Deep Learning for Automatic Segmentation of the Right Ventricle in cardiac MRI images

Machine Learning Nanodegree Capstone Proposal

Akshar Kumar

December 2017

1 Domain Background

The evaluation of the right ventricle (RV) in the heart is vital to understand its structure and function. RV segmentation is used when diagnosing diseases such as pulmonary hypertension, coronary heart disease, and cardiomyopathies, among others [1]. Currently, the accepted gold standard for evaluating RV volumes is a manual contour by a trained physician on cardiac magnetic resonance imaging (MRI) images. When segmenting the RV, the physician will create an endocardium and epicardium contour of the ventricle. Endocardium refers to the inner wall of the ventricle. Epicardium is the outer layer of the heart. Figure 1 shows both types of contours. The physician will perform these contours on consecutive images resulting in a set of 2D contours that can form a 3D volume of the ventricle. From this, the physician can determine the volume of the right ventricle at key points in the cardiac cycle, and then determine whether or not the heart is properly functioning.

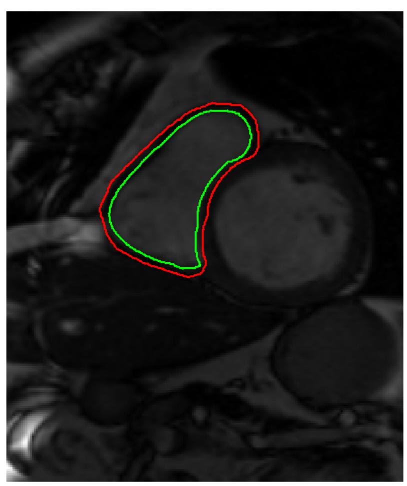

Figure 1: An example of endocardium and epicardium contours of the right ventricle. The green contour is the endocardium contour (inner layer), while the red contour is the epicardium contour (outer layer).

Since there is usually a large number of images to contour, the overall process of segmentation becomes very lengthy. Each image can take anywhere from 10-15 minutes. Aside from the large amount of time, there are some difficulties that can make the examination task more subjective:

- fuzziness of borders due to blood flow
- presence of wall irregularities

- complex shape of the RV - it can appear triangular when viewed longitudinally and crescent-shaped when viewed along the short axis
- variability in cine MRI equipment, institutions, and populations
- noise associated with cine MRI images [1, 2]

Because of these difficulties, the process is prone to intra- and inter-observer variability [1]. The complex nature of manual segmentation could be simplified with the help of automated methods. A trained model automatically segmenting the right ventricle in cardiac MRI images could significantly shorten the overall process time and provide consistent results between physicians. This in turn could assist the physician in making better diagnoses for the patient. One such model we will study is the convolutional neural network.

Convolutional neural networks (CNN) have become a popular solution in automatic segmentation of the RV. Since they are very successful in computer vision problems, they appear to be a natural fit for solving biomedical imaging segmentation [3]. One popular architecture used for biomedical image segmentation is the U-Net architecture, proposed by Ronneberger et al [4]. In this architecture, there is a downsampling path that follows the same structure as a generic CNN. There are convolutional layers followed by max pooling layers, with each step halving the overall image space. The architecture then introduces an upsampling path, which is needed to create the segmentation map of the image with similar dimensions. In this upsampling path, the image size increases, while the number of channels decrease. In this way, we get a segmentation map that will tell us whether or not a pixel is part of the region of interest. Figure 2 shows this architecture in more detail.

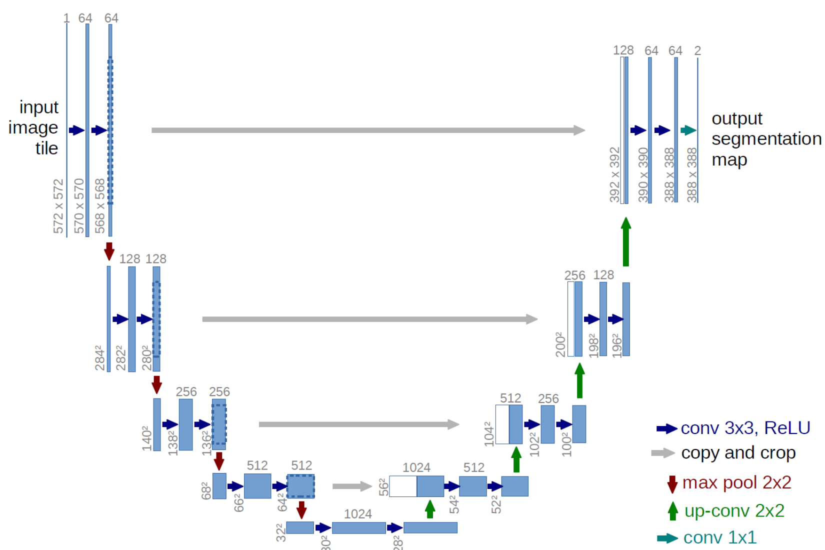

Figure 2: An example of the U-net architecture. Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. [4]

2 Problem Statement

In this project, we will apply a deep learning model to automatically segment the right ventricle in cardiac MRI images. Segmentation of the right ventricle is useful in characterizing the ejection fraction of the heart. We will compare the accuracy of the model to segmentation performed by physicians with years of experience. These manual contours will be what we call the ground truth. This problem was presented as a computer vision challenge at the International Conference on Medical Image Computing and Computer Assisted Intervention in October 2012 [1].

[1] `http://www.litislab.fr/?projet=1rvsc`

3 Datasets and Inputs

The dataset provided in the RSVC competition contained images from 48 patients: 16 training cases, 32 testing cases. For each patient, there are a total of 200-280 images. The images provided in the each case are 2D cine images with approximately 8-12 continuous images spanning a single cardiac cycle for each patient along the short axis view. There were also images from the conventional planes (2-, 3-, and 4-chamber views). The cardiac images have been zoomed and cropped to a 256x116 (or 216x256) pixel region of interest. We will primarily use the images along the short axis view for building our model, as the view is well-suited for estimating volume measurements.

In addition to the images, we are given manual RV segmentation images for the training dataset. The segmentation was performed on end-diastole (ED) and end-systole (ES) images. ED is the first temporal image of each stack, and ES is a mid short axis slice, corresponding with the smallest RV area. The expert manually delineated endocardial and epicardial borders of the RV on the ED and ES short axis slices. Processing time per patient was around 15 minutes. There are a total of 243 labeled images in our training dataset. Due to the small size of our dataset, it will be necessary to augment the data. This will involve making random rotations and translations to the images from the training data. By doing this, we will be able to train our model on variable data and prevent overfitting.

The testing dataset contains 514 MRI images from 32 patients. These images are the same size as the short axis images from the training set (256x216 pixels). The testing set does not contain manual contours. Instead, we will submit contours on the test images for final evaluation to the moderators.

The images are in the Digital Imaging and Communications in Medicine (DICOM) format. DICOM is a communication protocol and file format generally used in medical imaging. It can store medical information along with the patient's information in one file. Since the DICOM format for the data is complex, we will use the suggested tool to work with the images. In order to load the images into python structures for use in the model, we will use the package pydicom.

4 Solution Statement

In order to automatically segment the right ventricle, we will be employing a deep learning model. The specific deep learning model we will use is a convolutional neural network which will take the MRI images as input and output a segmentation map with 2 channels describing whether the pixel is part of the right ventricle or not. The CNN model is a good choice for this problem since it is a computer vision problem and CNNs are good at recognizing patterns in visual data. The contours that will be output by our CNN will be evaluated against manual contours using the Dice coefficient, a metric that indicates the overlap between two areas.

5 Benchmark Model

The benchmark model we will compare our model to is the fully convolutional neural network created by Phi Vu Tran for the competition [2]. This model is a 15-layer deep fully convolutional neural network (FCN). The FCN is comprised of 15 stacked convolutional layers and three max pooling layers. Each of the convolution layers uses a Rectified Linear Unit (ReLU) as the activation function. The total number of parameters is around 11 million. The output of this model is a heat map that predicts the class membership of each pixel in the image.

In order to compare the proposed model to this benchmark model, we will use the Dice coefficient. The DICE coefficient ranges from 0 (total mismatch) to 1(perfect match). For our benchmark model, the DICE coefficient for endocardium contours was 0.84(stdev of 0.21) while for epicardium contours it was 0.86 (stdev of 0.20). These values were the average Dice coefficients from the two test sets provided in the challenge.

6 Evaluation Metrics

For evaluating our model, we will use the Dice coefficient. The Dice coefficient is a measure of the overlap between two contours. The coefficient varies from 0 to 1, with a value of 0 indicating no overlap between two contours. Meanwhile, a value of 1 indicates a perfect overlap between two contours. We will be comparing our automated contours to a manual contour performed by an expert physician using the following equation:

$$D(A, M) = 2 * \frac{A \cap M}{A + M} \tag{1}$$

as described by [2]. Where D represents the Dice coefficient, A represents the area of the automated contour, and M represents the area of the manual contour performed by the expert.

7 Project Design

- **Programming Language**: Python 2.7
- **Libraries**: Keras, Tensorflow, OpenCV, scikit-learn, numpy, pydicom.

Before developing a convolutional neural network, we will have to preprocess the data provided. As provided, the dataset contains 16 patients that have corresponding cardiac MRI scans. Each patient directory has 200-280 cardiac MRI images in the DICOM format. In addition to these images, there is a directory for the manual contours performed by the expert. The expert contoured the right ventricle for each patient on the end diastole and end systole positions of the cardiac cycle. There are two contours for each image, one which is the endocardium (inner wall) contour and the other which is the epicardium (outer wall) contour. The contours are presented as text files with pairs of (X,Y) coordinates corresponding to the pixel position in the image. Since the manual contours are dependent on pixel position, we will not resize the images. However, one of the issues with this dataset is how small it is when we use only labeled data. The images that have corresponding manual contours totals about 200. This is not enough data to train a CNN. In order to increase the training data, we will perform data augmentation. The data augmentation we will perform will be rotations and translations of the images. It is also suggested to perform some deformations on the data, so we will create images that are zoomed and shear to add variability to our dataset. We will perform this augmentation randomly on the data during training.

The next step in the design process is to structure our CNN. The general structure of the CNN will follow that of the U-Net which is a neural network structure used for biomedical imaging segmentation [4]. The basic architecture of this network contains two components. There is a contracting path, and an expanding path. The contracting path follows a generic CNN structure of downsampling the initial image space, using 3 convolutional layers using ReLU activation followed by a max pooling layer. At each downsampling step, the number of channels is doubled, while the image space is halved. This is similar to how many CNNs are structured. However, in order to create a segmentation output map, we need to up sample the structure back to the original image size. In this path, we have convolutional layers that up-samples the feature map, and halves the number of feature channels. The upsampling is able to retain the resolution of the original image by taking a copy of the feature map after each convolutional downsampling layer, and combining that with the upsampled feature map. This allows the upsampling path to increase in resolution and provide a segmentation map of a similar size as the original image. This architecture can be viewed in Figure 2 above.

We will create this architecture and train our model with the training data provided in the competition. We will split our data into training and validation sets for each epoch that is run through the model. Before each epoch, we will randomly augment the data as described above. This will help prevent overfitting. The output of our model will be a segmentation map. The map will have pixels with one of two values, either white for being part of the right ventricle, or black for being outside it. We will further process this output, and find a contour of the right ventricle area. In order to create the contour, we will use the image processing functionality for finding contours in the OpenCV library. This will give us an output in pairs of coordinates that we can compare to the labeled data we have.

After training the model, we will need to evaluate the contours that are output on the testing sets. Since the competition provides unlabeled testing data, we will need to provide our output segmentation maps to the coordinators and they will respond with the evaluation metrics for our model. As described above, we will send the set of coordinates after finding the contours from the segmentation map. While this U-Net model is a start, there will likely need to be some optimization of the model. The main objective is to have the basic U-Net structure running, and then further optimize any hyper-parameters.

References

[1] C. Petitjean, M. A. Zuluaga, and W. Bai, "Right ventricle segmentation from cardiac mri: A collation study," *Medical Image Analysis*, vol. 19, no. 1, pp. 187–202, 2015.

[2] P. V. Tran, "A fully convolutional neural network for cardiac segmentation in short-axis mri." arXiv:1604.00494, 2017.

[3] B. Kayalıbay, G. Jensen, and P. van der Smagt, "Cnn-based segmentation of medical imaging data." arXiv:1701.03056, 2017.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation." arXiv:1505.04597, 2015.