Densely Connected Fully Convolutional Network for Short-Axis Cardiac Cine MR Image Segmentation and Heart Diagnosis Using Random Forest

Mahendra Khened, Varghese Alex, and Ganapathy Krishnamurthi(✉)

Indian Institute of Technology Madras, Chennai 600036, Tamil Nadu, India
gankrish@iitm.ac.in

Abstract. In this paper, we propose a fully automatic method for segmentation of left ventricle, right ventricle and myocardium from cardiac Magnetic Resonance (MR) images using densely connected fully convolutional neural network. Dense Convolutional neural network (DenseNet) facilitates multi-path flow for gradients between layers during training by back-propagation and feature propagation. DenseNet also encourages feature reuse & thus substantially reduces the number of parameters while maintaining good performance, which is ideal in scenarios with limited data. The training data was subjected to Fourier analysis and classical computer vision (CV) techniques for Region of Interest (ROI) extraction. The parameters of the network were optimized by training with a dual cost function i.e. weighted cross-entropy and Dice co-efficient. For the task of automated heart diagnosis, cardiac parameters such as ejection fraction, volumes of ventricles etc. where calculated from segmentation masks predicted by the network at the end systole and diastole phases. Further these parameters were used as features to train a Random forest classifier. On the exclusively held-out test set (10% of training set) the proposed method for segmentation task achieved a mean dice score of 0.92, 0.87 and 0.86 for left ventricle, right ventricle and myocardium respectively. For automated cardiac disease diagnosis, the Random Forest classifier achieved an accuracy of 90%.

Keywords: Cardiac MRI · Segmentation · CNN · FCN · DenseNet Inception · Dice loss

1 Introduction and Related Work

In clinical practice, MRI is preferred over ultrasound and CT due to its superior spatial-temporal resolution and non-ionizing radiation. Cardiac parameters such as left ventricular ejection fraction, volumes of the left ventricle and right ventricle, myocardial thickness are calculated routinely to diagnose a subject as healthy or diseased. For the aforementioned reason, segmentation of the structures such as left ventricle, right ventricle and myocardium from MR images

© Springer International Publishing AG, part of Springer Nature 2018
M. Pop et al. (Eds.): STACOM 2017, LNCS 10663, pp. 140–151, 2018.
https://doi.org/10.1007/978-3-319-75541-0_15

becomes a pivotal step. Manual segmentation of the structures from the surrounding tissues is a tedious task and often introduces inter-rater variability.

Convolutional neural networks (CNNs) [1] have been applied to wide variety of pattern recognition tasks, most common ones are image classification [2–4] and semantic segmentation using fully convolutional networks (FCN) [5]. CNNs have also been applied to medical image segmentation and classification [7,8]. In this paper, we propose a CNN based architecture for segmentation of the left ventricle, right ventricle and myocardium from short-axis view of cardiac MR images. Our network's connectivity pattern was inspired from DenseNets [10]. DenseNets connects each layer to every other layer in a feed-forward fashion by concatenation of all feature outputs. The output of the l^{th} layer is defined as

$$x_l = H_l([x_{l-1}, x_{l-2}, \cdots, x_0]) \tag{1}$$

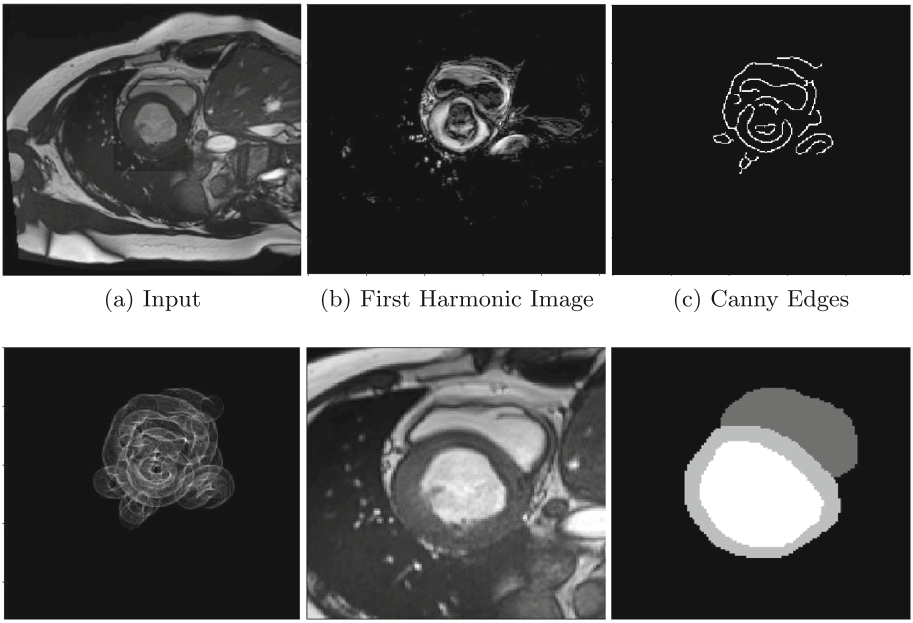
where x_l represents the feature maps at the l^{th} layer and $[\cdots]$ represents the concatenation operation. In our case, H is the layer comprising of Batch Normalization (BN) [22], followed by Exponential Linear Unit (ELU) [23], a convolution and dropout [21]. This kind of connectivity pattern aids in reuse of features and allows implicit deep supervision during training. The output dimension of each layer has k (growth rate parameter) feature maps. The number of feature maps in DenseNets grow linearly with depth. A Transition Down layer in DenseNets is introduced for reducing spatial dimension of feature maps which is accomplished by using a 1×1 convolution (depth preserving) followed by a 2×2 max-pooling operation. A Dense-Block refers to concatenation of new feature maps created at a given resolution.

2 Our Method

2.1 Data Pre-processing Pipeline

Region of Interest (ROI) Detection. The cardiac MR images of the patient comprises of the heart and various surrounding structures like the lungs and diaphragm. Since the task at hand was segmentation of various heart structures, an automated method for region of interest detection was carried out to delineate the heart structures from the surrounding tissues. The Fourier based techniques [13], employ the fact that each slice sequence in time captures one heartbeat. Fourier analysis was done to extract first harmonic images which captured the maximal activity at the corresponding heartbeat frequency. Assuming that the left ventricle approximates a circle, the first harmonic images were subjected to canny edge detector. The approximate radius & center of the left ventricle were calculated from the edge maps using circular Hough transform approach [14]. Figure 1 shows the extraction of ROI using the proposed technique.

Data Augmentation. Data augmentation was done to artificially increase the size of the dataset. Pixel-Spacing information was used to rescale the images to 1 mm spacing. The ROI detection estimates the approximate center of the left


(a) Input (b) First Harmonic Image (c) Canny Edges

(d) Hough Transform (e) ROI Cropping (f) Ground Truth Cropping

Fig. 1. Fourier based Region of interest (ROI) detection scheme is based on capturing pixel regions where there is maximal intensity variation over one full cardiac cycle. These pixels mostly correspond to ventricular regions of heart. Cropping a patch of fixed size centered around the left ventricle (LV) leads to removal of irrelevant structures. The steps involved in ROI detection are (a) Temporal slices, (b) Estimation of first harmonic image using Fourier Analysis, (c) Canny edge-detection on the harmonic image, (d) Circular Hough Transform on edge-map to localize LV, (e)–(f) ROI cropping on the input & ground-truth image

ventricle C, further a patch of size 128 × 128 centered around C was extracted from the rescaled image. This method helped in alleviating the huge class-imbalance problem associated with labels for heart structures seen in the full sized cardiac MR images. In addition, the proposed technique enables the network to precisely learn the fine-grained structures of the heart. Most importantly, this approach reduces the computation time required for learning the parameters of network and also during inference. The data augmentation scheme employed were:

- rotation: random angle between -5 and 5° (uniform)
- translation x-axis: random shift between -5 and 5 mm (uniform)
- translation y-axis: random shift between -5 and 5 mm (uniform)
- rescaling: random zoom factor between .6 and 1.4 (uniform)
- horizontal and vertical flipping: yes or no (bernoulli)

Normalization. Each slice of the patient's voxel intensities were normalized to the range of 0–1 using Eq. (2)

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2}$$

where X is voxel intensity.

2.2 Proposed Network Architecture: Densely Connected Fully Convolutional Network (DFCN)

Figure 2 illustrates the schematic diagram of our proposed network for segmentation. The down-sampling and up-sampling components adopts the fully convolutional DenseNets architecture for semantic segmentation as described in [9]. Each layer in the dense block is sequentially composed of BN-ELU and a 3×3 convolution layers. The first Dense-Block was prefixed with a naive version of Inception module [11] comprising of convolution filters of size 3×3, 5×5 and 7×7. In the down-sampling path, the input to a dense block was concatenated with its output, leading to a linear growth of the number of feature maps. The Transition-Down block (TD) consists of BN-ELU a 1×1 convolution and a 2×2 max-pooling layers. The last layer of the down-sampling path is referred to as Bottleneck.

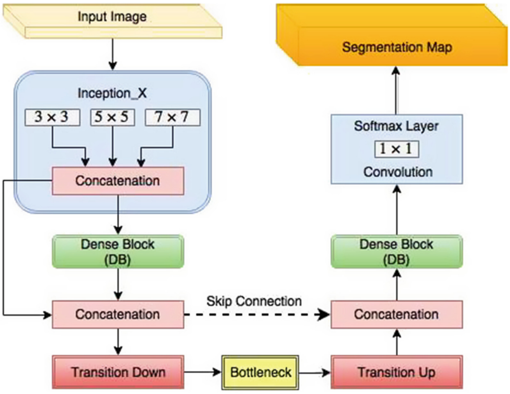


Fig. 2. Architecture of DFCN.

In the up-sampling path, the input of a Dense-Block is not concatenated with its output. Transition-Up (TU) block comprises of 3×3 transposed convolution layer with a stride of 2. The output feature maps of the TU block was concatenated (via skip connection) with the feature maps corresponding to those DB's from down-sampling path. The feature maps of the hindmost up-sampling component was convolved with a 1×1 convolution layer followed by a soft-max layer

to generate the final label map of the segmentation. To prevent over-fitting, a dropout of 0.2 was implemented following each convolution layer.
Table 1 summaries the individual blocks of our architecture. For the segmentation task, the proposed network's architecture is summarized in Table 2. The number of trainable parameters is about 4×10^6 (4M) in total, which is far lesser than number of trainable parameters in U-Net [6] (30M parameters). It was observed that using exponential linear units (ELUs) instead of rectified linear units (ReLUs) led to faster convergence.

Table 1. Building blocks of DFCN. From left to right: layer used in the model, Transition Down (TD) and Transition Up (TU).

Layer
Batch Normalization
Exponential Linear Unit
3×3 Convolution
Dropout $p = 0.2$

TD
Batch Normalization
Exponential Linear Unit
1×1 Convolution
Dropout $p = 0.2$
2×2 Max Pooling

TU
3×3 Transposed Convolution stride $= 2$

Table 2. Architecture details of model used in our experiments. The growth rate parameter $k = 8$

DFCN Architecture
Input_Size: (128×128), channels=1
Inception_X: 3×3 (16), 5×5 (4), 7×7 (4) convolutions, features = 24
Dense Block (3 layers) + Transition Down, features = 48
Dense Block (4 layers) + Transition Down, features = 80
Dense Block (5 layers) + Transition Down, features = 120
BottleNeck (8 layers), features = 176
Transition Up + Dense Block (5 layers), features = 216
TransitionUp + Dense Block (4 layers), features = 144
Transition Up + Dense Block (3 layers), features = 104
1×1 convolution, channels = 4
Softmax Layer

Number of Convolutional layers : 42
Number of parameters : 374292

2.3 Loss Function

The anatomical structures of interest in the medical images are sparsely represented in whole volume. This leads to class imbalance in the dataset, thereby making it hard for the network to learn subtle structures in the region of interest. In order to address this issue, the loss function used, weighting mechanism based on class frequencies. A weighted combination of two loss function, namely:- cross-entropy loss and a loss function based on Dice overlap co-efficient [12] was used to train the network.

The dice co-efficient is an overlap metric used for assessing the quality of segmentation maps. The dice coefficient between two binary volumes can be written as:

$$DICE = \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2} \tag{3}$$

where the sums run over the N voxels, of the predicted binary segmentation volume $p_i \in P$ and the ground truth binary volume $g_i \in G$.

For multi-class problem the dice loss can be written as weighted sum of Eq. 3. The weights are empirically determined based on the their relative class frequencies. The total dice loss for multi-class segmentation problem is given in Eq. (4):

$$dice\_loss = \frac{W_{class1} DICE_{class1} + \cdots + W_{classN} DICE_{classN}}{W_{class1} + \cdots + W_{classN}} \tag{4}$$

where W_{classN} is the empirically assigned weight based on its relative frequency, smaller the frequency higher the assigned weight.

The parameters of the network were optimized so as to minimize the *total_loss*, Eq. (5).

$$total\_loss = \lambda(cross\_entropy\_loss) + \gamma(1 - dice\_loss) + L2\_loss \tag{5}$$

where λ and γ are empirically assigned weights to individual losses. During training it was observed that the Dice loss allowed higher overlap scores than when trained with the loss function based on the cross entropy loss alone. In this work we set $\gamma = 0.75$ and $\lambda = 0.25$.

The proposed model was trained on a batch size of 10 2D-MR images for 200 epochs using ADAM [20] as the optimizer. The learning rate was set to 10^{-4} and additionally a $L2$ weight decay of 10^{-4} was added to the cost function as a regularizer.

2.4 Post-processing

The results of segmentation predicted by DFCN network were subjected to connected component analysis to remove false positives. The largest the component (heart structures) was retained, while the rest were discarded.

2.5 Cardiac Disease Diagnosis

To develop an automated cardiac diagnosis system the following 10 attributes from the training dataset were used:

- Ejection fraction of left ventricle and right ventricle
- Volume of the left ventricle at end systole and end diastoles phases
- Volume of the right ventricle at end systole and end diastole phases
- Mass of the myocardium at end diastole and its volume at end systole
- Patient height and weight

Initially, these 11 attributes were calculated from the training set and were used for training a Random Forest classifier [19]. The proposed Random Forest classifier comprises of 100 trees. On the test set, the segmentation maps predicted from the trained neural network was used for calculating the above listed 11 cardiac parameters. These parameters were fed as input to the trained Random Forest classifier to diagnose the patient as: Dilated cardiomyopathy (DCM), Hypertrophic cardiomyopathy (HCM), Myocardial infarction (MNF), Abnormal right ventricle (ARV) and normal patients (NOR).

3 Experimental Setup and Results

3.1 Dataset and Evaluation Criteria

The network was trained and tested on the ACDC STACOM 2017 challenge dataset, comprising of 100 patients. The patients were divided into 5 evenly distributed groups: dilated cardiomyopathy (DCM), hypertrophic cardiomyopathy (HCM), myocardial infarction (MNF), abnormal right ventricle (ARV) and normal patients (NOR). The end diastolic (ED) and the end systolic (ES) phases come with a pixel-accurate manual delineation by two independent medical experts. The dataset was split into 70 for training, 20 for validation and 10 for testing using stratified sampling (strata for sampling is based on the cardiac disease). In order to gauge performances on the held out test set, we report the clinical metrics such as the average ejection fraction (EF) error, the average left ventricle (LV) and right ventricle (RV) systolic and diastolic volume errors, and the average myocardium (MYO) mass error. For the geometrical metrics, we report the Dice and the Hausdorff distances for all 3 regions at the ED and ES phases. For the cardiac disease diagnosis the metrics used was accuracy, precision and recall.

3.2 Experimental Results

The proposed model was evaluated on the exclusively held-out testing data ($n = 10$). Table 3 shows the average dice scores achieved by the model at ED and ES phases of the heart. The model achieves a relatively higher dice score for LV when compared to RV and MYO. The proposed method relies on localization of the LV and cropping a patch of fixed size from the LV region's center as pre-processing step before feeding into the network. So, in cases of abnormally large RV, the model slightly under-performs when RV region extends beyond the patch size. The aforementioned reasons & irregular shape of RV when compared to LV leads to a dip in dice score & higher Hausdorff distance of RV.

Figures 3 and 4 shows the results of segmentation produced by the proposed network at ED and ES phase of the heart. It was observed that model generates good segmentations throughout the volume. However due to small structures at the base and close proximity of valves such as aorta at the apex regions leads to erroneous segmentation. For example, in Fig. 3(h) myocardium is slightly over-segmented at the apex region of the heart, while in Fig. 4(b) the model does some erroneous segmentation in the basal slices near valves of the heart.

The model's geometric metrics are slightly better at ED phase than at ES phase, whereas the clinical metrics (Tables 4 and 5) namely the LV & RV volume error and MYO mass error are relatively better at ES phase.

Table 6 compares the effect of training a network with proposed loss function as opposed to training with the vanilla cross-entropy. It was observed that the proposed loss function manifested in producing better segmentations when compared to vanilla cross entropy and thus led to improvement of dice score by 2%.

Table 7 shows the result of the cardiac disease diagnosis on the testing data. The Random Forest classifier's accuracy heavily depends on the clinical metrics, which in-turn depends on the segmentation results of the proposed model.

Table 3. Results of geometrical metrics on the testing dataset.

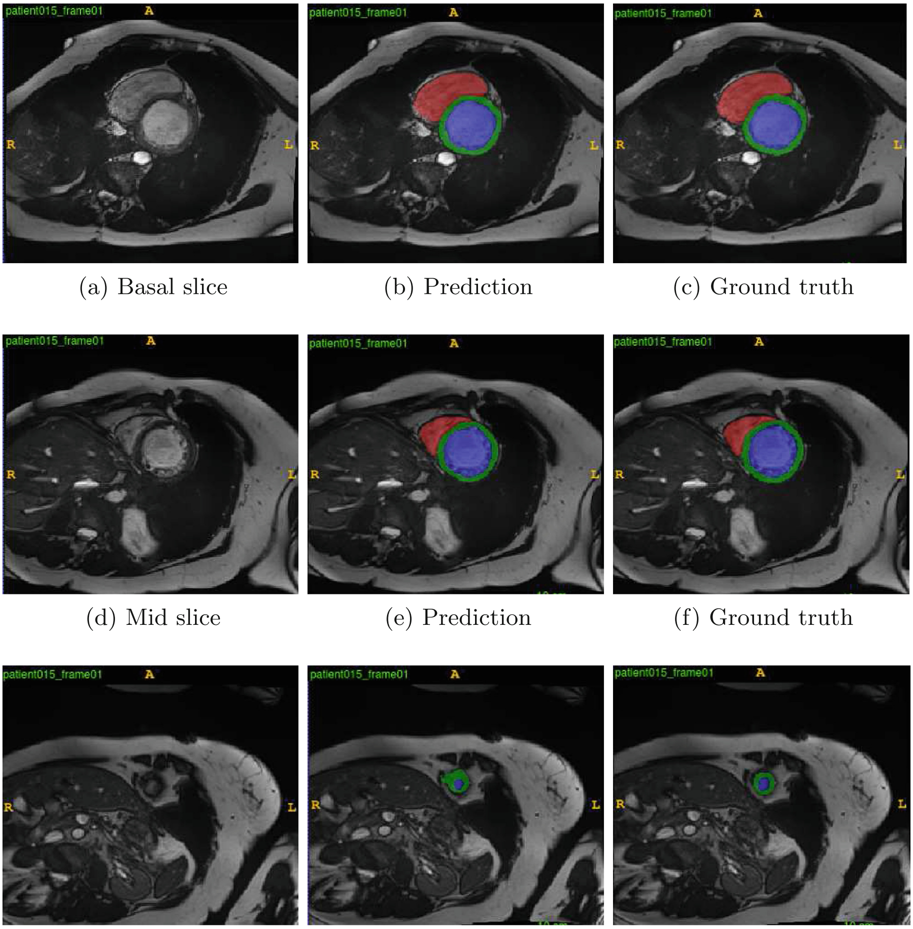
	Cardiac phase					
	End diastole			End systole		
Geometric metric	LV	RV	MYO	LV	RV	MYO
Dice score	.94	.89	.84	.89	.84	.87
Hausdorff distance	12.13	18.97	17.05	12.04	23.97	12.92

Table 4. Results of clinical metrics on the testing dataset.

Ejection fraction error (%)		Left ventricle volume error (mL)		Right ventricle volume error (mL)		MYO mass error (g)	
Left ventricle	Right ventricle	Diastole	Systole	Diastole	Systole	Diastole	Systole
2	11.1	3.2	1.7	9.7	4	14.1	10.6

Table 5. Results of clinical metrics on the testing dataset.

Clinical metric	Ejection fraction (%)		Volume ED (ml)		Volume ES (ml)			Mass ED (g)
	LV	RV	LV	RV	LV	RV	MYO	MYO
Correlation coefficient	0.980	0.889	0.968	0.903	0.983	0.960	0.894	0.859
BIAS	3.77	11.73	9.68	13.26	2.07	−6.98	−4.03	−10.09
LOA	[−6.21; 13.75]	[−9.37; 32.83]	[−11.86; 31.22]	[−28.63; 55.15]	[−27.02; 31.16]	[−29.03; 15.07]	[−39.07; 31.01]	[−48.53; 28.35]



(a) Basal slice (b) Prediction (c) Ground truth

(d) Mid slice (e) Prediction (f) Ground truth

(g) Apex slice (h) Prediction (i) Ground truth

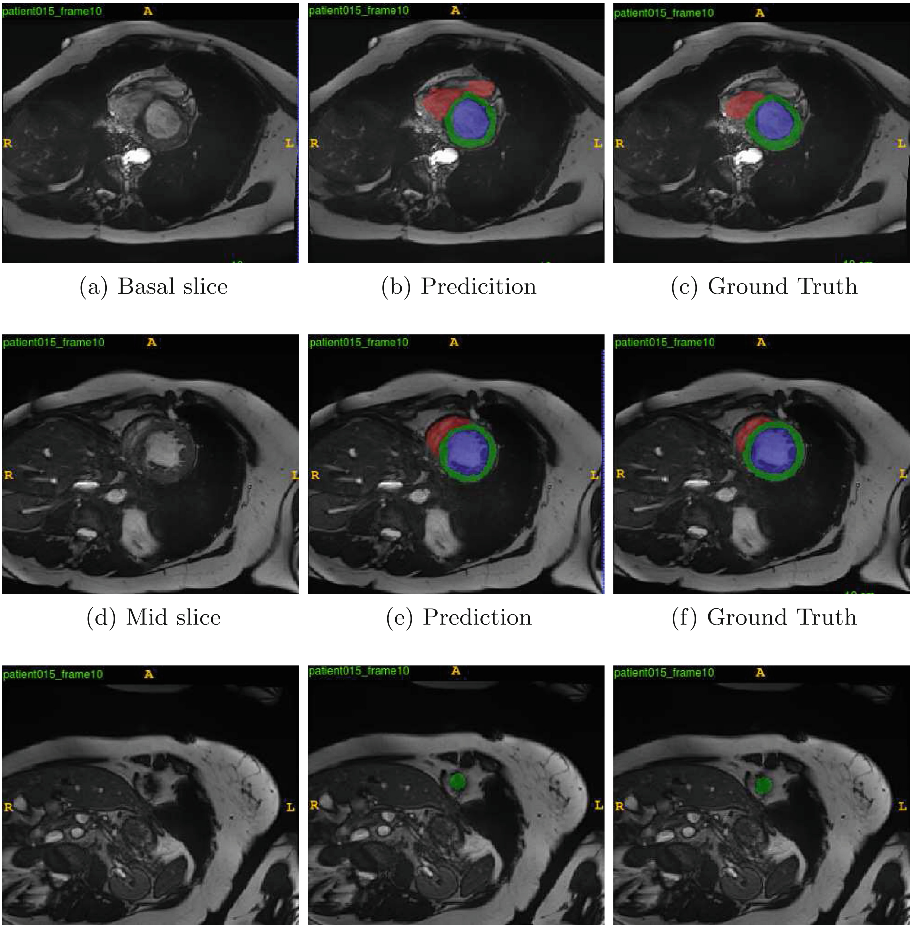
Fig. 3. The figure shows the segmentation results generated by the proposed model on test-set at ED phase of heart. The columns from left to right indicate: the input images, segmentations generated by the model and their associated ground-truths. The rows from top to bottom indicate: short axis slices of the heart at basal, mid and apex. In all figures the colors red, green and blue indicate RV, MYO and LV respectively. (Color figure online)

Table 6. Evaluation comparison for the proposed loss function

	Average dice score		
Heart Structure	Left ventricle	Right ventricle	Myocardium
Proposed loss function	.92	.87	.86
Vanilla cross-entropy loss	.90	.85	.83

Table 7. Results of automated cardiac diagnosis on the testing dataset.

Disease →	NOR	DCM	HCM	MINF	ARV
Recall	1	1	1	1	1
Precision	1	0.67	0.5	1	1
Overall classification accuracy: 0.90					



(a) Basal slice (b) Predicition (c) Ground Truth

(d) Mid slice (e) Prediction (f) Ground Truth

(g) Apex slice (h) Prediction (i) Ground Truth

Fig. 4. The figure shows the segmentation results generated by the proposed model on test-set at ES phase of heart. The columns from left to right indicate: the input images, segmentations generated by the model and their associated ground-truths. The rows from top to bottom indicate: short axis slices of the heart at basal, mid and apex. In all figures the colors red, green and blue indicate RV, MYO and LV respectively. (Color figure online)

3.3 Conclusion

We propose a new CNN based method for cardiac MR image segmentation which is based on DenseNet connectivity pattern and inception modules.

- The proposed architecture showed that higher performance can be achieved with fewer trainable parameters by properly designing the network connectivity pattern and loss function.
- The customized loss function was observed to perform to better when compared to vanilla cross-entropy loss.
- Replacing ReLUs with ELUs manifested in faster convergence & improved segmentation metrics.

The proposed model was implemented in Tensorflow [17] and Theano [15,16]. The network was trained on NVIDIA K20c GPU. The entire pipeline (ROI extraction, prediction and post-processing) takes about 3 s for one patient's heart volume comprising of 10 SAX-slices at ED and ES phases of heart.

References

1. LeCun, Y., et al.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (2012)
3. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
4. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
5. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
6. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. arXiv preprint arXiv:1505.04597 (2015)
7. Ciresan, D., et al.: Deep neural networks segment neuronal membranes in electron microscopy images. In: Advances in Neural Information Processing Systems (2012)
8. Menze, B.H., et al.: The multimodal brain tumor image segmentation benchmark (BRATS). IEEE Trans. Med. Imaging **34**(10), 1993–2024 (2015)
9. Jgou, S., et al.: The one hundred layers tiramisu: fully convolutional DenseNets for semantic segmentation. arXiv preprint arXiv:1611.09326 (2016)
10. Huang, G., et al.: Densely connected convolutional networks. arXiv preprint arXiv:1608.06993 (2016)
11. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
12. Milletari, F., Navab, N., Ahmadi, S.-A.: V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV). IEEE (2016)
13. https://www.kaggle.com/c/second-annual-data-science-bowl/details/fourier-based-tutorial

14. http://irakorshunova.github.io/2016/03/15/heart.html
15. Theano Development Team. Theano: a Python framework for fast computation of mathematical expressions (2016)
16. Lasagne Development Team. Lasagne: First release (2015)
17. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). http://www.tensorflow.org
18. van der Walt, S., Schnberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T.: scikit-image: image processing in Python. PeerJ **2**, e453 (2014)
19. Liaw, A., Wiener, M.: Classification and regression by random forest. R News **2**(3), 18–22 (2002)
20. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
21. Srivastava, N., et al.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
22. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (2015)
23. Clevert, D.-A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289 (2015)