Convolutional Neural Networks for Estimating Left Ventricular Volume

Ryan Silva
Stanford University
rdsilva@stanford.edu

Maksim Korolev
Stanford University
mkorolev@stanford.edu

Abstract

End-systolic and end-diastolic volumes of the left ventricle are important statistics in diagnosis of cardiac function. We describe a CNN model to automate the process of by estimating volume from multiple time series of MRI images taken from various positions and angles through the heart. We also show evaluation metrics for competing in Kaggle's Second Annual Data Science Bowl.

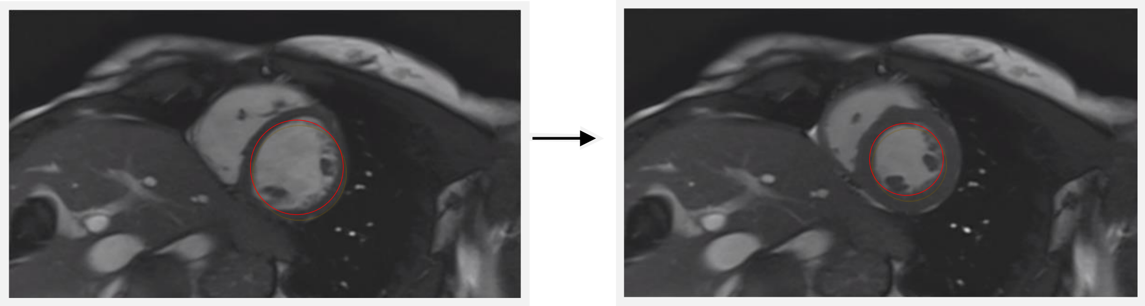
1. Introduction

Each day, 1,500 people in the U.S. alone are diagnosed with heart failure. [1] A key indicator of heart disease is ejection fraction, which is determined by end-diastolic and end-systolic volumes of the left ventricle (LV). End-systolic volume is calculated when the LV is the most contracted, whereas end-diastolic volume is when the LV is most filled with blood. Unfortunately, determining these volumes can take around 20 minutes for a skilled cardiologist to complete[1]. Automating this process would free the cardiologist for more important tasks. The goal of our research is to have a CNN model learn to efficiently estimate LV volume from a series of MRI images taken from different positions through the heart. Kaggle serves to catalyze the solution to this problem through its Second Annual Data Science Bowl competition.

End-systolic and end-diastolic volumes are used to derive the ejection fraction (EF). EF is the percentage of blood ejected from the left ventricle with each heartbeat. [1] Both the volumes and the ejection fraction are predictive of heart disease. While a number of technologies can measure volumes or EF, Magnetic Resonance Imaging (MRI) is most commonly used to assess the heart's proficiency.

To produce the dataset we are using, each patient had multiple MRI time scans of the heart taken in various locations and positions (angles). An entire study of a patient consists of all slices of the LV. A slice is a 2D image taken through the LV at some specified location and position (angle). Each slice in the study is a scan repeated over a full cardiac cycle to produce a time series. An example of the data can be seen in Figure 1. Our dataset consists of studies of both healthy patients and patients with heart problems.

Using the data as described above, the input to our model is a 4D tensor containing all time series slices from all studies in the dataset (a single input is a large stack of MRI images of a single patient).We train a CNN to predict systole and diastole volumes using a regression model trained with a mean squared error loss function.



<u>Figure 1.</u> *The Dataset.*
Each view consists of 30 MRI time series images of one heartbeat. Accentuated in red (not in the dataset) is the area to be calculated in a short axis slice. Total LV volume is calculated as the sum of these sax slice volumes.

2. Related Work

Solutions to the problem of left ventricle segmentation have been proposed since the early 1900's [2]. Methods that have been proposed before the 2000's include: Bayesian Classification [3], superquadratic fitting [4], and heirarchical LV modeling [5]. AI methods for LV segmentation did not appear until 1985. [6]

More recently, a method proposed by A. Newton uses an algorithmic approach to calculate systole and diastole volumes. [7] This method first calculates the location of the LV in an image using a Fourier Transform of the time series for each slice. Since the LV is pumping blood over the entire cycle, this region is expected to have the highest frequencies in the time series. The center of the LV can then be estimated using the method in Lin et al [8]. After finding the center, a region of interest (ROI) is proposed for the area that the LV takes up during the systolic and diastolic phase in the given slice. The estimated areas, along with metadata containing the depth of each slice, are then used to calculate total volume of the LV by assuming a geometric shape of the LV.

Another method proposed by V. Tran calculates volume through semantic segmentation of MRI heart scans with a CNN. [9] Long et al. has shown that CNNs can efficiently learn to make per-pixel predictions for semantic segmentation. [10] A semantic segmentation model is trained in Caffe on the Sunnyrook dataset, which contains slices of the LV that have been segmented by a medical expert. The model is then trained to calculate the systolic and diastolic LV contours for a given slice. After the slices have been segmented to get the areas of the the LV, the method proposed by A. Newton can then adapted to algorithmically estimate volume of the LV. [11]

3. Dataset and Features

The dataset consists of heart MRI scans of 500 patients. Each study contains roughly 11 slices, and each slice is a time series of 30 frames.

The most informative slice position is the short axis, or "sax", which is sliced perpendicular to the long axis of the heart. Cardiologists typically use these particular slices to calculate volumes. Figure 2a shows the location of these cuts in relation to the heart. Additionally included in the dataset were long axis slices of the four chamber and two chamber left views of the heart. However, since the sax slices in each study are taken along the entire length of the long axis, we choose to only use sax slices for training our model because these slices contain all the information necessary to make an estimate of the LV volume. Each image contains a variety of meta data including but not limited to: slice location, patient position, pixel width/height, and slice thickness. The full list of metadata is available at DICOM's website on image standards. [12]

The data comes from various hospitals using different procedures, so there is no uniform standard across images for scale, dimensions, and location and position of slices. For this reason we use the following prepossessing steps to standardize the data.

From inspection of the data, we choose to crop the images to 64x64 around the center of the LV. The center of the LV is found through a Fourier Transform method as done by A. Newton [13]. Briefly, we take the pixels with highest variation throughout the time series of the slice as most likely to be the LV, since the LV has the most blood flowing through it during the cardiac cycle (see methods for details). Next, we re-size each image in the slices to 64x64 pixels. This is done to make the dataset more manageable since the original version is very large (approximately 32GB for training data). Finally, we stack all sax slices together to create a single input for each study.

3.1 Combining SAX views

Cardiologists estimate LV volume by combining calculated volumes from multiple views of the heart. These are called slices, and each slice has its own 30 image time series to determine systole and diastole. Therefore, the optimal way for a neural network to determine systole and diastole volumes is to feed in the ordered slice time series images of the heart. Figure 2b shows this in a graphical format. Each slice was ordered according to location, from base to apex (from 15 to 4 as in Figure 2) and zero padded to organize the entire training set into matrix form.

4. Methods

We chose to use a Convolutional Neural Network model implemented in Keras[1] for estimating LV systolic and diastolic volumes. The approach we take for this problem is an end-to-end learnable network. By first standardizing and grouping the data in a systematic way, we can feed uniform data into a deep network for a more natural end-to-end learnable regression model.

The first step to standardize the data is to crop each image to 128x128 centered on the LV. To do this, we use a Fourier Transform approach to estimate where the LV lies within the image.

[1] We used skeleton code provided publicly by Kaggle member Marko Jocic. [14]

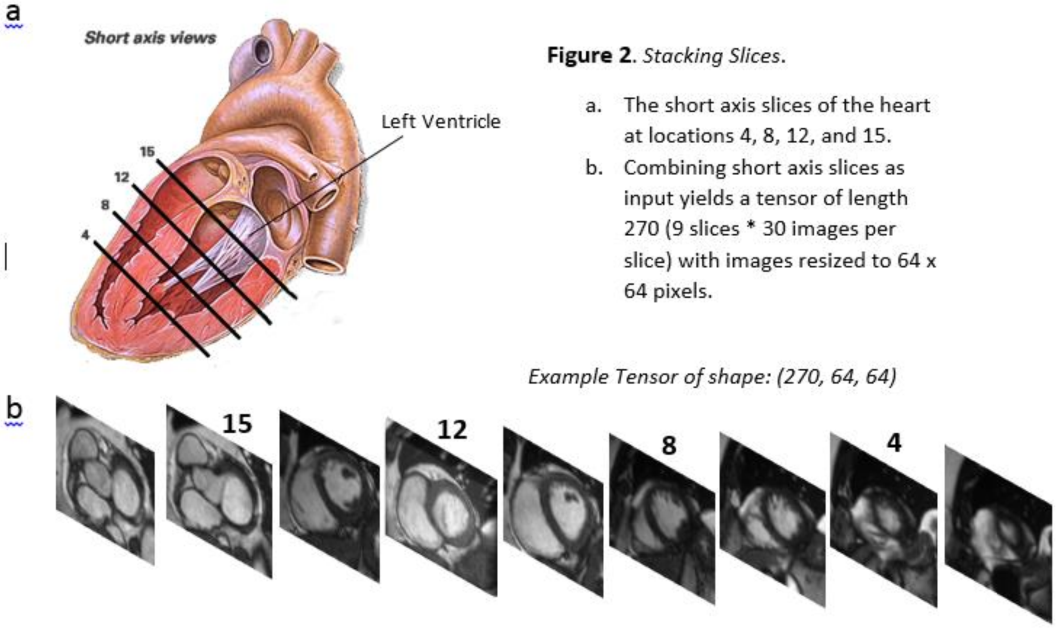


Figure 2. *Stacking Slices.*

a. The short axis slices of the heart at locations 4, 8, 12, and 15.
b. Combining short axis slices as input yields a tensor of length 270 (9 slices * 30 images per slice) with images resized to 64 x 64 pixels.

We use the 2D Fourier Transform:

$$A_{kl} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} a_{mn} \exp\left\{-2\pi i \left(\frac{mk}{M} + \frac{nl}{N}\right)\right\}$$
$$k = 0, \ldots, M-1; \quad l = 0, \ldots, N-1$$

The center of the LV can then be estimated using the method in Lin et al. [15]

We use a Convolutional Neural Network (CNN) for regression on the end-systolic and end-diastolic volumes. For a single patient, we take a number of time series from slices in order (base to apex) and stack them together into a 3D tensor ((n sax slices x 30 images) x height x length). Stacking together all sax time series maximizes the amount of information fed into the network for a single patient. This gives a powerful network all the necessary information to calculate volume since it has the entire mapping of LV. This model can then use supervised learning to regress on the known volumes for each patient in the training set.

We train our models using a Root Mean Squared Error loss function:

$$RMSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

We found that a regression approach for our model was most natural for the problem of estimating a single volume.

We also considered using a different algorithm with a softmax loss defined as:

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

However, for our particular task and given metric, we found that a regression model achieves higher score. This is because the accuracy of the models to predict end-systolic and end-diastolic volumes is judged by Continuous Ranked Probability Score (CRPS), which requires a predicted cumulative distribution function. It is calculated separately as follows for systolic and diastolic volumes from their predicted respective cumulative distribution functions:

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

where P is the predicted distribution, N is the number of rows in the test set, V is the actual volume in mL, and H(x) is the Heaviside step function.

We can estimate a CDF for our model based on our validation error. This error, which we calculate in terms of Root Mean Squared Error (RMSE) can be assumed to be normally distributed (heart volumes across a population will be normally distributed). Given this, our mean is our prediction and our standard deviation is RMSE, which allows us to calculate a normal CDF.

4.1 Metadata inclusion

The last aspects missing to correctly calculate LV volume is the pixel spacing in millimeters (a pixel is square and therefore has the same height/width) which can be used to calculate area of the slice of the heart, and slice thickness, which is used to convert area into volume of the slice. These slices are summed to generate LV volume.

The entire calculation for LV volume is:

$$\sum_{i=1}^{m} area(slice_i)$$

where:

$$area(slice_i) = \sum^{n} pixel\_spacing_i{}^2 * slice\_thickness_i$$

$m\ is\ the\ total\ number\ of\ slices\ in\ the\ study$
$n\ is\ the\ total\ number\ of\ pixels\ in\ the\ i^{th}\ segmented\ slice$

The CNN must detect the segment of the heart, identify diastole and systole, and then count the pixels and perform this calculation. An example of what must be detected can be seen in Figure 1.

We extract the metadata as a tuple of pixel spacing and slice width. This data is very important for the calculation of LV volume, as discussed above in section 3: Dataset and Features. Each study has its own parameters for these variables, and therefore a natural way to incorporate this data into the CNN is to append the tuple to the input of the fully connected layer. This can be seen graphically in Figure 3.

4.1) Model Architecture

The model we use has the following architecture. Each training example is a 3D tensor with shape (300, 64, 64). The first layer is a Batch Normalization layer, which accomplishes preprocessing by shifting and scaling each batch of data. The layers then proceed as: first, a convolution layer with 3x3 filter and stride one is applied, followed with ReLu non linearity activation function. This is repeated, after which a max pooling layer with a 2x2 filter and stride 2 is applied. This entire structure is then repeated two more times for a total of six convolution layers and 3 max pooling layers with ReLu activations in between. The network is flattened, and at this point the metadata for each training example is concatenated to the current activations. A fully connected layer with 1024 hidden units is then applied followed by another ReLu. A single output is produced by a fully connected layer. This architecture can be viewed in Figure 3.

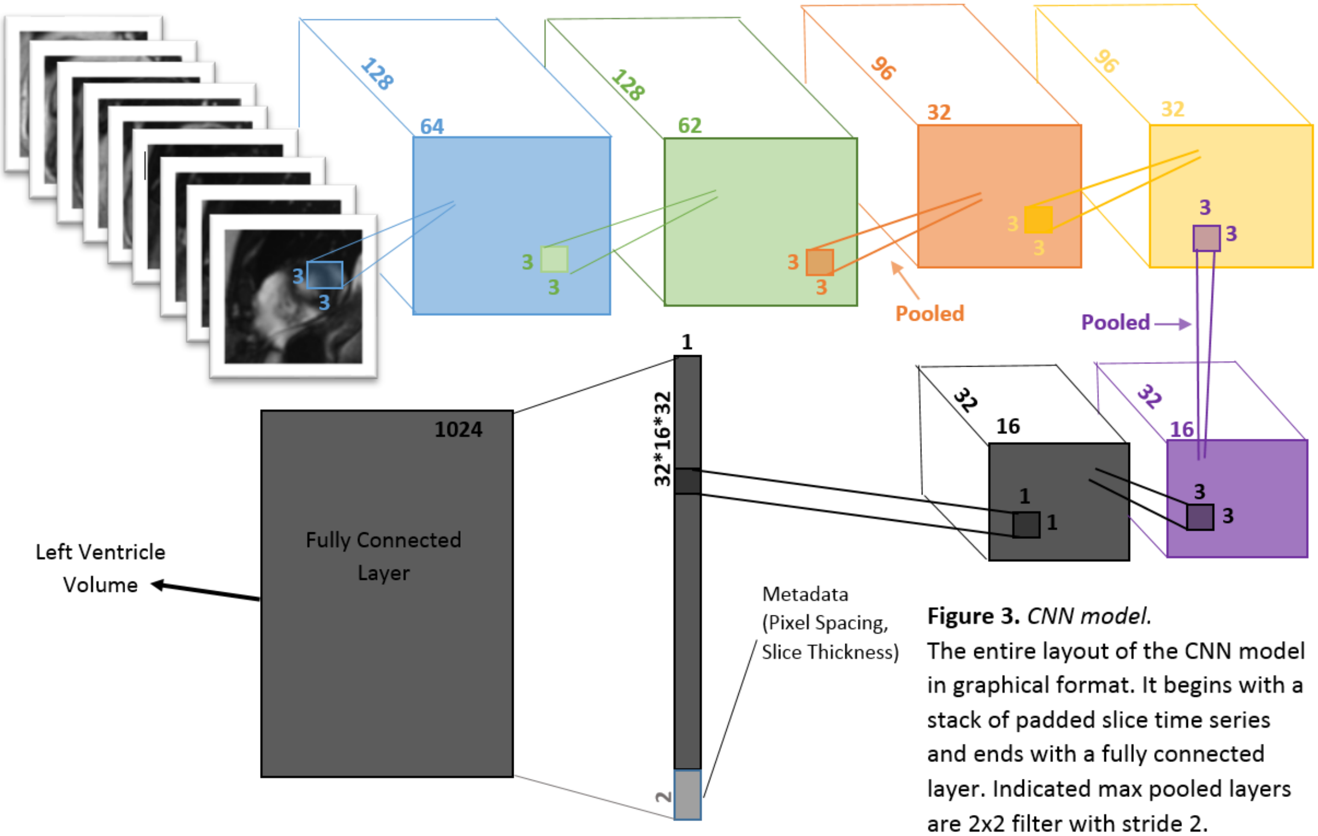


Figure 3. *CNN model.* The entire layout of the CNN model in graphical format. It begins with a stack of padded slice time series and ends with a fully connected layer. Indicated max pooled layers are 2x2 filter with stride 2.

5. Experiments/Results/Discussion

We trained our models using Adam optimization with a learning rate of 1e-4. We train on 80% of the training data set and use the remaining 20% for training time validation. This allows us to prevent over fitting a model by monitoring the loss on the validation set. While k-fold cross validation would be a better option, we have limited compute available. We use a relatively small batch size of 8 since our training examples are large and we train on a graphics processing unit with limited memory. We regularize the fully connected layers of the model with l2 regularization and a hyperparameter of 1e-1. Since the number of times we could run our model was limited by the size of the dataset, we chose popular parameters to use for training models.

The results of our experiments are plotted in Figure 4. We plot three metrics: loss functions for both systolic and diastolic models, CRPS loss as a function of time, and our predicted cumulative distribution function against the ground truth cumulative distribution function. The lowest average RMSE loss for systole and diastole on the validation set was 27.74 mL and 38.55 mL, respectively. The final CRPS loss was 0.0187 and 0.046 for training and test, respectively. Our baseline model CRPS validation score is 0.134308. This is the validation loss achieved by Kaggle’s introductory convolution neural net model. [16]

We noticed high over fitting while training our models. To combat this, we employed several techniques. We used dropout in three places in the network after ReLu activations. The first two dropout rates were 25% and the third was 50%. For layer parameters, we chose to reduce the number of filters at every other layer. We justify this because it reduces the number of parameters in the network, which in turn helps over fitting. Finally, we augment the data with random shifting which also supports a more generalized model.

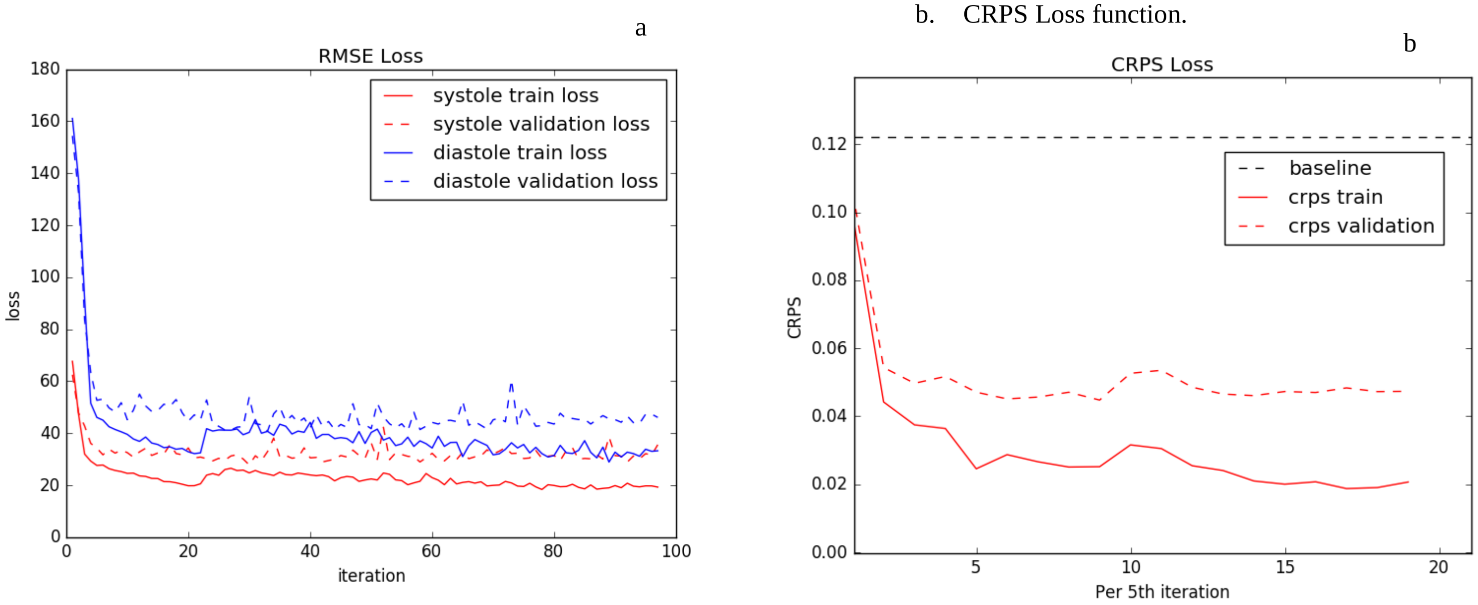
We also noticed that scores for predicting the smaller systolic volume were consistently much better than for the larger diastolic volume. We think that predicting larger volumes is harder is because the information needed to accurately compute larger volumes is spread across more pixels, so deeper neurons with larger receptive fields of the original image must be involved.

Training and validation losses for RMSE and CRPS can be found in Figure 4.

One example of a CRPS predicted probability distribution can be found in Figure 5.

Figure 4. *Losses*.

a. RMSE Loss function.
b. CRPS Loss function.



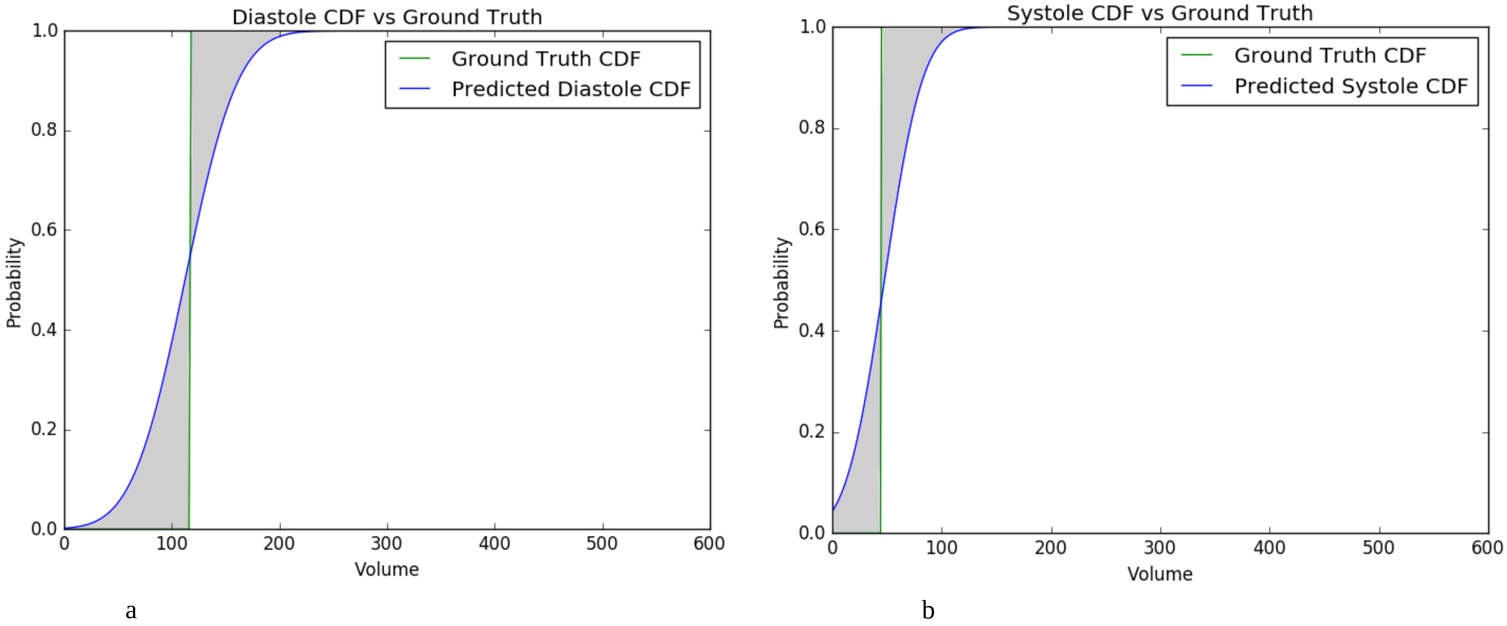


Figure 5. *CRPS Losses with Error Region Highlighted*.

a. CRPS Loss for diastole for a single example. Model sigma was 38.55 mL.
c. CRPS Loss for systole for a single example. Model sigma was 27.74 mL.

Overall CRPS score = 0.0368

6. Conclusion/Future Work

For our project, we aimed to estimate a single volume given many MRI heart scans of a patient. We evaluated our model on the CRPS function- a metric that compared the predicted CDF formed from our predicted volume with the ground truth CDF. Our highest performing model was a pair of eight layer convolutional neural networks trained using a RMSE loss function to output a single predicted systolic and diastolic volume for each patient.

We noticed that scores for predicting the smaller systolic volume were consistently much better than for the larger diastolic volume. As mentioned, it may be that predicting larger volumes is harder as the information needed to accurately compute larger volumes is spread across more pixels, requiring a larger receptive field of neurons. Given further time, we would want to test using larger filter sizes with the diastole model to increase the receptive fields of deep neurons.

If we had more resources, we could have utilized more computing power to train on the entire dataset without downsizing and cropping. We also would consider using an LSTM model to process the time series data within each slice, rather than having a convolutional net layer treat each frame as a separate channel as in our current model. Moreover, we could have used a hybrid convolutional/ LSTM architecture to process each time series with an LSTM, feeding into a CNN to predict the final volume.

References

1. Second Annual Data Science Bowl. *Second Annual Data Science Bowl*. Kaggle and Booz Allen Hamilton.
2. Suri, Jasjit S. Computer Vision, Pattern Recognition and Image Processing in Left Ventricle Segmentation: The Last 50 Years. *Pattern Analysis & Applications* 3.3 (2000): 209-42. Web.
3. Sheehan, Florence H. Method for Determining the Contour of an in Vivo Organ Using Multiple Image Frames of the Organ. Patent US 5734739 A. Mar.-Apr. 1998.
4. Bardinet E, Cohen LD, Ayache N. Tracking and motion analysis of the left ventricle with deformable superquadratics. Medical Image Analysis 1996; 1(2):129–149
5. Revankar S, Sher D. Constrained contouring in the polar coordinates. Proc Computer Vision and Pattern Recognition, 1993; 688–689
6. Grattoni P, Bonamini R. Contour detection of the left ventricle cavity from angiographic images. IEEE Trans Medical Imaging 1985; 4:72–78
7. Booz-allen-hamilton/DSB2. *GitHub*. Web. 13 Mar. 2016. https://github.com/booz-allen-hamilton/DSB2/blob/master/segment.py.
8. Lin, B. R. Cowan, and A. A. Young. Automated detection of left ventricle in 4d mr images: experience from a large study. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006, pages 728–735. Springer, 2006.
9. Tran, Vu. Booz-allen-hamilton/DSB2. *GitHub*. https://github.com/booz-allen-hamilton/DSB2/blob/master/FCN_tutorial.ipynb
10. Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.
11. Booz-allen-hamilton/DSB2. *GitHub*. Web. 13 Mar. 2016. https://github.com/booz-allen-hamilton/DSB2/blob/master/segment.py.
12. DICOM Homepage. *DICOM Homepage*. Web. 13 Mar. 2016.
13. Booz-allen-hamilton/DSB2. *GitHub*. https://github.com/booz-allen-hamilton/DSB2/blob/master/segment.py.
14. Jocic, Marko. Second Annual Data Science Bowl. *Keras Deep Learning Tutorial (~0.0359)* -.
15. Lin, B. R. Cowan, and A. A. Young. Automated detection of left ventricle in 4d mr images: experience from a large study. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006, pages 728–735. Springer, 2006.
16. Tran, Vu. Booz-allen-hamilton/DSB2. *GitHub*. https://github.com/booz-allen-hamilton/DSB2/blob/master/FCN_tutorial.ipynb