images of a database made of 100 patients. The result achieved from practical experiments shows significant precision for both methods for the entire database. The deep learning algorithm using TensorFlow backbone was trained several times with different dataset and training settings. The precision of the training for each trial will be discussed and concluded.

3. INTRODUCTION

A) PROBLEM DEFINITION

In this text, following subjects will be covered.

- “Register on the website and download the database(only the training dataset)
  - https://www.creatis.insa-lyon.fr/Challenge/acdc/
- Based on a scientific article (to find), develop tool to segment the left ventricle in diastole and systole only the endocardium)
  - Calculate the ventricular cavity volume in diastole and systole
  - Tips about bibliographic research in the S1 lectures
- Report and software (maximum 6 pages):
  - Little abstract presenting the topic, the context and the background.
  - Describe the used method and the developed tool
  - Discuss the results”

B) BACKGROUND

Cardiovascular diseases cause 17.5 million deaths every year in all around the world [29]. Practically, it has been proven that the early diagnosis of the cardiovascular diseases plays an important role in the recovery. Cardiac MRI image sequences, covering of one full period of cardiac cycle or over several periods, is considered as an important tool for evaluating cardiac function. Evaluation of cardiac function requires calculation of different cardiac parameters (i.e. ejection fraction (EF), left ventricle mass (LVM), left ventricle volume, wall thickness, or wall thickening). All of these parameters can be acquired from segmented endocardium, and epicardial contours of the left ventricle of the MRI image sequences. Traditionally, cardiologists do manual segmentation of these contours in all dataset (i.e. all time frames per all slices) which is a very time consuming task and takes a lot of time and effort. Therefore, the automatic segmentation of left ventricle is very important, and is considered as a challenging task [15]. Several semi-automatic and automatic algorithms were proposed for this problem which will be discussed later (see introduction). Definitely automatic algorithm attracts more attention as naturally it needs less human intervention, although due to more complicated scenarios in practice achieving high precision is usually hard. Recently, repaid development in artificial intelligence (AI) field and especially in deep learning has motivated many researchers to use AI approaches. Due to the same reason, this research will focus on using deep learning for left ventricle segmentation.

C) CONTRIBUTION

In this research, we will set up an easy training dataset generation, training and evaluation procedure which allows not only to be used in the left ventricle segmentation but also can be repeated on other cardiac segmentation (or other organs) and analysis using a Mask-RCCN [16]. The Mask-RCCN employs TensorFlow backbone to train a given model of conventional neural network for detection and segmentation. Different training dataset and parameters lead to different procession. However, the higher precision needs more time which can go beyond few days even. In this work, we developed a tool which simplified data selection and make procedure and evaluation fully automatic with minimum user intervention. Beside, an existing successfully tested CNN architecture will be used to get training weights (gain, bias and so on) which significantly decrease to whole CNN training time. It has been practically proved that our proposed tool will help the user and researchers as well as developers to save a lot time and fairly high precision.

D) REPORT ORGANIZATION

The remaining part of the report is organized as follows. In section 4, literatures related to left ventricle segmentation will be reviewed and summarized. This section will start with left ventricle structure and cardiac MRI image will be followed. Section 5 will be dedicated to selected algorithm and training as well as available development kit. Section 6 will detail implementation, dataset preparation, training, evaluation. Results and the effect of the training parameters as well as comparison with other methods will be explained in section 7. The paper will end up to conclusion and references.

4. RELATED WORKS

E) LITERATURE REVIEW

All algorithms presented for Left Ventricle segmentation can be divided into two categories, i.e., semi-automatic and automatic. Here in this literature only automatic segmentation will be reviewed. Hisham et al. [2] proposed a novel deep learning approach for the automated segmentation and quantification of the LV from cardiac cine MRI images using two consecutive F-CNNs, the first network for localization of ROI and the second network for precise segmentation. Dong et al. [13] uses two-networks architecture as [2] to segment left ventricle directly in 3D. They introduced a new network for the second stage, which called AtlasNet and the paper is all about how to design AtlasNet to get precise 3D left ventricle. Al Noman et al. [3] used parametric active contour model (PACM) to segment Left ventricle. They introduced an artificial neural network based regression model to predict the initial contour to detect the LV area in CMRI image. Kristanto et al. [18] submitted a US patient for LV segmentation by contrast enhancement. Moreno et al. [26] employed a combination of two convolutional neural networks (CNN) to develop a fully automatic LV segmentation method for Short Axis MRI datasets as of [2] but with little difference. The first CNN defines the region of interest (ROI) of the cardiac chambers based on You Only Look Once (YOLO) network. The output of YOLO net is used to filter the image and feed the second CNN, based on UNet network, which segments the myocardium and the blood pool. Wei et al. [33] proposed model leverages channel attention mechanism and shape correction auto-encoder, to adaptively enhanced the feature maps of different receptive field and correct the prediction shape that is inconsistent with the prior s, respectively. While most existing LV segmentation methods focus on cardiac images of single modality or multi-modality, few have been devoted to images of mixed-modality. Cong et al [9] presented a mixed-modality adaptive approached using MRI and CT image of left ventricle for training and testing. Lan et al. [19] presented a deep learning based on double snake segmentation.

Moradi et al. [25] presented a new MFP-Unet CNN for left ventricle segmentation using ultra-sound images and the result sound significantly precise.

Segmentation of Left Ventricle continues to be a challenging task despite significant evolution of techniques and network architectures in last ten years. [21] describes fully automated iterative thresholding method for LV segmentation. [6] talks about a deep convolutional encoder-decoder model seg-net for image segmentation and [32] uses Fully Convolutional Neural Network (FCN) for cardiac image segmentation task. U-Net architecture [25] is another encoder-decoder architecture that has performed well on biomedical image segmentation tasks. In recent years several papers have been published on direct LV volume predictions without segmentation. [24] describes the approach using deep convolutional network and compares the performances of volume prediction using VGG, Google-net and Resnet architectures. Several solutions [12] [7] that were presented during Data Science Bowl 2

Challenge used different algorithms and network architectures to perform LV segmentation and volume prediction.

F) LV STRUCTURE

Left ventricle (LV) cavity's shape is ellipsoid and mayocardium surrounded it which a normal range for its thickness is 6-16 mm. On the other hand, Right ventricle's shape is more complex and also faces lower pressure than LV for ejecting blood to the lungs .Due to these reasons, most of the research efforts are on LV since its function is of greater importance than RV.

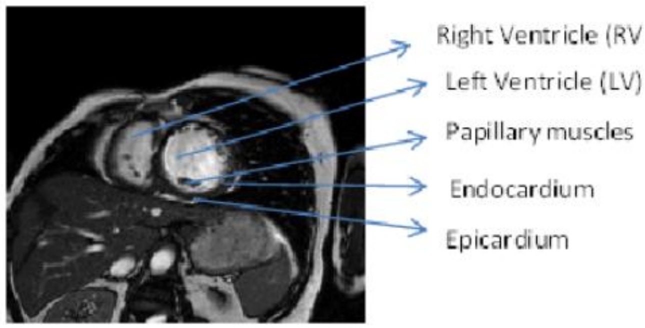


Figure 1: Left ventricle in a short-axis MRI image

The standard imaging plane is perpendicular to the long (apex-base) axis and called short axis plane as is shown in Figure 1. Imaging of the heart in MRI covers the whole organ with about 8–10 short-axis slices, distance between two adjacent slices ranging from 10 to 20 mm. Normal MR image demonstrate blood pools as bright whereas myocardium and surrounding structures are shown in dark. Heart segmentation of MR images consist of outer wall called epicardium and inner wall called endocardium. The epicardial wall is at the frontier between the myocardium and surrounding tissues (fat, lung), which does not show a good contrast with the myocardium. On the other hand, for endocardium which surrounds the LV cavity, MRI demonstrates quite good contrast between myocardium and the blood flow even without contrast medium. in overall, since endocardium is less difficult to be segmented than the other one and also for computing ventricular volume, it is the only contour which is required, most of the works focus only on that.

G) CARDIAC MRI

MAGNETIC RESONANCE IMAGING has unique features for accurate quantification of anatomy; function, flow, and perfusion of the cardiovascular system at rest and under stress conditions. Using multi-slice cine MRI techniques, the three-dimensional (3D) geometry of the heart can be imaged at high temporal and spatial resolution. As a 3D tomographic imaging technique, volumetric measurements do not rely on any geometrical assumption and therefore are accurate for both normal and abnormal ventricular anatomies; both left and right ventricular dimensions can be assessed with small margins of error. Several MRI techniques can be applied to assess regional ventricular function quantitatively such as:

- Cine MRI using short-axis slices may be used to assess wall motion and wall thickening. With the use of MRI tagging, cine MR images are acquired with a superimposed rectangular or radial grid of dark saturation lines.
- Velocity-encoded cine MR imaging (VEC-MRI), also termed MR flow velocity mapping, is another MRI technique that provides information on blood flow or myocardial motion over the cardiac cycle. For blood flow measurements, an imaging plane perpendicular to the expected flow direction has to be selected; then velocities in the through-plane direction are obtained for each pixel within the image at multiple points in the cardiac cycle.
- Paramagnetic contrast agents such as gadolinium-diethylene triamine pentaacetic acid (Gd-DTPA) can be used in combination with static MRI techniques to depict myocardial regions with increased or decreased uptake of the agent, related to myocardial infarction [29].

5. ALGORITHM AND DEVELOPMENT

There are several techniques used in the left ventricle segmentation. In general, we can classify these techniques according to the algorithm that used in the segmentation process to, image-based methods (i.e. which include thresholding, region growing, etc), pixel classification methods (i.e. using clustering, and classification methods), deformable models (i.e. like active contour model), model-based methods (i.e. active shape model (ASM), and active appearance model (AAM)), and finally the atlas guided method [15].

Another view, we can consider the different techniques to segment left ventricle as is shown in Figure 2. In the target tool, Automatic Technique through K-means and Deep Learning was used to segment the left ventricle from cardiac MRI.

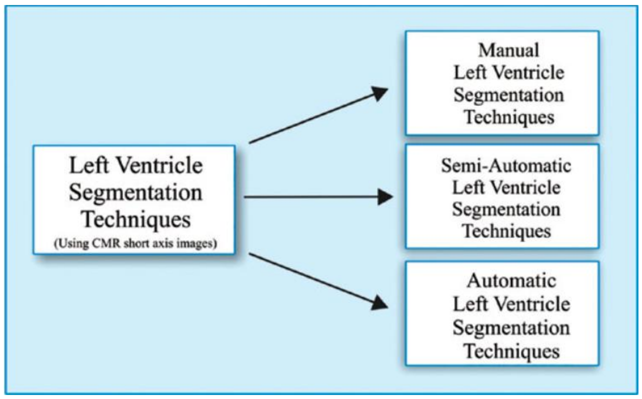


Figure 2: Techniques for LV segmentation

A) MASK R-CNN

The methodology to implement the left ventricle segmentation is Mask R-CNN. This technique is simple, flexible, and a general framework for object instance segmentation [16]. This approach efficiently detects objects in an image while is simultaneously generating a high-quality segmentation mask for each instance. This framework extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. The mask branch is a small FCN applied to each ROI, predicting a segmentation mask in a pixel-to pixel manner. Figure 3 shows some results of Mask R-CNN.

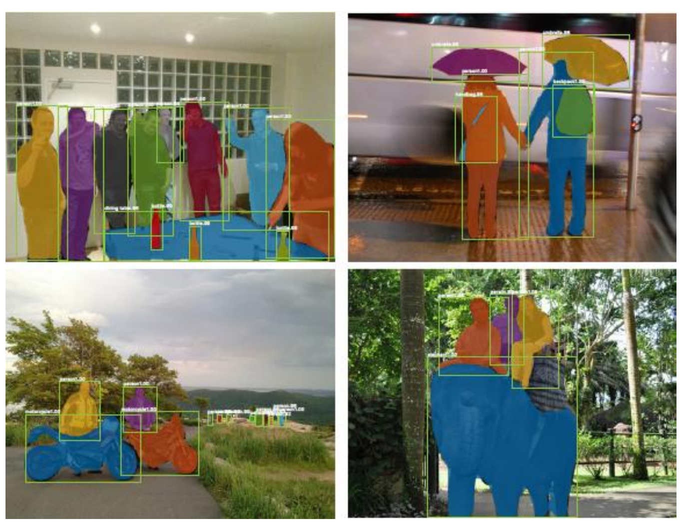


Figure 3: Mask R-CNN results on the COCO test set. Masks are shown in color, and bounding box, category, and confidences are also shown

Although Fast/Faster RCNN [14] [23] and Fully Convolutional Network (FCN) [23] are frameworks for object detection and semantic segmentation, respectively, Mask R-CNN is an enable framework for ***instance segmentation*** which is a challenge because it requires the correct detection of all objects in an image while also precisely segmenting each instance. It therefore combines elements from the classical computer vision tasks of object detection, where the goal is to classify individual objects and localize each using a bounding box, and ***semantic segmentation***, where the goal is to classify each pixel into a fixed set of categories without differentiating object instances.

The output of each candidate object in Mask R-CNN includes:

- A class label (Faster R-CNN)
- A bounding- box offset (Faster R-CNN)
- additional mask output which is distinct from the class and box outputs, requiring extraction of much finer spatial layout of an object.(* ***Mask R-CNN***)

Mask R-CNN has two stages:

1. Region Proposal Network (RPN) which proposes candidate object bounding boxes
2. predicting the class and box offset ***in parallel*** (Mask R-CNN also outputs a binary mask for each ROI)

The multi-task loss on each sampled ROI during training is defined as:

$$L = L_{cls} + L_{box} + L_{mask}$$

The mask branch has a Km^2-dimensional output for each ROI, which encodes K binary masks of resolution $m \times m$, one for each of the K classes.

To this a per-pixel sigmoid is applied, and define L_{mask} as the average binary cross-entropy loss. For an ROI associated with ground-truth class k, L_{mask} is only defined on the k-th mask (other mask outputs do not contribute to the loss). So different masks can be generated by the network for every class without competition among classes.

In the presentation of mask, an input object's spatial layout is encoded by the mask. So extracting the spatial structure of masks can be addressed naturally by the pixel-to-pixel correspondence provided by convolutions. This pixel-to-pixel behavior requires our ROI features, which they are small feature maps, to be well aligned to faithfully preserve the explicit per-pixel spatial correspondence.

B) TRAINING

As in Fast R-CNN, an RIO is considered positive if it has IoU with a ground-truth box of at least 0.5 and negative otherwise. The mask loss L_{mask} is defined only on positive ROIs. The mask target is the intersection between an ROI and its associated ground-truth mask. Images are resized such that their scale (shorter edge) is 800 pixels.

Each mini-batch has 2 images per GPU and each image has N sampled ROIs, with a ratio of 1:3 of positive to negatives. N is 64 for the C4 backbone.

C) INFERENCE

At test time, the proposal number is 300 for the C4 backbone and 1000 for FPN. We run the box prediction branch on these proposals, followed by non-maximum suppression. The mask branch is then applied to the highest scoring 100 detection boxes. Although this differs from the parallel computation used in training, it speeds up inference and improves accuracy (due to the use of fewer, more accurate ROIs). The mask branch can predict K masks per ROI, but we only use the k-th mask, where k is the predicted class by the classification branch. The $m \times m$ floating-number mask output is then resized to the ROI size, and binarized at a threshold of 0.5.

6. IMPLEMETATION

Several tools of left ventricle segmentation have already been presented and shared with other researchers for development and further exploration. Chen et al. [8] presented a source code for semi-automatic segmentation using active counter without edge. The open source implementation in MATLAB is accessible and can be downloaded from their GitHub page [5].Theory and mathematical detail behind their implementation is described in [28]. This job later was extended and integrated with another deep learning implementation. The complete source code is available in this link [4]. A part of this work has been used to test semi-automatic approach just to see the prose and crone of semi-automatic approach.

Julian Dewit et al [11] presented another python implementation for left ventricle segmentation by deep learning, which works on GPU. As we are quite limited in terms of hardware this work was referred to as general guideline only and the code was not used. Further detail of the background and theory behind this implementation presented in [10]. Another open source code was presented by Abdelmaguide et al. [1], which developed based on deep learning in python. The code uses TensorFlow backbone for training and evaluation and the detail of the CNN model presented in [1]. This development is highly depending on code published by De wit et al [11].

Our development is not relaying on this code and no part of their code was used, but their handout is a good source to get an idea where to start the development and how to establish the training and testing framework. Woshi et al. [30] has published another source code developed in python. The detail of the development and code can be read in [31]. None of the above work has been completely integrated in our development nor partially used. Our development was more motivated by approach explained by Khush Patel [27]. Patel explains how to perform object detection using simple example, this helped us to understand how to setup TensorFlow and used it for new dataset training and evaluation. The code primarily developed to detect the ROI for left ventricle. Then this code extended to be trained to segment certain object using mask. More specifically, different masks were extracted from training data set and the CNN network was trained and tested for left ventricle segmentation.

A) K-MEANS SEGMENTATION

The first approach is automatic Left Ventricle Segmentation of the heart algorithm using the K-means clustering and graph-searching on cardiac MRI [20]. Since Clustering is a certain similarity criteria to divide the data into groups having similar properties of subclasses, and internal similarities greater than the similarity between such.
Measurement of distance or similarity is the basis of clustering algorithms. K-means clustering algorithm is one of the most common iterative operators with adjustment K-means center of mass. K-means is a clustering method based on squared error, is also very famous hard clustering algorithm. The algorithm is simple and fast, and as with other clustering algorithms, K-means is an iterative optimization process in most of the slices, the LV has the highest intensity value. With this information, we assume to estimate the segmentation to extract the region of the endocardium. Also, since it is dealing with polar structure of short axis MRI and the intensity of the endocardium will give us information towards segmentation. In order to achieve the segmentation, we presented our algorithm in 2 different phases:

1. THE PREPROCESSING PHASE

This part consists of taking the sum of the patient like patient 1 folder. We calculate the seed points and initial LV information such as center of gravity and intensity statistics, we Compute K-means clustering of pixels intensity values then the Left Ventricle is segmented by K-means selecting the correct value of K. At the end, perform erosion to extract the region of the left ventricle.

2. THE PROCESSING PHASE

Processing: we applied the triangulation to align each segment image on the slices to the center of the reference segmentation image performed lastly in preprocessing phase. Then, segmented error is corrected by graph search. Here the idea is to visit every node exactly once. Observe the diagram below to understand the processing phase:

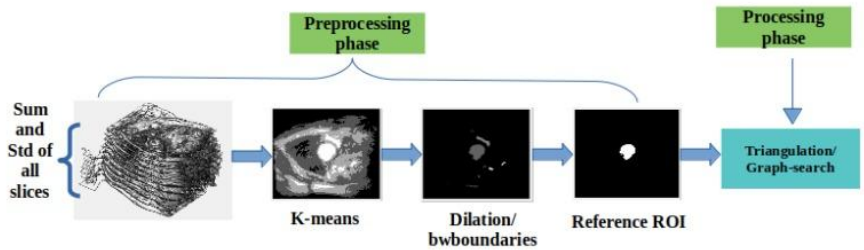


Figure 4: K-means process

We know that K-means is an exploratory analysis technique and also implements a non hierarchical method of grouping of objects together which means it just takes datasets as they come in and then group them together and also uses the centroid to Euclidean method to calculate distance and groups objects based on the minimum distance.

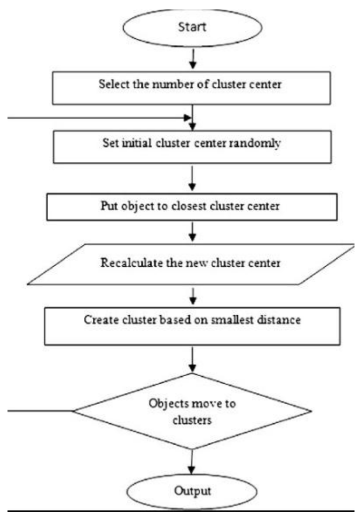


Figure 5: K-means process flow

In this paper, we give the method to determine the initial centre in the previous sample based on streamline, the maximum minimum distance method is introduced into density as a measure, the initial clustering centre K choose sample K features of highest density sample, then, iteration to determine the initial clustering centre.

Problem: The data set $X = (x_1, x_2, \dots, x_n)$

is divided into classes, making the error sum E of squares of the clustering is minimal.

Also: The data object $(x^1, x^2, \dots, x^p)$ and $(y^1, y^2, \dots, y^p)$

The distance between them is: $d(x, y) = \sqrt{(x^1 - y^1)^2 + (x^2 - y^2)^2 + \dots + (x^p - y^p)^2}$

B) DEEP LEARNING SEGMENTATION USING TENSORFLOW

The second approach is based on MASK-RCNN framework, which was established and tested in [16]. The code was first developed in python and for simplicity, usability, easier development and better visualization was wrapped under MATLAB code. Three main items of the segmentation network will be detailed, i.e., CNN network, ROI definition to left ventricle localization, different masks used for left ventricle segmentation, and training setting.

3. RCNN NETWORK ARCHITECTURE

Feature Pyramid Network (FPN) is a backbone architecture which was proposed by Lin et al. [22]. FPN uses top-down architecture with lateral connections to build an in-network feature pyramid from a single-scale input. Faster R-CNN with an FPN backbone extracts ROI features from different levels of the feature pyramid according to their scale. Using a ResNet-FPN backbone for feature extraction with Mask RCNN gives excellent gains in both accuracy and speed.

Also, the network ***head*** will be followed to which a fully convolutional mask prediction branch will be added. Details are shown in Figure 6. The head on the ResNet-C4 backbone includes the 5-th stage of ResNet (namely, the 9-layer 'res5' [17]), which is compute-intensive. For FPN, the backbone already includes res5 and thus allows for a more efficient head that uses fewer filters. Mask branches have a straightforward structure.

In the Mask R-CNN two existing Faster RCNN heads are extend. In Figure 6, Left/Right panels show the heads for the ResNet C4 and FPN backbones, from and, respectively, to which a mask branch is added. Numbers denote spatial resolution and channels. Arrows denote either conv, deconv, or fc layers as can be inferred from context (conv preserves spatial dimension while deconv increases it).

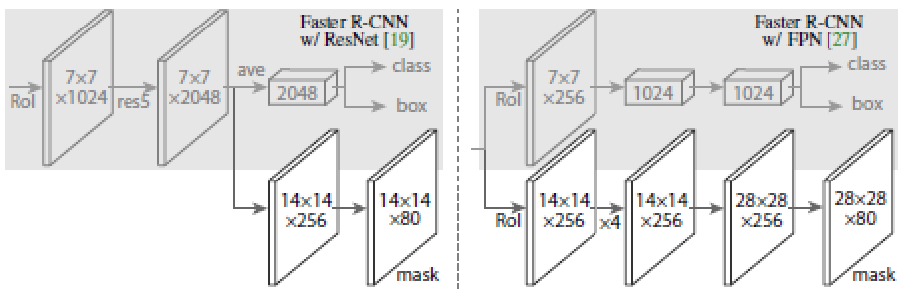


Figure 6: Head Architecture

As of other CNN network presented above and in related literature, our network has two main part, convolution layers (feature extraction) and neural network for classification. Two networks were used consecutively for ROI localization and segmentation.

4. ROI DETECTION

The first RCNN network detects the ROI. In fact, the ROI is selected for each image and stored in an *.XML file when the training data is prepared. The ROI is selected as rectangular and defined by six parameters $[x_{min}, y_{min}, x_{max}, y_{max}, w, h]$. Coordinate (x_{min}, y_{min}) defines the left top corner of the rectangle which defines ROI and w, h defines width and height of the rectangle. These data are stored for each image in a file with same name as training image.

5. SEGMENTATION MASK

The mask is used to segment left ventricle, which has already localized in ROI by the first RCNN. Different masks were created and used during the training. There are several possibilities to select the masks, either manually (polygonal, elliptical or free hand) or selected semi-automatically using snake algorithm [34] or any other precise segmentation algorithm. Practical experiments reveal the fact that creating mask manually gives rise to better and more precise result. 3 types of segmentation mask used in our implementation are shown in Figure 7.

Figure 7: (a) elliptical mask generated manually (b, c) mask generated automatically by snake [34] (d) mask generated manually using polygon method

6. TRAINING SETTING

The training parameters were setup as of Table 1. After setting the training dataset and training parameters and running the training procedure a *.h file which contains the different weighting values (bias, ANN weight and so on). This weight usually is used to test and validate the training result and CNN architecture for ROI detection and left ventricle segmentation.

Table 1:CCN network training parameters

BACKBONE	Resnet101	LEARNING_MOMENTUM	0.9
BACKBONE_STRIDES	[4,8,16,32,64]	WEIGHT_DECAY	0.0001
IMAGE_SHAPE	[1024 1024 3]	BATCH_SIZE	2
IMAGE_RESIZE_MODE	Square	COMPUTE_BACKBONE_SHAPE	None
IMAGE_MIN_SCALE	0	DETECTION_MAX_INSTANCES	100
IMAGE_MIN_DIM	800	DETECTION_MIN_CONFIDENCE	0.7
IMAGE_META_SIZE	14	DETECTION_NMS_THRESHOLD	0.3
IMAGE_MAX_DIM	1024	FPN_CLASSIF_FC_LAYERS_SIZE	1024
IMAGE_CHANNEL_COUNT	3	GPU_COUNT	1
GRADIENT_CLIP_NORM	5.0	IMAGES_PER_GPU	2
LEARNING_RATE	0.001	BBOX_STD_DEV	[0.1 0.1 0.2 0.2]
MASK_POOL_SIZE	14	MAX_GT_INSTANCES	100
MASK_SHAPE	[28, 28]	MEAN_PIXEL	[123.7,116.8,103.9]
MINI_MASK_SHAPE	(56, 56)	NUM_CLASSES	2
NAME	left_ventricle_cfg	POOL_SIZE	7
POST_NMS_ROIS_INFERENCE	1000	POST_NMS_ROIS_TRAINING	2000
PRE_NMS_LIMIT	6000	ROI_POSITIVE_RATIO	0.33
RPN_ANCHOR_RATIOS	[0.5, 1, 2]	RPN_ANCHOR_STRIDE	1
RPN_ANCHOR_SCALES	(32,64,128,256,512)	RPN_BBOX_STD_DEV	[0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD	0.7	STEPS_PER_EPOCH	130
RPN_TRAIN_ANCHORS_PER_I MAGE	256	TRAIN_ROIS_PER_IMAGE	200
USE_RPN_ROIS	True	TOP_DOWN_PYRAMID_SIZE	256
VALIDATION_STEPS	50	TRAIN_BN/USE_MINI_MASK	False/True
LOSS_WEIGHTS	{rpn_class_loss:1.0 ,rpn_bbox_loss:1.0,	,mrcnn_class_loss: 1.0 ,mrcnn_bbox_loss': 1.0,	mrcnn_mask_loss:1.0 }

C) CALCULATION

- With the segmented part, we can estimate the cavity volume and ejection Fraction EF. We took the slice thickness S_t = 5 and S_g = 5 to compute the volume cavity as:

$$V = (S_g + S_t) * \sum_{i=1}^{n} A_i$$

- Left Ventricle Mass LVM and Ejection fraction EF are defined as in the following equations:

$$LVM = [VED_{epi} - VED_{endo}] * 1.05$$

$$EF = \frac{EDV - ESV}{EDV} * 100\%$$

Note: VED_{epi} and VED_{endo} represent epi and endo cordial volumes in the end Diastole while EDV and ESV represent endocardial volume in the end diastole and systole volume phase.

7. PROCEDURE OF USING SOFTWARE PACKAGE

The target tool is developed in Python 3.7 and MATLAB2019. The python code was wrapped under MATALB for easier setting up the procedure and visualization. Besides, further development will be easier for those are quite new with the topic and want to contribute having dealt only with MATLB. The main menu is shown in **Error! Reference source not found.**. Main menu is made of two main functionality sections which give access to two segmentation algorithms. The user can select ***Segmentation Methodology*** by clicking on desire radio buttons on top of the menus; either K-means or Deep Learning. In the middle of the menu the result will be displayed after performing each action. At the right side of the menu the final result either in the form of video or photos will be displayed.

MATLAB code is used to implement all of the K-means and Deep Learning parts, except training and evaluation parts which are developed by Python and are wrapped under MATLAB.

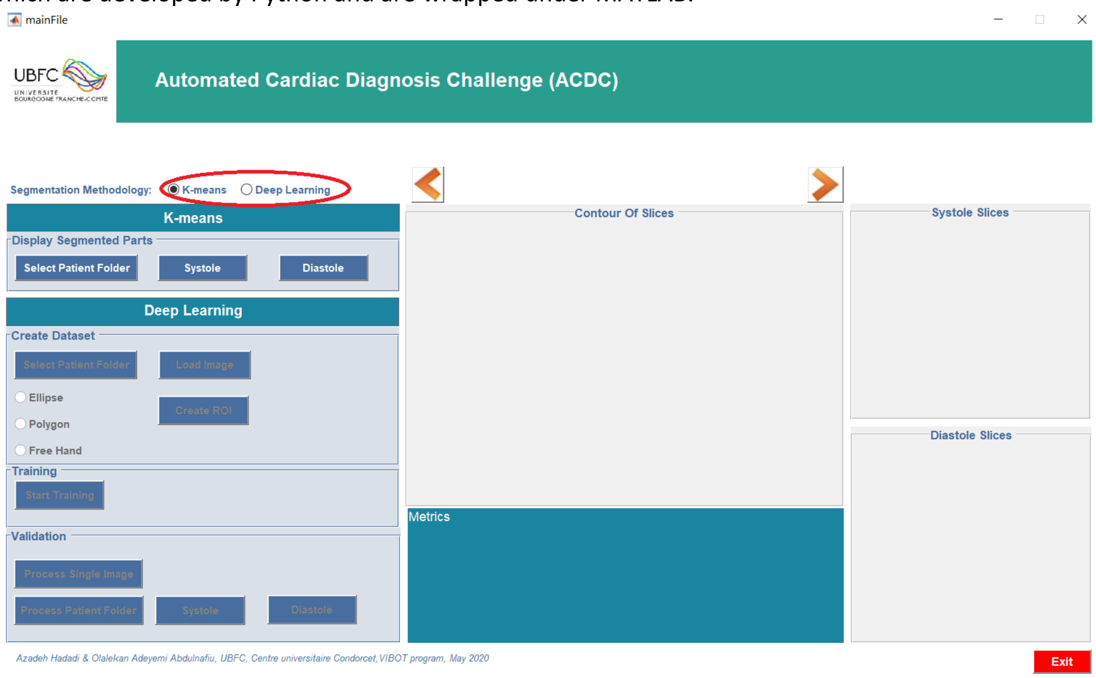
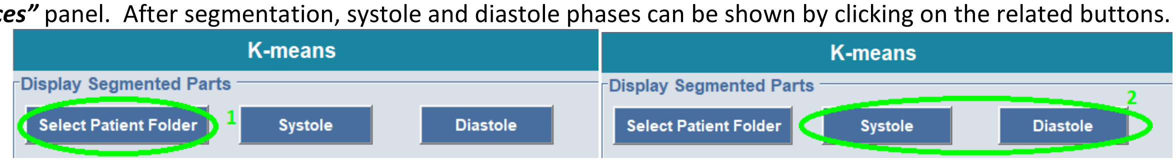


Figure 8: main menu (left: access to functions, algorithms and processing, middle: display images, right screen: result)

By selecting each of methodology (radio button), the other methodology section will be disabled. K-means has one menu named *Display Segmented Parts***. One patient's folder is selected by pushing on** *Select Patient Folder* **button as is shown in**

Figure 9(a). Then, all of the image systole and diastole slices will be extracted and segmented. Metrics, i.e. End-Systolic, End-Diastolic Volume and Ejection Fraction are calculated and result will be shown under "***Contour of Slices"*** panel. After segmentation, systole and diastole phases can be shown by clicking on the related buttons.



(a) (b)

Figure 9: select of a patient's folder (a) systole and diastole phases (b)

Deep learning section is made of three panels:

1. ***Create Dataset***
2. ***Training***
3. ***Validation***

The ***Create Dataset*** panel prepares training data for Mask R-CNN which gives access to two main functionalities:

1. Read NIFTI files (by clicking on ***Select Patient Folder*** button)
2. Generating masks (select one of the radio buttons ***Ellipse, Polygon or Free Hand*** and then click on the ***Create ROI*** button)

Left-Ventricle-Database folder in the project root contains a dataset for Mask R-CNN. It includes three folders:

1. annots
2. images
3. masks

These folders will be filled automatically after creating mask by the user.
The first step is reading patient NIFTI files pushing ***Select Patient Folder*** and choosing one patient folder in the main training dataset. The selected NIFTI files will be read and converted to jpg images and saved in ***Left-Ventricle-Database/images*** folder as illustrated in Figure 10. A successful completion of this step is announced by a message on the top of panels as shown in Figure 11.

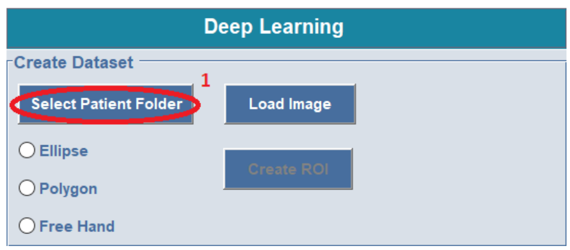


Figure 10: read and convert NIFTI files from selected patient folder

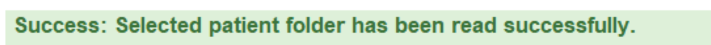


Figure 11: successful message of converting NIFTI files

Next step is to load image by pushing ***Load Image*** button and selecting jpg files which were generated in the previous step (Figure 12 (a)). Then, selecting images will display the resulted image in the ***Contour of Slices*** panel (on the right) and can be navigated pushing Next & Previous buttons on the top of this section (Figure 12 (b)).

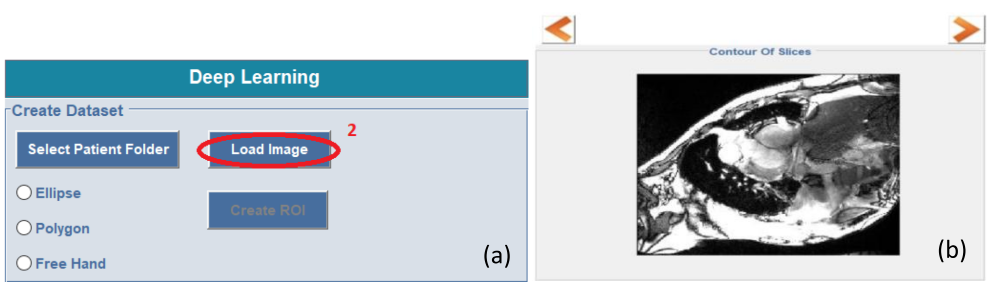


Figure 12: Load image action (a) contour of slices part to show loaded images (b)

Then, masks should be created manually. To do that, there are three approaches, including ***Ellipse***, ***Polygon*** and ***Free Hand***. This way, it is very flexible to select different shapes of ROI. Before choosing one of these radios, ***Create ROI*** button is disabled and as soon as one of them is selected, it will be enabled. In Figure 13, selected ROI by the three approaches are illustrated.

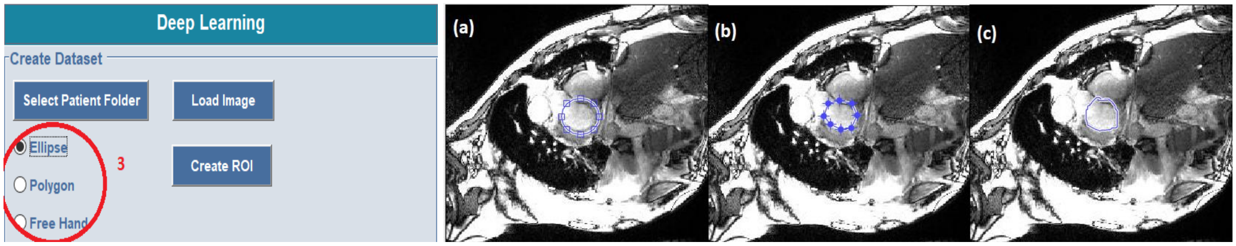


Figure 13: select ROI by Ellipse (a) Polygon (b) Free Hand (c)

After selecting ROI, ***Create ROI*** button shall be pushed as is shown in Figure 14, to create mask and annotation xml files in ***Left-Ventricle-Database/masks*** and ***Left-Ventricle-Database/annots*** respectively.

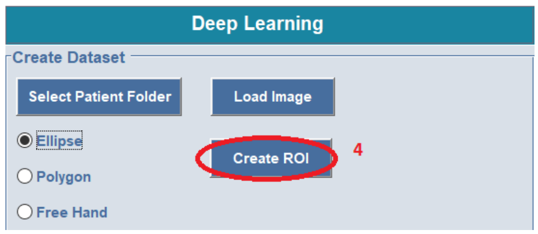


Figure 14: create ROI

Both created mask and xml files have the same name as the selected image. The xml file will be produced automatically and contains the ROI coordination and its width and height as described in section 6.B (for more detail of the content see Figure 15 (a)). The mask is binarized ROI like in Figure 15 (b).

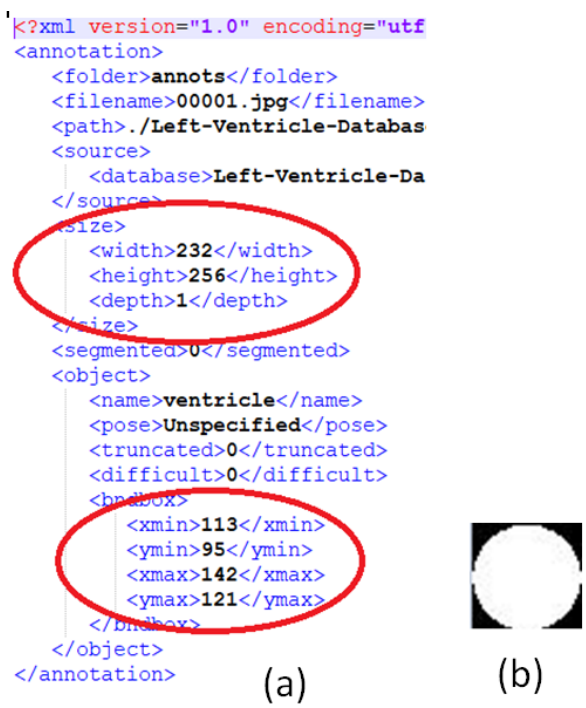


Figure 15: created xml file (a) created mask (b)

Described procedure is repeated for some patients to complete Deep Learning dataset. Masks and annotations are created for all the jpg files in ***Left-Ventricle-Database/images*** path. Then, Mask R-CNN is ready to train by pushing ***Start Training*** button as shown in Figure 16. This process takes from one to few hours to be completed depending on the size of the training data and number of Epoch selected. The size of the selected data and processing time will be discussed in the following sections (see result and discussion for further detail). Finally, during each complete training cycle five training weighting files will be created if number of epoch is set to five which can be used to do validation. Usually the best training weight is the last one as it has the lowest evaluation lost.

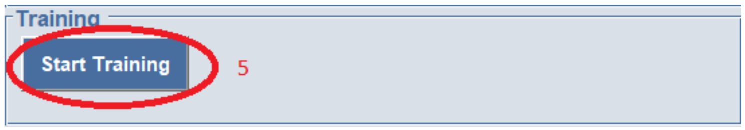


Figure 16: start training Mask R-CNN

The last panel is validating test images. There are two options: ***Process Single Image*** and ***Process Patient Folder***. Pushing ***Process Single Image*** button gives us opportunity to segment a single jpg image as shown in Figure 17.

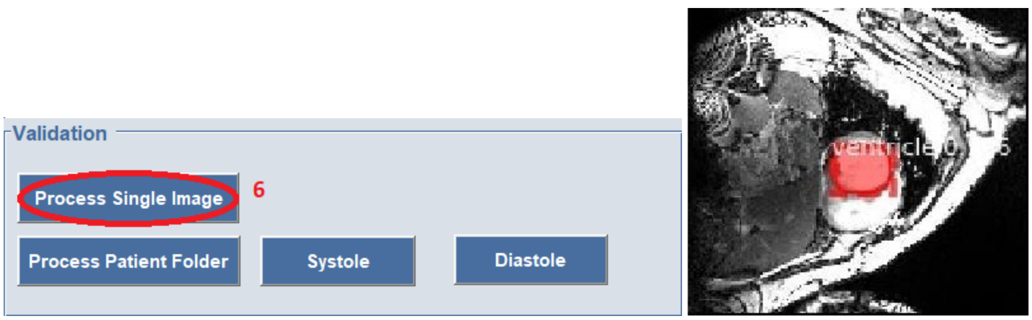


Figure 17: process single image

To segment all MRI images belong to a patient and calculate volume and ejection fraction, click on ***Process Patient Folder*** button and select the patient's folder. This will take few minutes to segment all systolic and diastolic slices. A progress bar is displayed to the user during time processing. After finishing the validation, related metrics are shown and systole and diastole phase images can be visualized in their related panels on the right by pushing ***Systole*** and ***Diastole*** buttons.

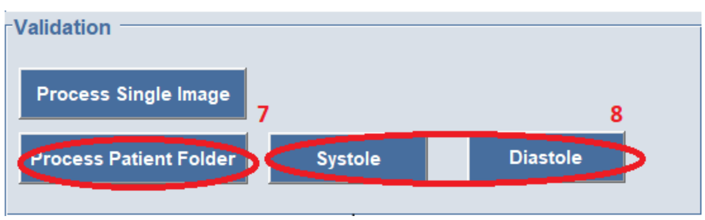


Figure 18: process all of a patient's slices and show systole & diastole phases

Finally we can exit the application by clicking the ***Exit*** button.

8. RESULT AND DISCUSSION

In this section, result of the above methodologies will be presented and detailed. K-means Clustering segmentation was done in the endocardium which produces a good result and the volume calculation of diastole and systole for patient2 as shown in the GUI image attached below:

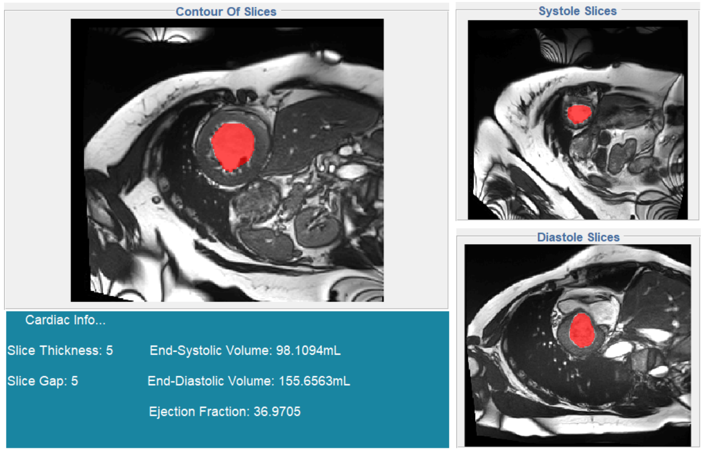


Figure 19: K-means Clustering result sample

However, the same result might not be generated for all patients in this method because this method is prone to failure under more complex scenarios such as intensity variation, imaging artefacts, surrounding argons with intersection as left ventricle in the same slices. The blockage of left ventricle due to cardio-vascular disease in the patient's image might create even more problem and leads to segmentation failure in this method. To overcome this problem we came up with other methods implemented using deep learning which has already been explained in more detail in the previous section. Here blow, first, we will have a furtive glance over the method results and then we will pass to the comparison of the result.

In Deep Learning methodology, by selecting a patient folder in ***Create Dataset*** panel, systole and diastole slices are read and converted to jpg files as is shown in Figure 20.

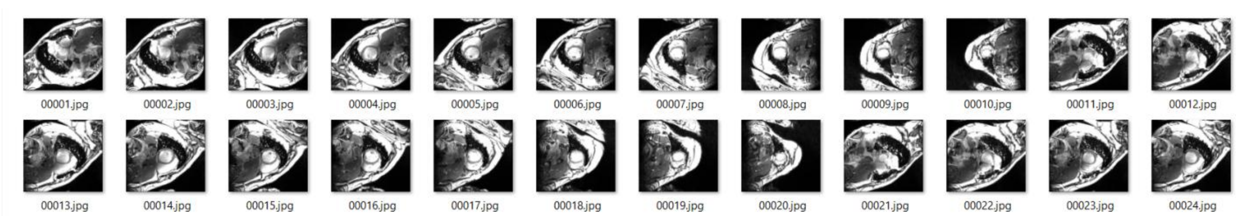


Figure 20: converted NIFTI files to jpg images

These images of the patient is as shown in Figure 21 which can be navigate back and forth slice by slice as shown in this figure and detail in section 7. The training and validation and testing procedure also were described in the same section.

Figure 21: loaded images

Now, we will have a look at some results achieved using by this method. The result of the segmentation of the left ventricle during diastole and systole is very similar to those have been generated by K-means algorithm. We keep the same display and visualization interface meaning the result will appear in the same panel as of K-means. One sample of original image and its segmented result is shown in Figure 22 (a) and (b) respectively. In this method, the segmented region is highlighted in red in Figure 22. Sometimes, the algorithm might find several ROI by the first network. Naturally in this case, somehow, the algorithm shall determine which region is the most probable candidate for left ventricle. For this purpose, the algorithm calculates a score for each segmented region. Definitely the highest score belongs to the most probable region of interest. For instance for a given image score for left ventricle is 0.996. Practical experiment reveals the fact that in 95% of the time only one region is detected and segmented. However, in 5% reaming cases not more than two regions are detected and segmented and when the score is considered as an additional criteria the correct region can be selected which is 98% of time correct candidate for left ventricle.

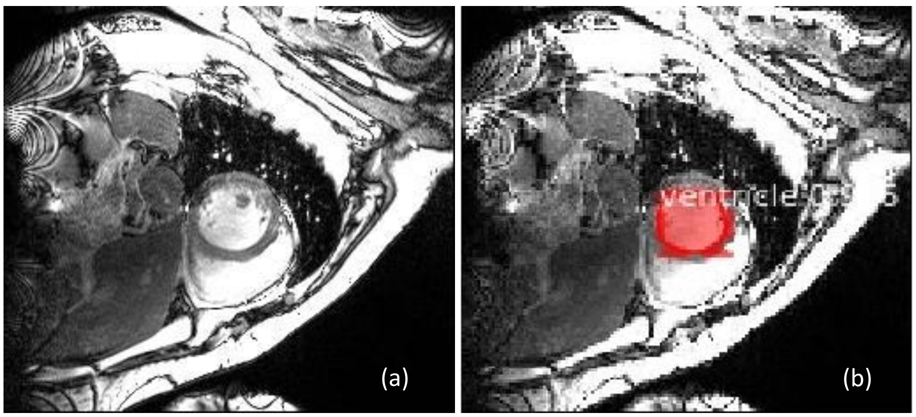


Figure 22: original single image (a) its segmented result by Deep Learning (b)

Each NIFTI file contains several images for per patient. When we select the patient folder and start segmentation all images were selected and segmented for diastole and systole. Finally, the patient's left ventricle is segmented by Deep Learning algorithm and volume for systolic and diastolic and ejection fraction metrics for slices are displayed as shown in Figure 23. The quality of the segmentation is highly depends on the selected training dataset and complexity of the scenarios involved in the images, the more complexity the more precision.

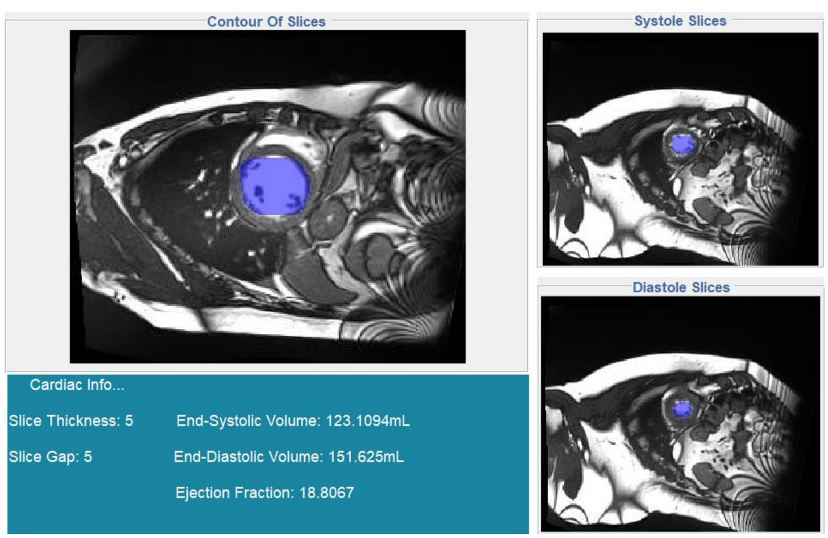


Figure 23: Deep Learning result sample

De wit et al. [10] presented some result achieved by deep learning using very similar approach as presented in this work. However the approach seems not work properly when intensity is changing. For that reason De wit teams uses a pre-processing step to improve the intensity mostly by applying intensity enhancement or histogram equalization and other methods which improves the segmentation results consequently. The processing has been advised by different researcher. However, we preceded different approach to resolve this issue by selecting proper training dataset and avoiding time-consuming pre-processing and image enhancement algorithm. In fact, we have selected training images from different patients with more verities of intensity and other image diversities when making the training dataset. It seems quite effective approached and leads to better result in practice although this approach necessarily will leads to longer training time and bigger training database. The practical experiment conducted on the unknown images shows the segmentation result is promising.

To investigate this approach more and to study the effect of the training dataset, image diversity and training time as well as accuracy of the segmentation, we have selected thee dataset, i.e., 20, 80, 300 MRI images. We setup training for each datasets and ran the training process as detail under section 7 with 5 epochs of training consecutively non-stop and in an automatic manner and at each epoch we perform 130 steps of trail for training. The number of trail for dataset with 20 images was 80 trials per epoch. The training weight was validated and tested but the results was very good because first of all several candidates were detected and segmented and the segmentation error was quite high. So this dataset was kept aside and only dataset with 80 and 300 images were considered. The training of both dataset was performed with 5 epochs and 130 trails per epoch. The training loss for both dataset for epoch 1 is shown in Figure 24. As seen, the training loss is quite significant at early trial and it decreases exponentially when the trail increases. The training loss for dataset with 300 images is most of the time lower than dataset with 80 images. This difference is consequently affecting the validation loss as shown in Figure 26. In each dataset, 20% of the images are selected for validation which was not involved in training process and remaining 80% is used for training only.

Figure 24: training loss of epoch 1 for dataset of 80 and 300 images

The training loss of remaining epochs (epoch 2-5) is shown in Figure 25. As seen inversely, all training loss for the remaining epochs are higher for dataset with 80 images while we expect vice versa. However, this discrepancy in the training loss does not affect the validation loss because of the diversity involved in the training as mentioned above. We observed that in early trails the losses increase and reach to a local pick and after exponentially decrease which show almost correct training curve as we expect. Another observation comparing between training with 80 images versus 300 images is that during the validation and testing process the number of candidate detected by the weight extracted from 80 images for left ventricle is more than one sometimes. Besides, the higher score might be allocated to the region which is not really true ROI as expected. This is not the case when weight extract from training which uses 300 images instead. In this case, always only one ROI for left ventricle is detected and segmented. In a very rare case, especially at the beginning of diastole, because of small area of left ventricle the detected ROI are not completely segmented. This is not a failure because even with the bare eye it is hard to recognize left ventricle in this scenario and needs more experience. Sometimes the same issues are observed for systole.

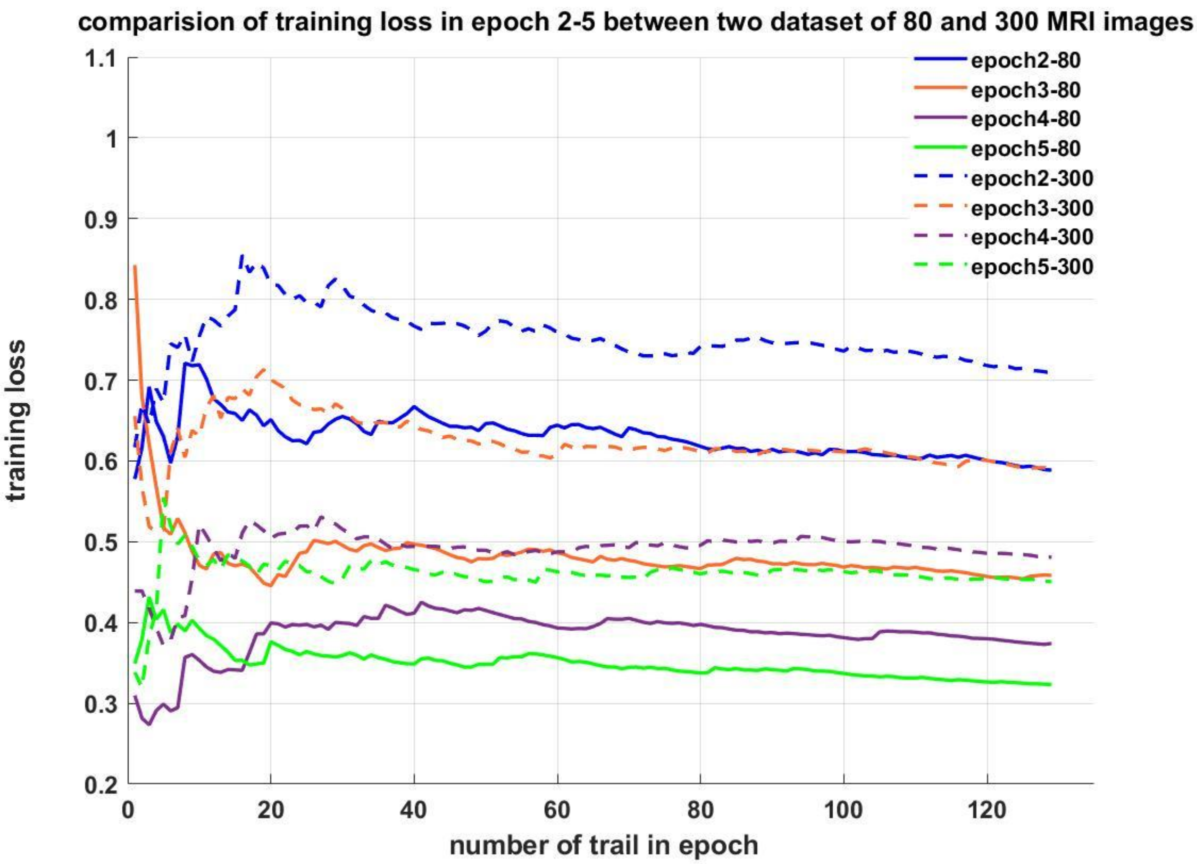


Figure 25: training loss of epoch 2-5 for both dataset with 80 and 300 images

Looking into the detail provided in Figure 26, validation loss, it is clearly visible that losses for segmentation with resulted weight of 300 images are smaller than weight extracted from dataset of 80 images. This difference for all epochs is significant and the difference increases when the training process steps into the higher epoch. The difference of the losses for epoch 5 is almost 50% which consequently leads to very high precision for the weight calculated in epoch 5. This level of precision naturally needs more computation time as shown in Figure 27. In this experiment the training executed using general purpose CPU with multiple processing cores and the computation time when the number of the images increases is quite significant (half a day up to a day). For instance, as seen in Figure 27, it took almost half a day to complete one training epoch when 300 images were used as training dataset. The overall times spend on training using dataset with 300 image roughly is two times than the training of dataset with 80 images as shown in Figure 27 which leads to two times more precision as seen in Figure 26.

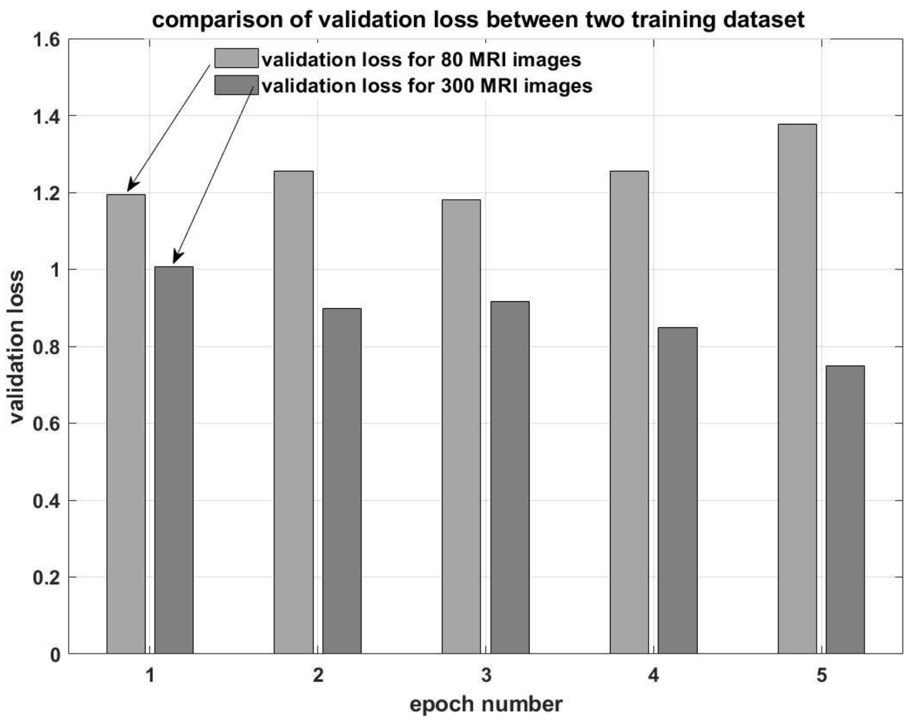


Figure 26: validation error of all epoch for both dataset of 80 and 300 images

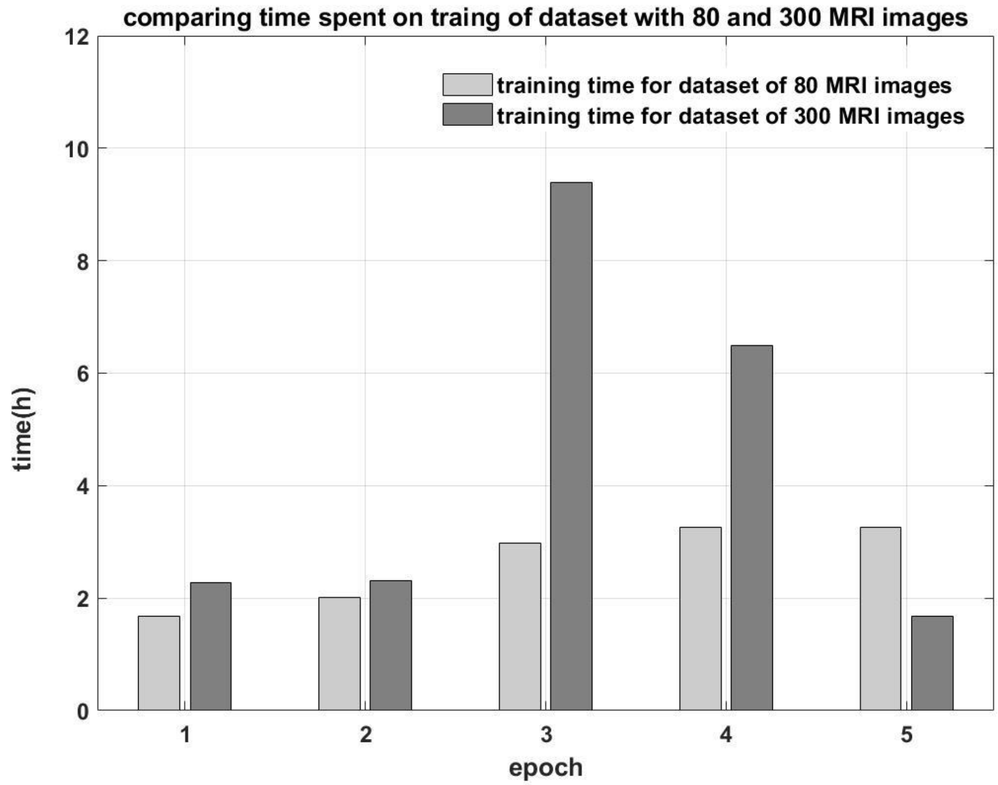


Figure 27: computation time difference of training for dataset of 80 and 300 images

It was recommended by the TensorFlow developer to use cluster GPU cards to accelerate the training process. We tested GPU Tensorflow and noticed always memory overflow just as a general rule to run this training on GPU to save the time, the target GPU should contain more than 12GB of internal RAM. Otherwise, the GPU implementation will fail which was the case for our experiment.

9. CONCLUSION

Two algorithms, k-means and deep learning, were proposed for automatic segmentation of left ventricle. Each algorithm was implemented and practical experiment was setup to validate and test the algorithm. The result of the practical experiment for k-means reveals the fact that k-mean may prone to failure under complex scenarios. For that reason, alternative algorithm using deep learning based on TensorFlow was developed, trained and tested. Two datasets of 80 and 300 MRI images each were selected to training deep learning network. The experimental result of training and testing show that the result achieved by deep learning algorithm is promising even under significant image changes and more complex scenarios. Besides, dataset with 300 images takes more time (almost two times) more than dataset with 80 images to be trained which leads to 95% to 98% precision in terms of ROI detection and segmentation. The segmentation loss is 50% less when the images of the dataset increase almost four folds.

10. REFERENCES

[1] Ehab Abdelmaguid, Jolene Huang, Sanjay Kenchareddy, Disha Singla, Laura Wilke, Mai H Nguyen, and Ilkay Altintas. Left ventricle segmentation and volume estimation on cardiac mri using deep learning. *arXiv preprint arXiv:1809.06247*, 2018.

[2] Hisham Abdeltawab, Fahmi Khalifa, Fatma Taher, Norah Saleh Alghamdi, Mohammed Ghazal, Garth Beache, Tamer Mohamed, Robert Keynton, and Ayman El-Baz. A deep learning-based approach for automatic segmentation and quantification of the left ventricle from cardiac cine mr images. *Computerized Medical Imaging and Graphics*, page 101717, 2020.

[3] Md Al Noman, ABM Aowlad Hossain, and Md Asadur Rahman. Initial point prediction based parametric active contour model for left ventricle segmentation of cmri images. In *Proceedings of International Joint Conference on Computational Intelligence*, page 533. Springer, 2012.

[4] Avendi. Lv segmentation in cardiac mri, 2019.

[5] Avendi. semi-automatic segmentation of lv in cardiac mr images, 2019.

[6] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[7] et al. Burms, Jeroen. Second annual data science bowl model documentation.https://github.com/317070/kaggle-heart/blob/master/documentation.pdf.

[8] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.

[9] Jinyu Cong, Yuanjie Zheng, Wufeng Xue, Bofeng Cao, and Shuo Li. Ma-shape: Modality adaptation shape regression for left ventricle segmentation on mixed mr and ct images. *IEEE Access*, 7:16584–16593, 2019.

[10] Julian de Wit. 3rd place solution for the second national datascience bowl, 2015.

[11] Julian de wit. Image segmentation with the u-net architecture, 2015.

[12] Julian de Wit. 3rd place solution for the second national datascience bowl, 2016.

[13] Suyu Dong, Gongning Luo, Clara Tam, Wei Wang, Kuanquan Wang, Shaodong Cao, Bo Chen, Henggui Zhang, and Shuo Li. Deep atlas network for efficient 3d left ventricle segmentation on echocardiography. *Medical Image Analysis*, page 101638, 2020.

[14] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[15] Marwa MA Hadhoud, Mohamed I Eladawy, Ahmed Farag, Franco M Montevecchi, and Umberto Morbiducci. Left ventricle segmentation in cardiac mri images. *American Journal of Biomedical Engineering*, 2(3):131–135, 2012.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Wisnumurti Kristanto and Jean-Paul Aben. Left ventricle segmentation in contrast-enhanced cine mri datasets, May 2 2019. US Patent App. 16/167,152.

[19] Y. Lan and R. Jin. Automatic segmentation of the left ventricle from cardiac mri using deep learning and double snake model. *IEEE Access*, 7:128641–128650, 2019.

[20] Hae-Yeoun Lee. Automatic lv segmentation with k-means clustering and graph searching on cardiac mri. *International Journal of Computer and Information Engineering*, 9(3):292–295, 2015.

[21] Hae-Yeoun Lee, Noel CF Codella, Matthew D Cham, Jonathan W Weinsaft, and Yi Wang. Automatic left ventricle segmentation using iterative thresholding and an active contour model with adaptation on short-axis cardiac mri. *IEEE Transactions on Biomedical Engineering*, 57(4):905–913, 2009.

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[24] Gongning Luo, Suyu Dong, Kuanquan Wang, Wangmeng Zuo, Shaodong Cao, and Henggui Zhang. Multi-views fusion cnn for left ventricular volumes estimation on cardiac mr images. *IEEE Transactions on Biomedical Engineering*, 65(9):1924–1934, 2017.

[25] Shakiba Moradi, Mostafa Ghelich Oghli, Azin Alizadehasl, Isaac Shiri, Niki Oveisi, Mehrdad Oveisi, Majid Maleki, and Jan Dhooge. Mfp-unet: A novel deep learning based approach for left ventricle segmentation in echocardiography. *Physica Medica*, 67:58–69, 2019.

[26] Ramon A Moreno, Marina FS de Sá Rebelo, Talles Carvalho, Antonildes N Assunção, Roberto N Dantas, Renata do Val, Angela S Marin, Adriano Bordignom, Cesar H Nomura, and Marco A Gutierrez. A combined deep-learning approach to fully automatic left ventricle segmentation in cardiac magnetic resonance imaging. In *Medical Imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging*, volume 10953, page 109531Y. International Society for Optics and Photonics, 2019.

[27] Khush Patel. custom object detection using tensorflow from scratch, 2019.

[28] Charnchai Pluempitiwiriyawej, José MF Moura, Yi-Jen Lin Wu, and Chien Ho. Stacs: New active contour scheme for cardiac mr image segmentation. *IEEE transactions on medical imaging*, 24(5):593–603, 2005.

[29] JHC Reiber RJ van der Geest. Quantification in cardiac mri. *JOURNAL OF MAGNETIC RESONANCE IMAGIN*, 1999.

[30] Alex Woshi Tencia Lee. left ventrile segmenation using deep learning, 2016.

[31] Qi Liu Tencia Lee. Automatic left ventricle volume calculation in cardiac mri using convolutional neural network, 2016.

[32] Phi Vu Tran. A fully convolutional neural network for cardiac segmentation in short-axis mri. *arXiv preprint arXiv:1604.00494*, 2016.

[33] Hongrong Wei, Wufeng Xue, and Dong Ni. Left ventricle segmentation and quantification with attention-enhanced segmentation and shape correction. In *Proceedings of the Third International Symposium on Image Computing and Digital Medicine*, pages 226–230. ACM, 2019.

[34] Song Chun Zhu and Alan Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 18(9):884–900, 1996.