Regularization of end-to-end learning for cardiac diagnosis by multitask learning with segmentation

By

Sjirk Gerard Snaauw

in partial fulfilment of the requirements for the degree of

Master of Science
in Biomedical Engineering

at the Delft University of Technology,
to be defended publicly on Friday September 28, 2018 at 2:00 PM.

Supervisor:	Prof. dr. W. J. Niessen	
Thesis committee:	Dr. ir. J. Dijkstra,	LUMC
	Dr. ir. M. C. Goorden,	TU Delft
	Dr. ir. R. F. Remis,	TU Delft

TU Delft

Abstract

Cardiac magnetic resonance (CMR) is used extensively in the diagnosis and management of cardiovascular disease. Deep learning methods have proven to deliver segmentation results comparable to human experts in CMR imaging, however, no successful attempts have been made at fully automated diagnosis. This has been contributed to a lack of sufficiently large datasets required for end-to-end learning of diagnoses. Here we propose to exploit the excellent results obtained in segmentation by jointly training with diagnosis in a multitask learning setting. We hypothesize that segmentation has a regularizing effect on learning and promotes learning of features relevant for diagnosis. Results show a three-fold reduction of the classification error to 0.12 compared to a baseline without segmentation, both results are obtained by training on just 75 cases in a dataset (ACDC) that is equally distributed over 5 classes.

Acknowledgements

This project was performed at the Australian Institute of Machine Learning, part of the university of Adelaide, and in collaboration with the South Australian Health and Medical Research Institute.

I would like to thank my supervisors.
- Wiro Niessen for the support and guidance during my thesis work. Despite the distance he was very involved in these final stages in my journey as a master's student.
- Gustavo Carneiro for helping me grasp the details of many machine learning concepts that allowed me to make a success of this project.
- Dong Gong for teaching me how to implement my first network and the many interesting discussion on network design choices.
- Johan Verjans for his clinical input, his support in both professional and personal aspects during my stay in Australia, and making this incredible experience possible to begin with.

I would also like to thank all the people in the AIML and ACRV for the interesting discussions and showing me their projects with such great enthusiasm that allowed me to make so much progress in such a short time, especially Gabriel 'famous guy' Maicas for the discussions on medical applications and their implications.

I gratefully acknowledge the funding received from the Dutch heart foundation and the foundation of the Vrijvrouwe van Renswoude that allowed me to travel to Australia and perform this research at the university of Adelaide.

Finally, I would like to express my profound gratitude to my parents and sister for their unfailing support throughout my life. This accomplishment would not have been possible without the security of that support. Thank you.

Contents

List of abbreviations

2D, 3D, 4D	{2,3,4}-dimensional
ACDC	Automated Cardiac Diagnosis Challenge
ARV	Abnormal Right Ventricle
BN	Batch Normalization
CB	Classification Branch
CMR	Cardiac Magnetic Resonance
CNN	Convolutional Neural Networks
Conv	Convolution
CVD	Cardio Vascular Disease
DCM	Dilated Cardiomyopathy
DeConv	Transposed Convolution (Deconvolution)
ECG	Electrocardiography
ED	End-Diastole
EDV	End-Diastolic Volume
EF	Ejection Fraction
ES	End-Systole
ESV	End-Systolic Volume
FC	Fully Connected
GAN	Generative Adversarial Network
HCM	Hypertrophic Cardiomyopathy
LV	Left Ventricle
LVM	LV myocardial Mass
RV	Right Ventricle
MB	Main Branch
MINF	Myocardial Infarction
MLP	Multilayer Perceptron
Myo	Left Ventricle Myocardium
NOR	Normal
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SB	Shared Branch
SGD	Stochastic Gradient Descent
SSFP	Steady-State Free Precession
SV	Stroke Volume
SVM	Support Vector Machine
RF	Random Forest
WT	Wall Thickness

1 Introduction

Cardiovascular disease (CVD) is consistently ranked the leading cause of death worldwide, killing more people in 2016 than the next four causes together [1]. Due to the aging population and increasing prevalence of risk factors such as diabetes and obesity, the number of CVD related deaths is predicted to increase by 30% to 23.3 million a year in 2030 [2, 3]. Aiming at earlier detection and disease management, several non-invasive imaging options have been designed for the assessment of CVD.

Cardiovascular Magnetic Resonance (CMR) imaging has proven to be of particular great value in CVD diagnosis and management. A combination of factors such as lack of ionizing radiation, excellent soft tissue contrast, and high reproducibility have made it the preferred imaging modality in the quantification of ventricular volumes, myocardial function and scarring [4-6]. Increasing clinical use has also resulted in an increased application of CMR in large cohort studies [7]. This proliferation of medical imaging datasets will impact the need for automated tools, making machine learning for big imaging data a very promising field.

Several advances in deep learning have enabled machines to outperform humans in image classification if provided with a database of millions of images as in the ImageNet challenge [8, 9]. In the medical field, highly task specific databases of medical photographs have shown capable of accurate diagnosis. Esteva et al. [10], for example, used dermatology images to train a model that detects skin cancer, and Gulshan et al. [11] used retinal fundus images for detection of diabetic retinopathy. Both scored on par with certified clinical experts in their respective field. While the underlying data is very different from cardiac volumetric data, the wide range of application shows that deep learning can indeed be used for diagnosis in the medical field.

Current machine learning based methods for automated cardiac diagnosis focus on detection and segmentation of the heart, followed by the extraction of features that are then used for diagnosis. This approach is reflected in the 2017 Automated Cardiac Diagnosis Challenge (ACDC) where aim the aim is to automatically perform segment and diagnosis on a 4D cine-CMR scan. All but one participant in the segmentation part of the challenge used deep learning, scoring on par with clinical experts [12], while none of the participants in the classification part did. Instead they performed classification using support vector machines (SVM) and random forests (RF) on handcrafted features extracted from obtained segmentation maps [12].

As the handcrafted features define clinical diagnosis of the pathology, extraction of those features is a reasonable approach. Subsequent diagnosis using machine learning based methods shows the demand for flexibility in the current diagnostic process that cannot easily be captured in rule-based methods. Deep learning can provide the required flexibility and perform accurate segmentation and classification as shown in state-of-the-art methods [13]. However, no attempts have been made at end-to-end learning for diagnostic purposes in cardiology. One explanation for this is insufficient data, a recurring statement for deep learning in medical image analysis [14, 15].

In this thesis we propose to exploit the excellent result of cardiac segmentation in deep learning by jointly training with disease prediction on the ACDC dataset in a multitask learning setting. The combined training with segmentation serves two purposes. First, a typical cine-CMR scan contains millions of voxels that are individually labeled and evaluated together to provide a smooth learning update that could balance out the crude update from a single diagnostic prediction. This regularization by segmentation allows the model to learn faster. Second, as the

segmentation maps have shown to contain information relevant for diagnosis [16], the use of features that are important for segmentation can guide the learning of relevant diagnostic features.

Additionally, we propose to include a third task in our multitask learning set-up. We include a regression model that estimates the handcrafted features used for clinical diagnosis as inspired by [17]. It is important to note that accurate segmentation or quantification of cardiac indices are not part of the objectives, they only serve as a means of improving diagnostic accuracy.

In the remainder of this thesis we will first look at some background information on the heart, deep learning, the ACDC dataset, and available literature, to allow this work to be seen in the right perspective. This is followed by a description of the experiments in the materials and methods section. In the results we show that that multitask learning is capable of obtaining diagnostic results comparable to state-of-the-art in the challenge. This thesis is concluded with a summary of our contributions, discussion of the limitations, and suggestions for future research.

2 Background

This section contains a brief summary of the heart, dataset, and basics of deep learning. These topics serve as a non-exhaustive introduction to the main concepts required for placing this research in the right perspective, and familiarizing readers with basic concepts.

2.1 Heart

Cardiac anatomy, see Figure 1, is split in a left and right half, both containing an atrium and ventricle. Left ventricle (LV), left atrium (LA), right ventricle (RV), and right atrium (RA). The right half pumps blood through the lungs for oxygenation, followed by distribution through the body by the left half of the heart. The ventricles are responsible for the pumping function of the heart while the atria prevent stasis of venous blood flow during systole (i.e. contraction of the ventricles). A valve is located at each end of the ventricles to prevent backflow of blood. Two surfaces are defined in the heart. The *epicardial surface* describes the outer surface of the heart while the *endocardial surface* refers to the lining on the inside of a chamber. In imaging, the epicardial contour usually refers to the outer contour of the LV myocardial (e.g. muscle) tissue (Myo), though partially anatomically incorrect. Three *coronary arteries* originate directly after the aortic valve (i.e. the valve between LV and aorta) that traverse the epicardial surface of the heart to supply the myocardium with oxygen and other nutrients. The *axis* of the heart is defined as the line between the *apex* and *base*). Long-axis planes are parallel to this axis while short-axis planes are defined perpendicular to the axis.

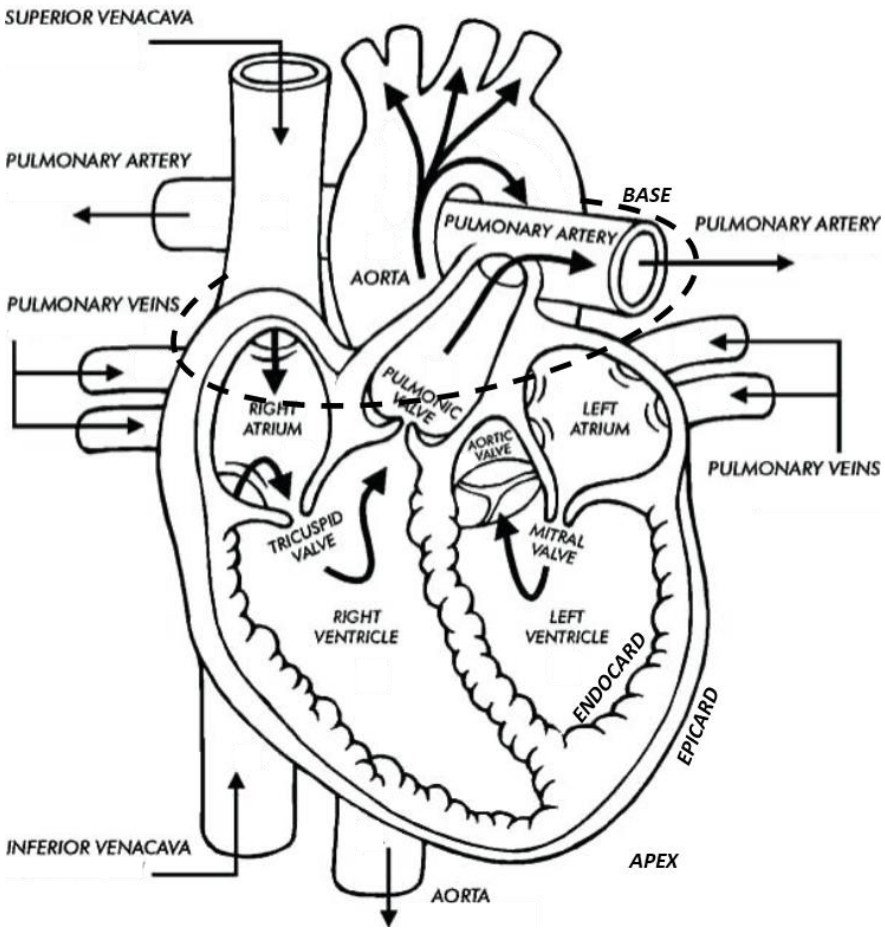

Figure 1: Anatomy of the heart.

Definition of relevant terms and indices commonly extracted from cine-CMR are given below.

- Body Surface Area (BSA): Dubois and Dubois $0.007184 \cdot (weigth^{0.425} \cdot heigth^{0.725})$
- End-Diastole (ED): Moment when mitral valve closes right before contraction of the heart.
- End-Systole (ES): End of contraction when the mitral valve opens.
- ED, ES volume (EDV, ESV): Blood volume in the ventricle at ED and ES respectively
- ED, ES volume index (EDVI, ESVI) = EDV/BSA, ESV/BSA: Volume normalized by division with BSA.
- Stroke volume (SV) = EDV-ESV: Blood volume pumped by the ventricle during a single heartbeat.
- Cardiac output (CO) = SV * heart rate: Blood volume pumped per minute by the heart.
- Ejection fraction (EF) = SV/EDV: Fraction of blood removed from the heart during a beat.
- LV mass (LVM): LV myocardial volume at ED multiplied with the tissue density (1.05).
- Wall Thickness (WT): Distance from a random point on endocardial surface to its closest point on epicardial surface.

2.2 ACDC dataset

The ACDC challenge [12] provides a total of 150 short-axis cine-CMR scans, split in a training and test set that contain 100 and 50 scans respectively. Both sets are equally distributed over five disease classes and contain the patients' height and weight. Labels available for the training set consists of disease class label and segmentation maps. Segmentation maps are available for the end-systolic (ES) and end-diastolic (ED) phase and contain four class labels, namely LV and RV cavity (LV and RV), LV myocardium (Myo), and background (BG). Diagnostic labels present in the dataset are normal or healthy (NOR), myocardial infarction (MINF), dilated cardiomyopathy (DCM), hypertrophic cardiomyopathy (HCM), and abnormal right ventricle (ARV). Characteristics for the disease classes are given in Table 1. Patients with ambiguous (e.g. borderline) values are excluded from the study.

Short-axis cine scans consist of stacked single slice temporal recordings (i.e. stacked 2D + time recordings) to generate a 4D dataset. Data acquisition is performed with a magnetic field strength of 1.5T or 3.0T using the conventional SSFP sequence in breath hold with ECG gating. Slices are recorded with a spatial resolution from 1.34 to 1.68 mm^2/pixel, thickness of 5 or 10 mm, and contained an inter-slice gap of 5 mm in some cases. ECG-gating is used to divide the cardiac cycle in 28 to 40 steps. Long-axis scans (not provided) were used for positioning of short-axis acquisitions and definition of cardiac phase using mitral valve motion – closing defines ED, opening ES.

		NOR	MINF	DCM	HCM	ARV*
LV	EF (%)	> 50	< 40	< 40	> 55	
	EDVI (mL/m^2)	< 90 ♂ | < 80 ♀	↑	> 100		
RV	EF (%)	> 40				< 40
	EDVI (mL/m^2)	< 100		↑		> 110 ♂ | > 100 ♀
Myo	WT (mm)	< 12	↑ (local)	< 12	> 15	
	Contraction	Normal	Abnormal			
	Mass (g/m^2)			↑	> 110	

*Table 1: Indices defining the five diagnostic classes as provided by the challenge organizers [12]. For the four pathological cases, only the indices that define the pathology are given. Ambiguous cases are excluded from the study.↑ Possibly elevated due to compensation for pathology; * One criteria needs to be satisfied for ARV diagnosis.*

2.3 Deep learning

Main components and concepts are introduced to understand the rest of this thesis. The book 'Deep Learning' by Goodfellow et al. [18], and several review articles can be consulted for an extensive overview of the field in general [19, 20] and its application to medical imaging [15, 21].

Learning from training data

At its basis, deep learning is a series of algorithmic advances building on the foundations of neural networks that have been around for a couple of decades. The reason deep learning is so successful can be contributed to those algorithmic advances, but also, by and large, to the increases in computational power and availability of big data.

The perceptron is the earliest and simplest form of an artificial neural network [22]. It consists of an input layer and output layer, see Figure 2. In this feedforward neural network, information propagates forward from one layer to the next, but not backwards or within the layer. From the input vector $\mathbf{x} \in \mathbb{R}^N$, the perceptron calculates the output vector $\mathbf{y} \in \mathbb{R}^M$ as a weighted sum $\mathbf{W} \in \mathbb{R}^{M \times N}$ of the inputs plus some offset values $\mathbf{b} \in \mathbb{R}^M$. The weights and biases are the learned parameter set $\theta = \{\mathbf{W}, \mathbf{b}\}$ of the perceptron. In imitation to the brain, where a neuron fires if it receives the right combination and strengths of inputs from its downstream neurons, an activation function $\sigma(\cdot)$ can be applied to all the nodes in the output layer. A more complex transformation can be learned by adding hidden layers between the input and output layer, turning it in a multilayer perceptron (MLP). In an MLP, the output of one layer serves as the input of the next (2.1).

$$\hat{\mathbf{y}}(\mathbf{x}, \theta) = \sigma^{(2)}\left(\mathbf{W}^{(2)}\sigma^{(1)}\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right) + \mathbf{b}^{(2)}\right) \tag{2.1}$$

Activation functions are sometimes referred to as non-linearities as they add a non-linear component to a layer (unless it is an identity activation). This allows the network to handle complex data patterns that cannot be captured by a linear model. There exist many types of activation functions. However, it is beyond the scope of this thesis to evaluate them all. Here we only give the most popular ones, Figure 2. Historically the sigmoid function was used for activation and later replaced by the hyperbolic tangent (tanh). While effective, both are computationally expensive and have several unfavorable properties for learning. Rectified linear units (ReLU) $\sigma(x) = \max(0, x)$ and Leaky ReLU or parametrized ReLU (PReLU) $\sigma(x) = 1(x < 0)(\alpha x) + 1(x \geq 0)(x)$ (where α is a hyper or learned parameter respectively) are simple yet effective activation functions that can be found in most current models [9, 23].

So far the network is only capable of predicting an output given an input, but it is not learning yet. A combination of back-propagation and stochastic gradient descent (SGD) can be used to update, or learn, the model parameters. The back-propagation algorithm calculates the gradient of the loss function $L(\theta)$ with respect to the model parameters at each point in the network under certain conditions [24]. The assumption in gradient descent is that by updating each model parameter with a small step (η learning rate) in opposite direction of the gradient, the parameter set values get a bit closer to a (possibly local) optimum for the model's task. In SGD, the gradient of the loss is computed over a subset of the dataset (termed mini-batch or batch) to update model parameters.

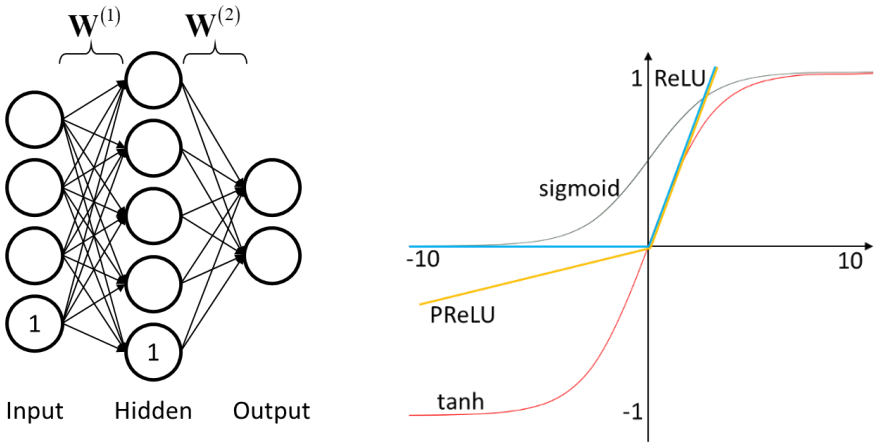

Figure 2: Left – Multilayer perceptron (MLP) with one hidden layer. All lines between two subsequent layers can be collected in a weight matrix. Bias values can be learned as weights by adding an extra node with constant value to a layer. Activations are applied directly after the learned layer. Right – Visualization of the four common activation functions.

$$\theta^{(\tau+1)} = \theta^{(\tau)} - \eta \nabla_{\theta} L\left(\theta^{(\tau)}\right) \tag{2.2}$$

If the mini-batch size is large enough, its gradient can give a good approximation for the gradient over the entire dataset. As a result, SGD performs parameter updates similar to the gradient over all data but more frequently, allowing the model to converge faster. This of course only works if the mini-batch samples the entire feature space of the available data.

Processing each pixel in an image independently as in a fully connected perceptron would require learning of $(N+1)\times M$ parameters per layer learned, where N is the number of nodes in the input layer and M the nodes in the output layer. For any reasonably sized image, the number of parameters needed to learn would be unfeasible beyond a few layers. It would also require the model to be presented with every possible variation on the same image as moving an object one pixel would result in completely different activations in the network that still need to be recognized as the same object.

Convolutional neural networks (CNN) are the main method for processing images, Figure 3. CNNs compute the next layer in the network by convoluting the image with a learned filter (usually applied as correlation). Filters apply a function to a local patch that results in a large output value if the underlying image patch matches the filter. By applying a filter to every local patch in an image, a feature map is generated that shows where the filter's features are present in the image. This approach requires only the parameters of the filters to be learned that are shared over the entire image, significantly reducing the numbers of parameters to be learned. Applying the same filter to every patch in the image makes CNNs shift invariant. Different forms of filters can be applied to increase receptive field (larger size, dilated filter), decrease image size (strided convolution, pooling), and many more. See the review of Gu et al. [25] for more information on CNNs.

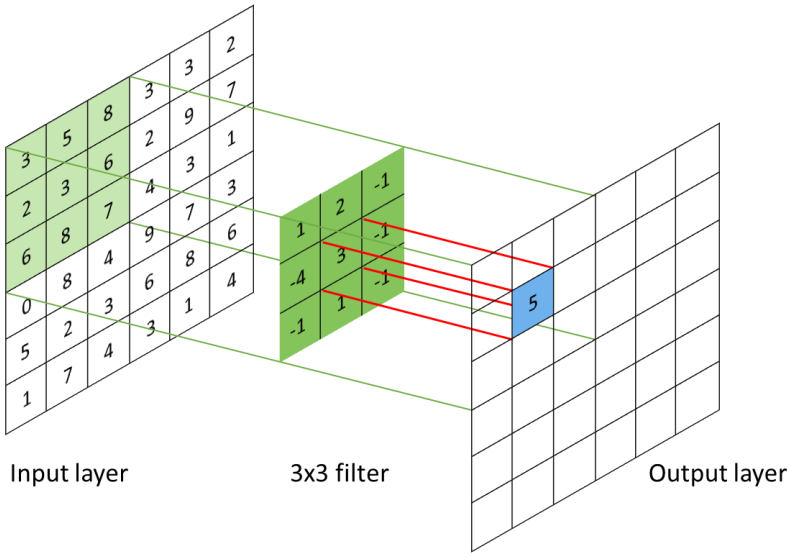

Figure 3: Convolution operation applied in CNNs. A 3x3 filter is applied to the input layer as a sum of the elementwise products to compute the value in the output layer. The values in the filters are the learned parameters in deep learning. Compared to the fully connected setting of MLP, CNNs require only a fraction of the parameters to be learned. As the same filter is applied to the entire image, CNNs are translation invariant.

Regularization

Once a model is able to learn, it will start to update the parameters in a way that produces optimal results on the training set. Optimal parameter values for the training set may not be optimal for describing the distribution of data in general. For instance, let's say the training data is sampled from parabolic function and contains some measurement noise or discretization errors. This may result in finding an optimal fit by a function much higher than second order. This higher order function is said to overfit the training data and needs to be regularized so it generalizes to the underlying distribution, a second order function, Figure 4. Regularization is achieved by introducing additional information to prevent overfitting. This can either be explicitly, by working on the model parameters, or implicitly, by adding information elsewhere in the model that indirectly impact parameter values.

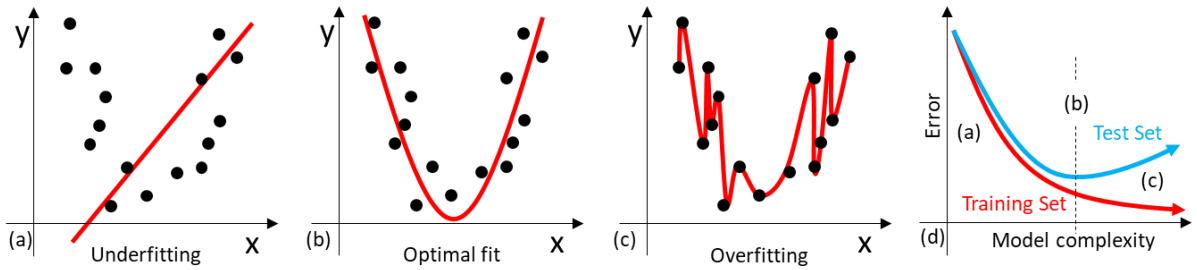

Figure 4: Different stages in learning. When a model starts to learn it will underfit the data and is said to have high bias (a). Once it is learning more, a model's performance will gradually move through an optimal fit (b) and eventually overfit the data, resulting in high variance (c). (d) shows the error curves for the training and test set. Optimal results for a model are obtained when the error on the test set starts to increase again. Regularization aims to lower the error where the test set curve reaches its tipping point. Ideally, regularization forces the curve for test and training to be the same.

Norm penalties, such as L1 and L2 norm, can be added to the loss function of the model to penalize high parameter values. The sort of effect the penalty will have depends on the norm used. L1 for example, generates sparse outputs, effectively removing some parameters from the network while L2 reduces the size of parameter values. Making the parameters sparse or low valued prevents overfitting by reducing the number of parameters or the influence of single parameter on the overall model respectively. The penalty is added as a sum of all parameter

norms multiplied by hyperparameter value lambda. An appropriate value for lambda needs to be found to get a good balance between penalty and the other values in the loss function. If lambda is too small or too big, the model will overfit or underfit respectively. It can be shown that adding small amounts of noise to the input of the network is equivalent to the L2 norm.

Dropout is another popular regularization method directly working on the model parameters [26]. In dropout, each node in the network (and its connections) has a probability *p* of being dropped from the network. This prevents co-adaptation where a specific feature is only useful in combination with other features, i.e. it forces the learned feature detectors to produce features that are relevant on their own. The network effectively learns multiple less complex networks that are averaged at test time (where no nodes are dropped).

Early stopping aims to prevent overfitting by stopping the learning process as it starts to overfit on the training data. By restarting from the previously most successful model parameters, the model's performance may increase again as the learning process contains several steps with stochastic effects.

As the complexity the model can handle depends on the amount of data it is trained on, a simple way of generalization is using more data. This can be done by actually getting more data but is usually an expensive and time-consuming process. Semi-supervised learning can reduce the amount time and costs related to increasing dataset size by adding data that is not labelled. Data augmentation is another way of getting more data, by generating new data from the available data. Common ways of data augmentation are translation, rotation, scaling, mirroring, and cropping of true images. Another way is generating completely new data using generative adversarial networks (GAN) [27]. Two networks are trained in GANs, one network generates a new image and the other discriminates between the generated and true image. This allows new images to be generated that appear to be real. Artificially generating more data can be useful in certain circumstances, however, care must be taken to create new data that is still an accurate representation of the true data being modeled. For example, in our case, scaling of images could result in normal hearts that appear to be enlarged or the other way around.

Multitask learning involves learning of multiple related tasks at the same time. If the tasks can partially be described by the same features, part of the network can learn a general representation for all tasks. On top of this general representation, some task specific features can be learned [28]. For example, if asked to describe a certain breed of dog to someone that has never seen a dog, most of the description would apply to all dogs, and contain a few details that distinguishes it from other breeds. In multitask learning, this general concept of a dog would be described in the shared part of the network. The few characteristic features that are required to distinguish the different breeds of dogs that are learned in the task specific parts of the network.

.

3 Literature

3.1 Automatic Cardiac Diagnosis Challenge

The organizers of the challenge published an extensive overview of all proposed methods and compared the results to each other and experts [12]. Below we give a summary, split in the two objectives of the challenge – segmentation and diagnosis.

Segmentation

Of the ten competing groups, nine used deep learning for image segmentation. The one group that didn't use deep learning used level-sets in combination with a Markov random fiels (MRF) graph cut for an initial segmentation and finetuned their result with spline fitting [29]. Eight groups used U-net type of structures, varying their implementations in 2D vs 3D [30-32], implementation of skip-connections [33, 34] or transition layers [35], addition of inception layers [36], incorporation of an atlas [37], and loss functions – mainly cross-entropy and Dice loss. Wolterink et al. [38] are the only group that did not use an encoder-decoder structure. Instead, they used a CNN with a series of dilated convolutional kernels with increasing dilation to capture sufficient context.

Results are evaluated on the 50 test scans using Dice index and Hausdorff distance. Dice coefficients measure overall correspondence of predicted segmentation and ground truth, while Hausdorff distance measures local boundary inaccuracies. Ignoring the results of the only group with a partial entry, deep learning methods significantly outperform the only non-deep learning method. All deep learning methods produce consistent results and the top 5 differ less than a single voxel for 9 out of 12 metrics. An investigation in to where the methods fail shows that there is no difference in performance for specific pathology or 1.5T vs 3.0T scanners. However, methods perform significantly worse on slices at the apex and base compared to mid-ventricular slices. This is contributed to contour blurring due to partial volume effects and can also be witnessed in manual segmentations by experts.

To compare the state-of-the-art deep learning segmentation methods with expert segmentations, test set images are segmented by two experts of which one segmented the images twice (one month apart) to measure inter- and intra-observer variation. Interestingly enough, the average of all deep learning method scores in-between the inter- and intra-observer on all Dice scores, and Hausdorff distance is only 2-3 mm larger than the inter-observer score. For comparison, average in-slice voxel size is about 1.6 mm. If slices at the base and apex are excluded, the Hausdorff distance becomes lower than inter-observer score as well. In answer to the – Is the problem solved? – question raised in the paper's title, the authors note that there are still issues in segmentation of apex and base and there is need for a new metric that can replace Dice index in evaluation of these issues. The authors also suggest that the remaining issues can possible be solved by training on larger datasets such as the UK Biobank [39]. Since then, a deep learning based segmentation has been compared to inter- and intra-observer variation on the UK Biobank that shows similar overall performance between man and machine and same issues at base and apex [40].

Diagnosis

All groups that entered the diagnosis part of the challenge extracted features using the segmentation maps. Three groups extracted hand designed features that are used in clinic for diagnosis from their obtained segmentation maps, and added patient height and weight and some of their own features. Diagnosis is performed using a 1000-tree RF [38], an ensemble of multilayer perceptrons + RF [30], and a 100-tree RF [36] for accuracies of respectively 0.86%,

0.92%, and 0.96%. The best performing group has since then improved their model's accuracy to 100% by adding a cascade of classifiers to handle challenging cases [16]. The remaining group were the only ones to use semi-automatic segmentation and used a radiomics [41] approach to extract a set of 567 features. Diagnosis was performed with an SVM to obtain an accuracy of 0.92% [42]. Interestingly enough, where nearly all segmentation methods consisted of deep learning techniques, not a single classification method did.

3.2 Applications in cardiac imaging

While neural networks have been around for a few decades, required computational power and other practical issues prevented them from having significant contributions to (medical) image analysis until recently. First LV segmentation using deep learning was performed by Carneiro et al. [43] using a deep believe network (DBN) in ultrasound images, a method they improved to include tracking [44], and adapted for steering the learning of a multi-atlas segmentation process [45]. DBNs were also used in CMR by Ngo et al. [46] to initialize a level-set framework for LV segmentation.

CT is preferred over CMR for assessment of the coronary arteries where several methods have been developed for calcium scoring [47, 48], centerline extraction [49], or (landmark) localization [50-52], and LV segmentation [53]. All CT based methods used a CNN architecture.

In CMR, the main application has been segmentation of the LV, using a fully convolutional network (FCN) [38, 40, 54], patch-based CNNs combined with active contours [55], a combination of stacked auto-encoders (SAE) and CNN for initialization and optimization of a deformable model [56], many U-net kind of structures [30-37], an recurrent neural network (RNN) to process an entire stack [57], multi-organ segmentation using SAE [58] and CNN [59], and many more. This large amount of work in segmentation can in part be contributed to two challenges that focused on segmentation and cardiac volume prediction – the 2015 Kaggle Data Science Bowl[1] and the 2017 ACDC challenge[2].

Other applications of CNNs consist of LV detection in a slice [60], detection of base/apex for quality control of a scan [61], and super-resolution to counter issues due to relative thick slices in CMR [62]. Besides processing an image stack for segmentation, RNNs have also been used for cardiac phase prediction [63-65].

Xue et al. [64, 65] have performed full left ventricle quantification via deep multi task relationship learning. They used a combination of CNNs and RNNs to quantify six measures of regional wall thickness, three cavity dimensions, end-systole and end-diastole cavity area, and the binary cardiac phase from short-axis cine scans by learning the relationship between the measures. This dataset is currently used in the 2018 Left Ventricle Full Quantification Challenge[3] and has the same objective.

Bello et al. [66] are, as far as we know, the only ones that used deep learning for computer aided diagnosis in cardiac imaging. They predicted mortality due to pulmonary hypertension from 3D shape models of RV motion patters using a deep survival network [67] that consists of just

[1] www.kaggle.com/c/second-annual-data-science-bowl

[2] www.creatis.insa-lyon.fr/Challenge/acdc/

[3] lvquan18.github.io/

3 layers. Other groups that allege to use deep learning for cardiac disease prediction only do so for segmentation, followed by other machine learning methods on the handcrafted features.

Having the ability to learn its own features for a highly non-linear classification process provides deep learning models with many benefits over handcrafted features. Since the demonstration AlexNet's [68] superior performance on the ImageNet challenge, deep learning models are consistently top ranked at all (medical) image classification and segmentation challenges. While clearly outperforming other approaches on a range of tasks, only one diagnostic model currently exist in cardiology [66]. Deep learning has made more progress in the area of diagnosis in oncology and brain imaging. In oncology it is frequently used to detect and classify lesions as benign or malignant, and in brain imaging it has been used for disorder classification. The amount of available data plays a crucial role deep learning. It is the consistently given as main issue in deep learning, or as reason to prefer other classification methods over end-to-end learning [15, 21, 53].

4 Materials and methods

4.1 Data

The dataset is represented by $\mathcal{D} = \left\{\left(\mathbf{x}_{ED}, \mathbf{x}_{ES}, \mathbf{m}_{ED}, \mathbf{m}_{ES}, y\right)_i\right\}_{i=1}^{|D|}$, where $\mathbf{x}_{ED}, \mathbf{x}_{ES} : \Omega \to \mathbb{R}$ are the short-axis cine volumes at ED and ES instances, $\mathbf{m}_{ED}, \mathbf{m}_{ES} : \Omega \to \mathbb{R}$ the corresponding segmentation maps with per pixel label $m \in \mathcal{M} = \{BG, RV, Myo, LV\}$, and $y \in \mathcal{Y} = \{NOR, MINF, DCM, HCM, ARV\}$ the diagnostic class label. In both cases, $\Omega \in \mathbb{R}^3$. The 100 fully annotated samples are split 3:1 equally distributed over the diagnostic classes in a training and validation set.

Short-axis cine scans of the two phases are combined to form a single triple-channel volume $\mathbf{x} = (\mathbf{x}_{ED}, \mathbf{x}_{sub}, \mathbf{x}_{ES})$ per sample where $\mathbf{x}_{sub} = \mathbf{x}_{ED} - \mathbf{x}_{ES}$ is the subtraction volume of the two phases, explicitly incorporating temporal information in the input.

A set of $R = 13$ handcrafted volumetric cardiac features are extracted from the segmentation maps and predicted. The extracted features are the six volumes of LV, RV, and Myo from both phases; the three ratios of these volumes (EF); and two LV/RV ratios and two Myo/LV ratios from both phases. The resulting handcrafted feature vector is given by $\mathbf{r}$.

4.2 Preprocessing

As the cine scans consist of a stack of individually acquired slices, each at a different breath hold allowing translation of the heart and intensity scaling differences from one acquisition to the next, preprocessing is performed on individual slices.

Cine slices are resampled to $1.0 \times 1.0 \times Z$ mm per voxel using bicubic interpolation and Z the samples original slice thickness, followed by grey value normalization to zero-mean and unit-variance. Resampled segmentation maps are created by generating a pseudo probability map for each of the four labels using bicubic interpolation on a binary map of each individual label. In the resampled map, voxels are assigned to the label that has the highest interpolated value for a given voxel.

Three-dimensional bounding boxes are extracted around the heart from the segmentation maps and used to center crop the resampled slices and segmentation maps to $192 \times 192 \times S$ voxels where S is the number of slices in a volume.

4.3 Network architecture

Three branches are identified in the network, namely main or shared (MB), segmentation (SB), and classification branch (CB), Figure 5. The main branch is shared by all tasks and together with the segmentation branch they form a U-net like structure [69]. Skip-connections between the two are omitted to prevent the segmentation task from bypassing the flow of information coming from the classification task at the bottom of the U-net.

All three branches apply a DenseNet-like structure that is adapted to handle 3D data and is optimized for each branch's respective task. Main components in DenseNet structures are DenseBlocks followed by transition layers [70]. Each DenseBlock consists of a series of composite functions $\mathcal{H}(\cdot)$ (also referred to as layers) that produce a fixed number of feature

maps each as defined by the growth rate. The input of the composite function is concatenated with its output to produce the input for the next composite function. This provides each layers in the block with the block's input and all features maps created by previous layers, improving the backflow of information for learning.

The composite function in all dense blocks consist of batch normalization (BN) [71], activation by a rectified linear unit (ReLU) [23], and convolution (Conv) with a $3\times3\times3$ filter $\left(\mathcal{H}(\cdot)=BN-ReLU-Conv\left(3\times3\times3\right)\right)$. Growth rate is set to $k=12$. Transition layers consist of a bottleneck $\left(BN-ReLU-Conv\left(1\times1\times1\right)\right)$ that halves the number of feature maps, followed by a decrease (MB, CB) or increase (SB) of the feature maps' size.

In the main branch, a $Conv\left(7\times7\times7\right)$ with stride 2 is applied in the x-y-plane to generate 64 feature maps from the input $\mathbf{x}$. This is followed by three DenseBlocks, with 6, 6, and 12 layers, separated by average pooling layers that each half the size of the feature maps' first two dimensions.

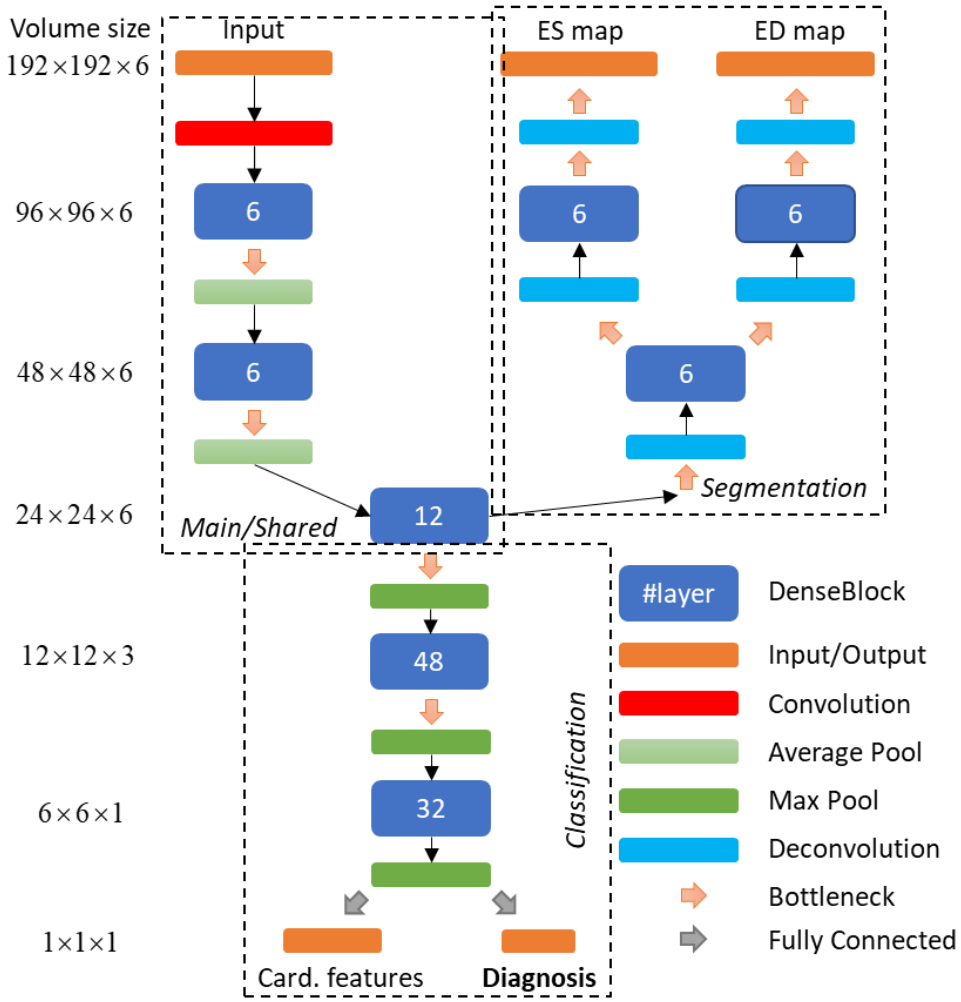

Figure 5: Architecture of the network consists of three branches, each adapted to its own task. The main branch (MB) is shared by all three training tasks. Input to the model during training are six consecutive slices of a volume selected from a random starting slice. The middle six slices are evaluated during testing to obtain deterministic results.

Branch	Layer	Feature maps		Operation
		Size	Amount	
Main	Input	$192\times192\times6$	3	
	Convolution	$96\times96\times6$	64	$Conv(7\times7\times7), stride\ 2$
	DenseBlock	$96\times96\times6$	136	$\left[\mathcal{H}(\cdot)\right]\times6$
	Transition	$96\times96\times6$	68	*Bottleneck*
		$48\times48\times6$	68	$AveragePool(2\times2\times1)$
	DenseBlock	$48\times48\times6$	140	$\left[\mathcal{H}(\cdot)\right]\times6$
	Transition	$48\times48\times6$	70	*Bottleneck*
		$24\times24\times6$	70	$AveragePool(2\times2\times1)$
	DenseBlock	$24\times24\times6$	214	$\left[\mathcal{H}(\cdot)\right]\times12$
Classification	Transition	$24\times24\times6$	107	*Bottleneck*
		$12\times12\times3$	107	$MaxPool(2\times2\times2)$
	DenseBlock	$12\times12\times3$	683	$\left[\mathcal{H}(\cdot)\right]\times48$
	Transition	$12\times12\times3$	341	*Bottleneck*
		$6\times6\times1$	341	$MaxPool(2\times2\times3)$
	DenseBlock	$6\times6\times1$	725	$\left[\mathcal{H}(\cdot)\right]\times32$
	Transition linear	$1\times1\times1$	725	$MaxPool(6\times6\times1)$
	Cardiac features	*Linear*	13	$FC(725\rightarrow13)$
	Diagnosis	*Linear*	5	$FC(725\rightarrow5)$
Segmentation	Transition	$24\times24\times6$	107	*Bottleneck*
		$48\times48\times6$	107	*DeConv*
	DenseBlock	$48\times48\times6$	179	$\left[\mathcal{H}(\cdot)\right]\times6$
	From here separate branch for ES and ED			
	Transition	$48\times48\times6$	89	*Bottleneck*
		$96\times96\times6$	89	*DeConv*
	DenseBlock	$96\times96\times6$	161	$\left[\mathcal{H}(\cdot)\right]\times6$
	Transition	$96\times96\times6$	80	*Bottleneck*
		$192\times192\times6$	80	*DeConv*
	Segmentation	$192\times192\times6$	4	*Bottleneck*

Table 2: Overview of network architecture separated in the three branches; main (MB), classification (CB), and segmentation (SB). CB and SB both connect to last layer in MB. Growth rate is k=12. FC at the end of CB refers to fully connected layer.

In the segmentation branch, transpose convolutions $(DeConv)$ are learned in the transition layer to undo the down sampling. Three transpose convolutions are applied to obtain the original volume size, separated by two 6-layer DenseBlocks. The model splits in two symmetrical branches after the first block to obtain a softmax probability map for ED $\mathbf{p}_{ED}$ and ES $\mathbf{p}_{ES}$ from the triple channel input.

In the classification branch, transition layers switch to max pooling to extract the strongest features and now also reduce feature map size in de slice (z-)direction. Three transition layers are separated by two DenseBlocks with 48 and 32 layers respectively. Empirical testing showed large number of layers are required to improve on random diagnosis. The last pooling layer extracts 725 singleton feature maps that are fully connected to the cardiac feature and diagnosis output layers. For cardiac feature prediction, the output layer estimates values of the handcrafted features $\hat{\mathbf{r}}$. For diagnosis, the output is followed by softmax to obtain the class probability vector $\mathbf{p}_y$.

4.4 Training

All input volumes are required to be the same size for architectural reasons. Six consecutive slices were selected at random form a sample during each training iteration, the lowest number of slices for a sample in the dataset. During testing, use of the center six slices ensured a deterministic outcome.

Training was performed with the ADAM solver where learning rate was set to $5 \cdot 10^{-4}$ [72]. Dropout is applied to individual parameters in all branches with probability $p = 0.5$ of a parameter in the network being dropped, and probability of 0.2 for input voxels being dropped [26].

Training loss $\mathcal{L}_{tot}$ is computed as a weighted combination of the losses over de three different outputs.

$$\mathcal{L}_{tot} = (1-\alpha) \cdot \mathcal{L}_{Segmentation} + \alpha \left((1-\beta) \cdot \mathcal{L}_{CardFeat} + \beta \cdot \mathcal{L}_{Diagnosis} \right) \tag{4.1}$$

Where $\mathcal{L}_{Segmentation}, \mathcal{L}_{CardFeat}, \mathcal{L}_{Diagnosis}$ are the losses for their respective task, and α, β are hyperparameters controlling the backflow of information at the end of MB and CB respectively. Values are set to $\alpha = 0.3, \beta = 0.6$.

Segmentation performance was evaluated as a weighted sum of the binary Dice loss for each label [69].

$$\mathcal{L}_{Segmentation} = \sum_{m}^{\mathcal{M}} w_m \left(1 - \frac{2 \sum_i^N v_{i,m} g_{i,m}}{\sum_i^N v_{i,m}^2 + \sum_i^N g_{i,m}^2} \right) \tag{4.2}$$

Where $v_{i,m} \in \{\mathbf{p}_{ED}, \mathbf{p}_{ES}\}$ and $g_{i,m} \in \{0,1\}$ are the probability and binary ground truth for a given label and voxel combination, N the number of voxels in a volume, and w_m the label weight. Background is assigned weight 0.1, other labels are assigned weight 0.3 so $\sum_m^{|\mathcal{M}|} w_m = 1$.

Estimation of the cardiac features is evaluated using the mean squared error.

$$\mathcal{L}_{CardFeat} = \frac{1}{R} \left\| \hat{\mathbf{r}} - \mathbf{r} \right\|^2 \tag{4.3}$$

Diagnosis is evaluated using the cross-entropy loss with $p_{y_j} \in \mathbf{p}_y$ the probability corresponding to the ground truth class label.

$$\mathcal{L}_{Diagnosis} = -\log\left(\frac{e^{p_{y_i}}}{\sum_j^y e^{p_{y_j}}}\right) \tag{4.4}$$

4.5 Experiments

Four experiments are performed to evaluate the change in diagnostic performance due to combined training with segmentation and prediction of cardiac features, Table 3. Baseline diagnosis and segmentation experiments are performed by eliminating the SB and CB from the model respectively. Baseline experiments were also used to find optimal settings for each task. In the double task learning model, optimal settings from segmentation and diagnosis are combined. In the triple task learning model, prediction of cardiac features is added to the learning. In all but the baseline segmentation model, accurate diagnosis was the objective. Mini-batch size was limited to 4 by the memory requirements of the deconvolution layers. When removing SB, mini-batch size was fixed to 20% of training data to benefit from stochastic effects of learning.

Model name	Diagnosis	Segmentation	Cardiac features	Mini-batch size
Baseline diagnosis	✓	×	×	15
Baseline segmentation	×	✓	×	4
Double task learning	✓	✓	×	4
Triple task learning	✓	✓	✓	4

Table 3: Overview of tasks included for learning in each experiment. Mini-batch size was limited due to high memory usage of SB. In baseline diagnosis, batch-size was fixed to 20% of training data (n=15).

5 Results

5.1 Segmentation

Figure 6 shows the segmentation results of a single slice for two patients and the corresponding ground truths masks. Shown images are examples of typical good and bad segmentation results for the baseline method and the double task method. Dice coefficients and Hausdorff distances for the three methods performing segmentation are shown in Table 4 – together with the best results reported in the challenge. Our data is only evaluated on the center six slices of each case. As apical and basal slices are harder to segment, shown results are an overestimation of performance. Baseline segmentation results are marginally worse than state-of-the-art for LV and Myo. RV results are less accurate, with a doubling of the Hausdorff distance and reduction in the dice index of 0.05 to 0.90 and 0.09 to 0.82 for ED and ES respectively. Adding other tasks to the learning process significantly reduces segmentation results. Dice scores reduce by 0.22 on average and Hausdorff increases by a factor of up to 6.4 when trained jointly with diagnosis and cardiac feature prediction. In all but one case, LV and RV Dice scores are higher for ED than for ES while Myo shows the opposite trend. As segmentation errors mostly occur on object boundaries, this pattern might be due to an opposite change in volume to surface ratio of the three areas when the heart contracts. On visual inspection, the baseline method shows decent segmentation for nearly all cases, while combined training with diagnosis and cardiac feature prediction results in anatomically impossible results even in better performing cases.

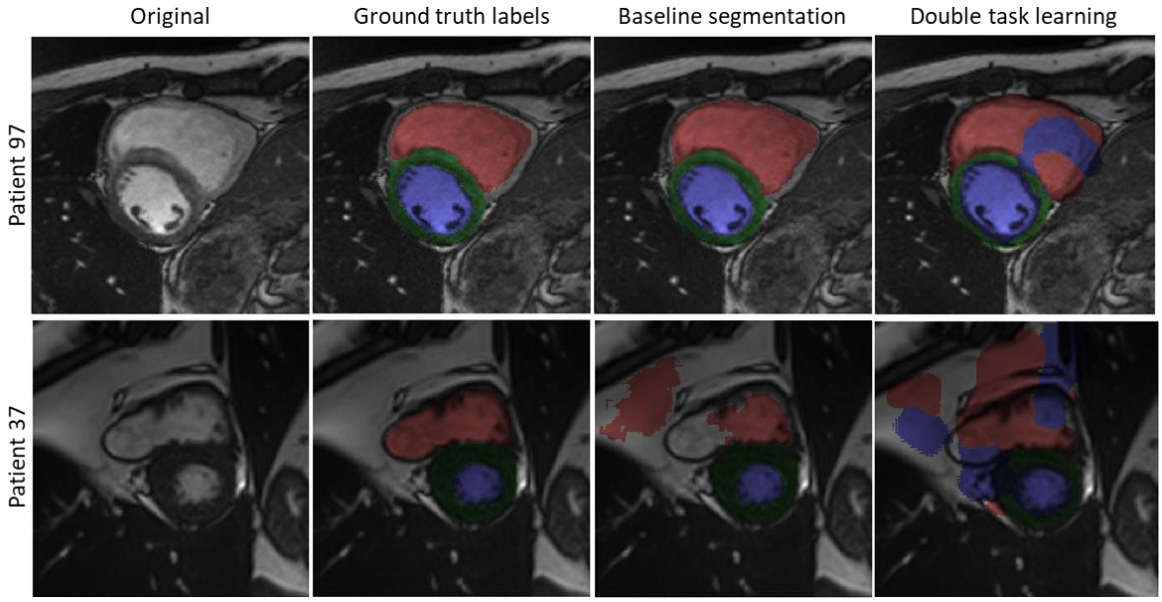

Figure 6: Selected segmentation results of a single frame of two patients for baseline and double task (combined segmentation and diagnosis) methods. Top row representative for methods' best results, bottom for worst.

	Dice						Hausdorff (mm)					
	LV		RV		Myo		LV		RV		Myo	
	ED	ES	ED	ES	ED	ES	ED	ES	ED	ES	ED	ES
Isensee et al. [30]	0.97	0.93	0.95	0.91	0.90	0.92	7.4	6.9	10.1	12.1	8.7	8.7
Baseline segm.	0.96	0.92	0.90	0.82	0.86	0.88	8.6	12.1	23.4	19.8	6.4	12.0
Double task	0.70	0.66	0.75	0.66	0.65	0.64	13.5	14.1	43.3	38.4	16.1	16.6
Triple task	0.72	0.58	0.77	0.71	0.45	0.63	50.8	77.7	51.7	27.0	22.0	65.0

Table 4: Segmentation results measured by dice similarity coefficient and Hausdorff distance for the three experiments involving segmentation and the best challenge entry.

5.2 Diagnosis

Figure 7 shows the training and validation diagnostic classification error during training for the three experiments involving diagnosis. Of the three methods, the combined segmentation and diagnosis approach achieves overall best results. It converges fastest and has the lowest error. Addition of cardiac feature prediction is still an improvement on single task learning, but not as good as the double task. The triple task approach converges considerably slower than the other two. One could argue that the triple method has not converged yet, however, shortly after the shown range, all three methods start to destabilize and default to predicting a single class for all cases at around 1500 iterations. Diagnosis prediction is stable for all three methods around 1100 iterations where they misclassify 3, 7, and 9 out of 25 cases. This corresponds to an accuracy of 64%, 88%, and 72% for the single, double, and triple task learning approach respectively, Table 5. Figure 8 shows the misdiagnoses per type for the double task learning approach and the best diagnosis results in the challenge. All incorrect diagnoses in our model involve the RV while the majority of misdiagnoses in the challenge could be contributed to switching dilated cardiomyopathy and myocardial infarctions cases. Single and triple task errors showed no misclassification pattern.

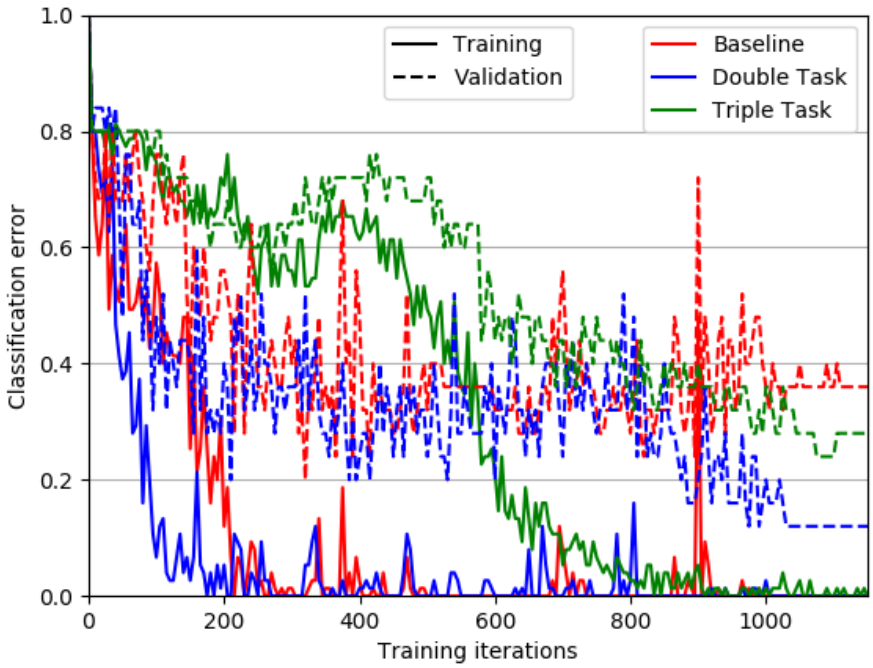

Figure 7: Evolution of diagnostic performance during training for the three methods involving diagnosis. Validation set contains 25 patients (i.e. classification error increases 4% per incorrect diagnosis).

	Method	Diagnostic accuracy
Ours	Baseline	64%
	Double task	88%
	Triple task	72%
Challenge	Khened et al.* [36]	96%
	Cetin et al. [42]	92%
	Isensee et al. [30]	92%
	Wolterink et al. [38]	86%

*Table 5: Comparison of diagnostic accuracy for our deep learning based approach on a subset of the provided training data and the results obtained in the challenge on the provided test set. *Improved their results to 100% after the challenge.*

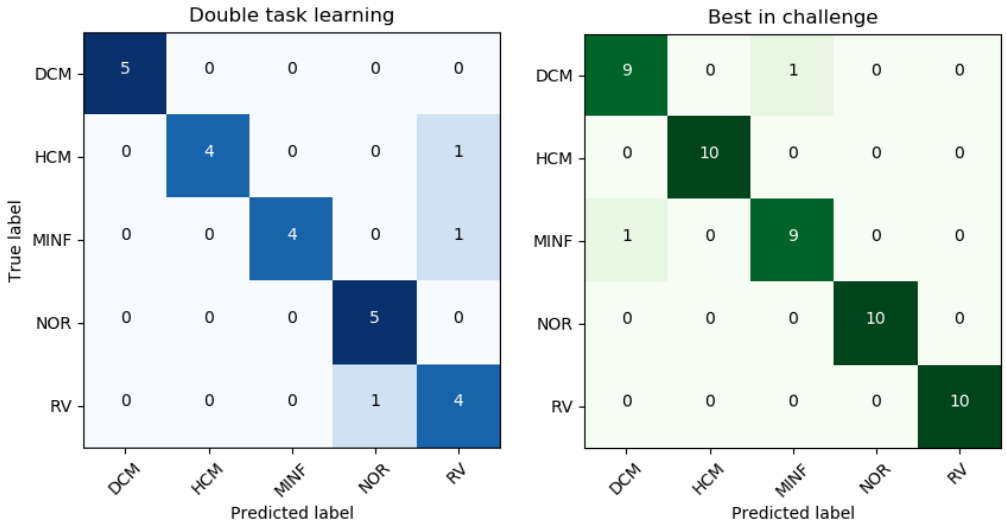

Figure 8: Confusion matrices with diagnosis results for our best method and the best entry in the challenge (adapted from [12]).

6 Discussion and Conclusions

We applied several methods for end-to-end learning of cardiac diagnosis. Multitask learning, where segmentation and diagnosis are trained jointly, produced the highest diagnostic accuracy (88%). All misclassifications involved the RV, suggesting that our model does not accurately capture the concept of an RV. If compared to the challenge results, our best method performs similar to the mid-range entries in the challenge. This indicates that end-to-end learning is feasible for diagnostic purposes, even on small datasets with just 15 training cases per class. However, we note that the dataset contains cases of above average image quality, hand-picked for the challenge. Generalizing ability to a realistic dataset that includes diagnostic boundary cases and image artifacts or other quality issues needs to be investigated.

The inferior performance of diagnosis in the triple task learning method, that includes cardiac feature prediction, is contributed to suboptimal network architecture. In its current implementation, the diagnosis and feature prediction output layers are directly connected to the same layer. Both output layers therefore compete in 'teaching' relevant features to the shared previous layer. In future work it will be investigating if performance increases if more distance is created between the two output layers. This would allow the model to learn features relevant to both tasks in the shared part of the network while finetuning the features in the layers unique to each task.

Segmentation results show that our U-net like structure is capable of accurate segmentation when the classification branch is removed. Performance is slightly less than the best method in the challenge that used an ensemble method to generate segmentation maps. Skip-connections are omitted by design to prevent backpropagation of errors bypassing the classification branch. Adding the skip-connections could bring results up to the same level as challenge entries. LV and Myo scores are marginally worse that best in the challenge, however, RV segmentation performance is significantly worse. The short-axis used for image acquisition is defined in the LV, resulting in a consistent appearance of the LV in images. RV appearance in the images shows more variation, possibly explaining consistent worse performance compared to LV. Including data augmentation by random cropping and rotation of volumes artificially increases the information available on the RV and could improve segmentation. As a result, the model could learn a better concept of the RV, possibly leading to a reduction of diagnosis misclassifications.

In a multitask setting, segmentation results decrease to below a clinically acceptable standard. Unless the feature maps in the shared branch are unique to either the segmentation or classification branch, it is expected that the crude update of a single prediction in diagnosis will partially undo the refined segmentation update coming from $4.4*10^5$ voxels per patient. That this effect is indeed present shows that the segmentation task updates features that are also used for diagnosis, thereby guiding the learning process of diagnosis. The crude update of diagnosis would normally require a smaller learning rate, however, results here use the same value for all methods and show that combined learning with segmentation converges faster using the same learning rate and a batch size that nearly 4 times smaller.

Direct comparison of the results with the challenge is unjust as our results are evaluated on a subset of the training data while challenge results are evaluated on a separate test set. Furthermore, as no test set was used, current result may be a product of overfitting on the validation set. Cross-validation could be performed to increase certainty about results, however, improving the preprocessing pipeline to automatically detect the heart for center cropping would allow our method to be applied to the test set for a fairer comparison.

Segmentation evaluation also needs to be improved for a fair comparison with challenge results. First, segmentation is currently only evaluated on the center six slices, excluding basal and/or apical slices in the majority of cases. Slices at the base and apex have proven to be harder to segment for both expert and machine [12, 40]. Second, predicted segmentation maps are evaluated using resampled versions of the ground truth. While mostly accurate, resampling has resulted in small changes at the ground truth boundary between objects. Resampling of the softmax segmentation maps to the original resolution followed by evaluation on original ground truth maps would therefore provide a better comparison. As accurate diagnosis was the objective of this study, no further time was invested in segmenting accuracy or fair comparison to challenge results.

In clinical practice, the diseases present in this dataset are diagnosed based on the cardiac features that are predicted, see Table 1. It is therefore unsurprising that these features can be used for automated diagnosis when segmentation results are accurate, as proven by the 100% accuracy obtained on the dataset [16]. However, the cut-off values for diagnosis are based on population averages and therefore do no justice to the individual. Clinicians can take other factors (e.g. patient's size, or relationship between posture and cardiac anatomy) in to consideration when diagnosing a patient that are not captured in the statically defined cardiac features. We argue that deep learning based methods may be able to learn these factors. In time, deep learning could then outperform rule based methods in cardiac diagnosis and may provide new imaging biomarkers for diagnosis and CVD management. We therefore welcome new challenges, such as the left ventricle full quantification challenge[4], that focus on direct measurement on (cardiac) images instead of extracting features indirectly via segmentation.

[4] lvquan18.github.io

Bibliography

1. World Health Organization. *WHO Mortality Database*. 2017; Available from: www.who.int/healthinfo/mortality_data/en/.
2. Mathers, C.D. and D. Loncar, *Projections of global mortality and burden of disease from 2002 to 2030.* PLoS medicine, 2006. **3**(11): p. e442.
3. Murray, C.J., et al., *Disability-adjusted life years (DALYs) for 291 diseases and injuries in 21 regions, 1990–2010: a systematic analysis for the Global Burden of Disease Study 2010.* The lancet, 2012. **380**(9859): p. 2197-2223.
4. Salerno, M., et al., *Recent advances in cardiovascular magnetic resonance: techniques and applications.* Circulation: Cardiovascular Imaging, 2017. **10**(6): p. e003951.
5. Peng, P., et al., *A review of heart chamber segmentation for structural and functional analysis using cardiac magnetic resonance imaging.* Magnetic Resonance Materials in Physics, Biology and Medicine, 2016. **29**(2): p. 155-195.
6. Attili, A.K., et al., *Quantification in cardiac MRI: advances in image acquisition and processing.* Int J Cardiovasc Imaging, 2010. **26**(1): p. 27-40.
7. Medrano-Gracia, P., et al., *Challenges of cardiac image analysis in large-scale population-based studies.* Current cardiology reports, 2015. **17**(3): p. 9.
8. Russakovsky, O., et al., *Imagenet large scale visual recognition challenge.* International Journal of Computer Vision, 2015. **115**(3): p. 211-252.
9. He, K., et al. *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification*. in *Proceedings of the IEEE international conference on computer vision*. 2015.
10. Esteva, A., et al., *Dermatologist-level classification of skin cancer with deep neural networks.* Nature, 2017. **542**(7639): p. 115.
11. Gulshan, V., et al., *Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs.* Jama, 2016. **316**(22): p. 2402-2410.
12. Bernard, O., et al., *Deep Learning Techniques for Automatic MRI Cardiac Multi-structures Segmentation and Diagnosis: Is the Problem Solved?* IEEE Transactions on Medical Imaging, 2018.
13. He, K., et al., *Mask r-cnn.* IEEE transactions on pattern analysis and machine intelligence, 2018.
14. Zreik, M., et al., *Deep learning analysis of the myocardium in coronary CT angiography for identification of patients with functionally significant coronary artery stenosis.* Medical image analysis, 2018. **44**: p. 72-85.
15. Litjens, G., et al., *A survey on deep learning in medical image analysis.* Medical image analysis, 2017. **42**: p. 60-88.
16. Khened, M., V.A. Kollerathu, and G. Krishnamurthi, *Fully Convolutional Multi-scale Residual DenseNets for Cardiac Segmentation and Automated Cardiac Diagnosis using Ensemble of Classifiers.* arXiv preprint arXiv:1801.05173, 2018.
17. Dhungel, N., G. Carneiro, and A.P. Bradley. *The automated learning of deep features for breast mass classification from mammograms*. in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 2016. Springer.
18. Goodfellow, I., et al., *Deep learning*. Vol. 1. 2016: MIT press Cambridge.
19. Schmidhuber, J., *Deep learning in neural networks: An overview.* Neural networks, 2015. **61**: p. 85-117.
20. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* nature, 2015. **521**(7553): p. 436.

21. Shen, D., G. Wu, and H.-I. Suk, *Deep learning in medical image analysis.* Annual review of biomedical engineering, 2017. **19**: p. 221-248.
22. Rosenblatt, F., *The perceptron: a probabilistic model for information storage and organization in the brain.* Psychological review, 1958. **65**(6): p. 386.
23. Nair, V. and G.E. Hinton. *Rectified linear units improve restricted boltzmann machines*. in *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010.
24. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors.* nature, 1986. **323**(6088): p. 533.
25. Gu, J., et al., *Recent advances in convolutional neural networks.* Pattern Recognition, 2018. **77**: p. 354-377.
26. Srivastava, N., et al., *Dropout: a simple way to prevent neural networks from overfitting.* The Journal of Machine Learning Research, 2014. **15**(1): p. 1929-1958.
27. Goodfellow, I., et al. *Generative adversarial nets*. in *Advances in neural information processing systems*. 2014.
28. Caruana, R., *Multitask learning.* Machine learning, 1997. **28**(1): p. 41-75.
29. Grinias, E. and G. Tziritas. *Fast Fully-Automatic Cardiac Segmentation in MRI Using MRF Model Optimization, Substructures Tracking and B-Spline Smoothing*. in *International Workshop on Statistical Atlases and Computational Models of the Heart*. 2017. Springer.
30. Isensee, F., et al. *Automatic cardiac disease assessment on cine-mri via time-series segmentation and domain specific features*. in *International Workshop on Statistical Atlases and Computational Models of the Heart*. 2017. Springer.
31. Patravali, J., S. Jain, and S. Chilamkurthy. *2D-3D fully convolutional neural networks for cardiac MR segmentation*. in *International Workshop on Statistical Atlases and Computational Models of the Heart*. 2017. Springer.
32. Baumgartner, C.F., et al. *An exploration of 2D and 3D deep learning techniques for cardiac MR image segmentation*. in *International Workshop on Statistical Atlases and Computational Models of the Heart*. 2017. Springer.
33. Zotti, C., et al. *GridNet with automatic shape prior registration for automatic MRI cardiac segmentation*. in *International Workshop on Statistical Atlases and Computational Models of the Heart*. 2017. Springer.
34. Yang, X., et al. *Class-Balanced Deep Neural Network for Automatic Ventricular Structure Segmentation*. in *International Workshop on Statistical Atlases and Computational Models of the Heart*. 2017. Springer.
35. Jang, Y., et al. *Automatic Segmentation of LV and RV in Cardiac MRI*. in *International Workshop on Statistical Atlases and Computational Models of the Heart*. 2017. Springer.
36. Khened, M., V. Alex, and G. Krishnamurthi. *Densely Connected Fully Convolutional Network for Short-Axis Cardiac Cine MR Image Segmentation and Heart Diagnosis Using Random Forest*. in *International Workshop on Statistical Atlases and Computational Models of the Heart*. 2017. Springer.
37. Rohé, M.-M., M. Sermesant, and X. Pennec. *Automatic Multi-Atlas Segmentation of Myocardium with SVF-Net*. in *International Workshop on Statistical Atlases and Computational Models of the Heart*. 2017. Springer.
38. Wolterink, J.M., et al. *Automatic segmentation and disease classification using cardiac cine MR images*. in *International Workshop on Statistical Atlases and Computational Models of the Heart*. 2017. Springer.

39. Petersen, S.E., et al., *Imaging in population science: cardiovascular magnetic resonance in 100,000 participants of UK Biobank-rationale, challenges and approaches.* Journal of Cardiovascular Magnetic Resonance, 2013. **15**(1): p. 46.
40. Bai, W., et al., *Automated cardiovascular magnetic resonance image analysis with fully convolutional networks.* 2018.
41. Lambin, P., et al., *Radiomics: extracting more information from medical images using advanced feature analysis.* European journal of cancer, 2012. **48**(4): p. 441-446.
42. Cetin, I., et al. *A Radiomics Approach to Computer-Aided Diagnosis with Cardiac Cine-MRI*. in *International Workshop on Statistical Atlases and Computational Models of the Heart*. 2017. Springer.
43. Carneiro, G., J.C. Nascimento, and A. Freitas, *The segmentation of the left ventricle of the heart from ultrasound data using deep learning architectures and derivative-based search methods.* IEEE Transactions on Image Processing, 2012. **21**(3): p. 968-982.
44. Carneiro, G. and J.C. Nascimento, *Combining multiple dynamic models and deep learning architectures for tracking the left ventricle endocardium in ultrasound data.* IEEE transactions on pattern analysis and machine intelligence, 2013. **99**(1): p. 1.
45. Nascimento, J.C. and G. Carneiro. *Multi-atlas segmentation using manifold learning with deep belief networks*. in *Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on*. 2016. IEEE.
46. Ngo, T.A., Z. Lu, and G. Carneiro, *Combining deep learning and level set for the automated segmentation of the left ventricle of the heart from cardiac cine magnetic resonance.* Medical image analysis, 2017. **35**: p. 159-171.
47. Lessmann, N., et al. *Deep convolutional neural networks for automatic coronary calcium scoring in a screening study with low-dose chest CT*. in *Medical Imaging 2016: Computer-Aided Diagnosis*. 2016. International Society for Optics and Photonics.
48. Wolterink, J.M., et al., *Automatic coronary artery calcium scoring in cardiac CT angiography using paired convolutional neural networks.* Medical image analysis, 2016. **34**: p. 123-136.
49. Gülsün, M.A., et al. *Coronary centerline extraction via optimal flow paths and CNN path pruning*. in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 2016. Springer.
50. Moradi, M., et al. *A hybrid learning approach for semantic labeling of cardiac CT slices and recognition of body position*. in *Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on*. 2016. IEEE.
51. de Vos, B.D., et al. *2D image classification for 3D anatomy localization: employing deep convolutional neural networks*. in *Medical Imaging 2016: Image Processing*. 2016. International Society for Optics and Photonics.
52. Noothout, J.M., et al., *CNN-based Landmark Detection in Cardiac CTA Scans.* arXiv preprint arXiv:1804.04963, 2018.
53. Zreik, M., et al. *Automatic segmentation of the left ventricle in cardiac CT angiography using convolutional neural networks*. in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. 2016. IEEE.
54. Tran, P.V., *A fully convolutional neural network for cardiac segmentation in short-axis MRI.* arXiv preprint arXiv:1604.00494, 2016.
55. Rupprecht, C., et al., *Deep active contours.* arXiv preprint arXiv:1607.05074, 2016.
56. Avendi, M., A. Kheradvar, and H. Jafarkhani, *A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac MRI.* Medical image analysis, 2016. **30**: p. 108-119.

57. Poudel, R.P., P. Lamata, and G. Montana, *Recurrent fully convolutional neural networks for multi-slice MRI cardiac segmentation*, in *Reconstruction, Segmentation, and Analysis of Medical Images*. 2016, Springer. p. 83-94.
58. Shin, H.-C., et al., *Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data.* IEEE transactions on pattern analysis and machine intelligence, 2013. **35**(8): p. 1930-1943.
59. Zhou, X., et al., *Three-dimensional CT image segmentation by combining 2D fully convolutional network with 3D majority voting*, in *Deep Learning and Data Labeling for Medical Applications*. 2016, Springer. p. 111-120.
60. Emad, O., I.A. Yassine, and A.S. Fahmy. *Automatic localization of the left ventricle in cardiac MRI images using deep learning*. in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. 2015. IEEE.
61. Zhang, L., et al. *Automated quality assessment of cardiac MR images using convolutional neural networks*. in *International Workshop on Simulation and Synthesis in Medical Imaging*. 2016. Springer.
62. Oktay, O., et al. *Multi-input cardiac image super-resolution using convolutional neural networks*. in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 2016. Springer.
63. Kong, B., et al. *Recognizing end-diastole and end-systole frames via deep temporal regression network*. in *International conference on medical image computing and computer-assisted intervention*. 2016. Springer.
64. Xue, W., et al., *Full left ventricle quantification via deep multitask relationships learning.* Medical image analysis, 2018. **43**: p. 54-65.
65. Xue, W., et al. *Full quantification of left ventricle via deep multitask learning network respecting intra-and inter-task relatedness*. in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 2017. Springer.
66. Bello, G.A., et al., *Using Three-Dimensional Cardiac Motion for Predicting Mortality in Pulmonary Hypertension: A Deep Learning Approach.* 2018.
67. Katzman, J.L., et al., *Deep survival: A deep cox proportional hazards network.* stat, 2016. **1050**: p. 2.
68. Krizhevsky, A., I. Sutskever, and G.E. Hinton. *Imagenet classification with deep convolutional neural networks*. in *Advances in neural information processing systems*. 2012.
69. Ronneberger, O., P. Fischer, and T. Brox. *U-net: Convolutional networks for biomedical image segmentation*. in *International Conference on Medical image computing and computer-assisted intervention*. 2015. Springer.
70. Huang, G., et al. *Densely Connected Convolutional Networks*. in *CVPR*. 2017.
71. Ioffe, S. and C. Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. in *International Conference on Machine Learning*. 2015.
72. Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980, 2014.