Transforming How We Diagnose Heart Disease

Jacob Jonsson and Henrik von Kleist

***Abstract*—We worked on the prediction of the heart chamber volume from MRI scans as part of the Kaggle challenge "Second Annual Data Science Bowl". We computed the location of the center of the heart by intersecting orthogonal MRI scans in 3-D space and used the centers to extract smaller pictures that only contained the heart. In order to get a sense of the data, we then visualized the preprocessed data, mainly focusing on the UMAP method. We supplied the cropped images together with additional information of the spacing between pixels to a convolutional neural network which predicted the desired heart chamber volumes. We observe good results compared to methods that use only a convolutional neural network with the whole upper body MRI scan as input.**

I. Introduction

A. Problem definition

The systolic and diastolic volume (the volume of the heart at the beginning and the middle of a breath) are used to compute the ejection fraction, a key factor for the prediction of a heart disease. The problem with using MRI to measure cardiac volumes and derive the ejection fraction is that the process is manual and slow. A skilled cardiologist must analyze all MRI scans. Therefore, the goal is to improve this process with an automated computation of a cumulative density function of these volumes. The results will be evaluated using the Continuous Ranked Probability Score (CRPS):

$$C = \frac{1}{600N} \sum_{m=1}^{N} \sum_{n=0}^{599} (P(y \leq n) - H(n - V_m))^2 \qquad (1)$$

where the CDF is discretized from 0 to 599 mL. H represents the the Heaviside step function. M is the total number of patients for systole and diastole and V_m is the true value of the respective volume.

B. Data

The training set consists of MRI scans of 700 patients. For each patient, three different types of MRI scans are available. These are multiple short-axis views (SAX) that come from parallel slices, a 2-chamber view (2CH) and a 4 chamber view (4CH) (see figure 1). For each type, 30 MRI images for different time frames throughout the cardiac cycle were provided. The number of different SAX slices was varying and for some patients, the 2CH or 4CH views were not available. The labels, consisting of the systolic and diastolic volumes determined by a cardiologist were also provided for the training set.

II. Background and Related Work

A. Data Preprocessing

For this problem of supervised learning, the challenge organizers offered the code for a sample solution using a Neural Network approach which we improved. The original idea was to supply the cardiac MRI videos to a neural network and to train this network. Since the MRI scans are made from the whole upper body of the patient, large parts of the scans are useless. Therefore, the first task we had to solve was to automatically find the location of the center of the heart which we solved following the approach by another competitor. [1]

We used the information of the center to crop smaller pictures that only contain the heart which we then use in the neural network. We also use information about the spacing between pixels (how many millimeters are between each pixel), which we also feed to the neural network, since it certainly affects the volumes.

1) Intersection of the different views: Since the SAX view planes are perpendicular to the 2 chamber and 4 chamber view planes, they intersect in 3-D space in one point, the center of the heart. In order to do this, the line where CH2 and SAX slice intersect is projected onto the SAX slice. The line where CH4 and SAX intersect is also projected onto the SAX slice. The intersection of both lines gives the center of the heart.

In figure 2, an example of such an intersection is shown. In a subsequent step, the location of the center is used to cut out an image of the heart that is then used as a new input for the neural network.

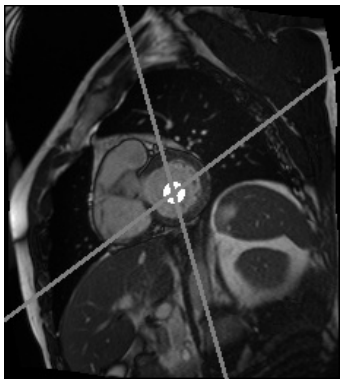


Fig. 1. Located left ventricale by intersection of SAX, CH2 and CH4 views in 3-D space

B. Data visualization

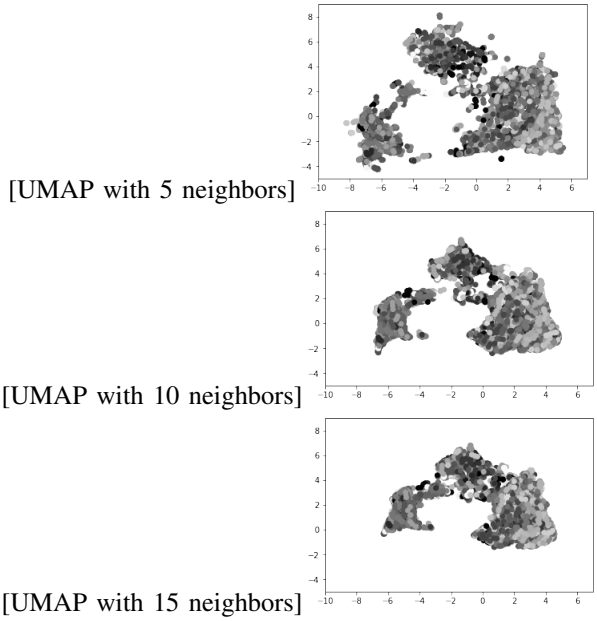
To visualize the data, we viewed the preprocessed pictures with the located heart cut out. By analyzing the labels of volumes, we realized that a few volumes were far larger than others. To improve the visualization, we removed these outliers. Furthermore, instead of looking at a full sequence

[1] https://github.com/ZFTurbo/KAGGLE/_DSB2/blob/master/find_ventricle_location.py

of 30 time frames, we decided to choose the data point of a single time frame for each SAX slice.

To be able to visualize the data in 2D, we used the UMAP method which yielded more telling results than other methods such as t-SNE and PCA, while keeping computational times low.

In figure 3, the resulting data points were plotted with a dark gray color for a low volume and a lighter color for a larger volume. The number of neighbors for the UMAP was experimented with, but using 10 neighbors one can see how larger volume samples seem to tend towards the second quadrant while smaller volume samples have grouped centrally.



[UMAP with 5 neighbors]

[UMAP with 10 neighbors]

[UMAP with 15 neighbors]

Fig. 2. Data visualization using UMAP algorithm with different number of neighbors

III. Model

The model used for the evaluation was a CNN model using three repeated convolutional group layers with max pooling and dropout. The first two convolutional layer groups also used a zero padding layer. As input we use 30 time frames of the preprocessed SAX slice pictures. We don't use the CH2 or CH4 pictures or additional patient information, except for the pixel spacing. As opposed to the reference neural model, we added the meta data for the pixel spacing which scales the picture input. To do this, we had to refrain from the Keras sequential API and instead use the Keras models functional API. The scaling input was added as a flat layer together with the output from the convolutional groups before the final dense layer. A figure of the detailed model can be found in the appendix.

The output of the model, the prediction of the systolic and diastolic volumes is then transformed into a cumulative density function using the validation loss function as a confidence measure (CDF of a Gaussian where the validation loss is used as a variance). Since we predict a solution for every SAX slice video and we have multiple SAX videos for each patient, we also get multiple solutions for each patient. We chose to combine our solutions by taking the average over all predictions.

IV. Training

The training of the model was done using a GPU for significantly improved running times. The system was a deep learning AMI running on an Amazon EC2 instance. We used Python 3.6 (CUDA 9) with a TensorFlow backend. Since the number of training samples is very low, additional training samples were generated by rotation and translation of the pictures. The model was optimized using an Adam optimizer over a root mean squared error function.

V. Results

The prediction of the CDF of the systolic volume for the first patient is shown in figure 4 together with the prediction of the CDF from only one of the available SAX slices of that patient and the Heaviside step function of the true volume.

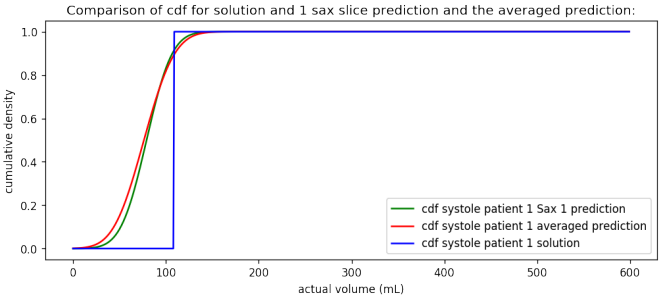


Fig. 3. Patient 1 CDF

The actual CRPS for the test set are computed to be 0.03286 for the case where every sax slice is evaluated individually and 0.02834 when the Sax slice predictions are averaged for a patient.

VI. Conclusion and Future Work

The results from improving the neural model by localizing the heart in the preprocessing in combination with scaling the input data improved the results significantly, taking the solution to a level relative to the 70th place in the Kaggle competition. Using more sophisticated methods, the score could have been improved even further.

One area of improvement is how we handled unexpected results. Redesigning the model, NaN losses initially occurred, likely due to exploding gradients. This was solved by clipping the gradient values to 0.5. Initially, the norm of the gradient was clipped to 1 without results. Thus, we didn't evaluate further if a better result could have been reached by clipping the value higher since the function might have reached a local minima.

In general, the available data was not complete and therefore not easy to handle. The number of time frames per SAX slice was in some cases also not equal to 30. Additionally, some

sax slices missed the heart and therefore did not convey useful information.

Cutting out the heart definitely improved the results. Further improvements can be achieved with a more rigorous cropping. In some cases, however, finding the center did not work. It might be necessary to filter out these cases manually. Another improvement can be made if CH2 and CH4 slices are also used as inputs for the convolutional neural network.

Averaging the results for different SAX slices to predict the true volume for a given patient is a simple and reasonable approach, but a more advanced combination of the different SAX slice results might improve the results significantly.

VII. Appendix

Architecture diagram of the used convolutional neural network: