Real-time temporal coherent left ventricle segmentation using convolutional LSTMs

Erik Smistad
Norwegian University of Science and Technology
Dept. of Circulation and Medical Imaging
SINTEF Medical Technology
Trondheim, Norway
erik.smistad@ntnu.no

Ivar Mjåland Salte
Sørlandet Hospital
Inst. of Clinical Medicine
University of Oslo
Kristiansand, Norway
ivar.mjaland.salte@sshf.no

Håvard Dalen
Norwegian University of Science and Technology
Dept. of Circulation and Medical Imaging
St. Olavs Hospital
Trondheim, Norway
havard.dalen@ntnu.no

Lasse Lovstakken
Norwegian University of Science and Technology
Dept. of Circulation and Medical Imaging
Trondheim, Norway
lasse.lovstakken@ntnu.no

2021 IEEE International Ultrasonics Symposium (IUS) | 978-1-6654-0355-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/IUS52206.2021.9593668

***Abstract*—Most work on left ventricle (LV) ultrasound image segmentation using deep learning has focused on single-frame segmentation of end-diastole (ED) and end-systole (ES) frames. Using these neural network models on the entire cardiac cycle often results in segmentation flickering and sudden large segmentation errors. Neural networks that perform some form of temporal reasoning is needed to solve these issues. In this work, we have investigated the use of neural networks with convolutional long short-term memory (ConvLSTM) layers for real-time temporal coherent LV segmentation. A comparison on a dataset of 174 apical 4-, 3- and 2-chamber ultrasound recordings indicated that increasing the number of frames annotated from the cardiac cycle improves temporal segmentation, while using weighted moving average post processing can reduce segmentation flickering, and using ConvLSTM layers reduces large temporal errors considerably. The runtime of the ConvLSTM segmentation network was 13 ms when used in a real-time application for automatic ejection fraction.**

I. Introduction

Deep neural networks (NNs) are state-of-the-art for left ventricle (LV) segmentation [1]. However, most studies have only trained and evaluated accuracy on end-diastole (ED) and end-systole (ES) images, leaving the rest of the cardiac cycle unstudied. We have studied the accuracy in the entire cardiac cycle of NNs trained only on ED and ES frames over several years and observed several temporal issues such as: segmentation flickering in areas with low signal, incorrect placement of the atrioventricular plane when the mitral valve is open, and other sudden large errors in the segmentation. Automatic echocardiography measurements such as LV volume and ejection fraction are very sensitive in terms of segmentation, even small changes in the contour can impact the volume significantly. Correct ejection fraction measurements are also dependent on having a temporal coherent segmentation over time from ED to ES, meaning that it follows the same physical contour over time. We hypothesize that these temporal issues occur because the NNs only process one image at a time and thus have no memory.

There exist several segmentation methods for temporal image data. The simplest approach is to provide some temporal information as additional input to the NN. This can for instance be the segmentation of the previous frame, and optical flow information such as used in the MaskTrack method [2]. This approach however limits the memory to just the previous frame. NNs with 3D convolutions have been used for temporal LV segmentation [3]. 3D networks are quite heavy in terms of parameters and runtime, and are harder to train. They also have a memory limited to the number of frames sent to the network. Due to these reasons, 3D NNs are not very suitable for real-time use. Long short-term memory (LSTM) can be used to create recurrent NNs. LSTM NNs can also be used in a *stateful* manner, where the NN has a state, which is remembered for each execution of the network. This makes them suitable for real-time use because they can remember over many frames, while still only processing one frame at a time. Still, the standard LSTM layers are fully-connected and 1-dimensional, and thus doesn't fit well into the standard fully-convolutional encoder-decoder segmentation architectures such as U-net. Using them directly in a segmentation NN would lead to a massive increase in parameters, and reduced inference speed and 2D spatial locality. For this reason, the more efficient 2-dimensional Convolutional LSTM (ConvLSTM) [4] layers are better suited for segmentation.

In this work, we have studied how ConvLSTM layers can be used to create real-time temporal coherent LV segmentation.

II. Methods

A. Dataset and annotation

174 apical 4-, 3- and 2-chamber ultrasound recordings from a Norwegian population study dataset (HUNT) were annotated by an expert using Annotation Web [5]. To evaluate and train on the full cardiac cycle instead of just ED and ES, seven frames of a single cardiac cycle were annotated starting with

This research was funded by the Research Council of Norway under project 237887.

The HUNT study (Nord-Trøndelag Health Study) is a collaboration between the HUNT Research Centre (Faculty of Medicine and Health Sciences, Norwegian University of Science and Technology), Nord-Trøndelag County Council, Central Norway Regional Health Authority, and the Norwegian Institute of Public Health. We thank the Nord-Trøndelag Hospital Trust and for support and for contributing to data collection in this research project.

1 ED frame, 2 systole frames, 1 ES frame, 2 diastole frames, and finally 1 ED frame. The endocardium, epicardium and left atrium were annotated in a similar manner to the CAMUS dataset [1]. ED and ES frames were selected as the frame before mitral valve closure as recommended in [6].

B. Neural network architecture

The base NN architecture used in this study was the multi-view fully-convolutional encoder-decoder network described in [7]. This architecture has six levels and uses max pooling in the encoder and 2×2 repeat upsampling in the decoder. Two 3×3 convolution layers are used at each level, together with ReLU activation. The final layer uses softmax activation. Network input is an ultrasound image of size 256×256 together with a binary value indicating if it is an apical 3-chamber/long-axis view or not. The image pixel intensities is normalized to a 0-1 range by dividing by 255. The output is a map of same size as the input image, where each pixel is classified as either LV, myocardium, left atrium or background. This network has about 2 million parameters and was designed for real-time use with a runtime of only a few milliseconds on a modern GPU.

It is not clear where in a NN the ConvLSTMs layers should be used. Thus, we have experimented using ConvLSTMs layers in the encoder, the decoder, last layer and in the bottleneck of the base NN. For each block with two 3×3 convolution layers in the base NN, the first layer was replaced with a ConvLSTM layer with the same filter count and size.

C. Training and loss function

The base NN was trained with a Dice loss function. In order to teach the temporal ConvLSTM NNs to create smooth segmentations over time, we created a new temporal loss function L_{temporal} which simply measures the Dice score D between two consecutive frames (t and $t-1$):

$$L_{\text{temporal}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \text{clip}(D(y_t, y_{t-1}), c, 1.0) - c \quad (1)$$

where clip is a function which clips all values below a minimum value c to c. Since we know there is some movement in the images for each pair of frames due to the beating heart, we perform clipping with a threshold $c = 0.01$ in an attempt to allow the segmentation to change a little without penalty. The final loss was a linear combination of the Dice loss and the temporal loss with weights 0.4 and 0.6 respectively.

For training of the temporal NNs, the annotations of the seven frames of each US sequence were interpolated, to create annotations for every frame in the cardiac cycle. This is a potential source of error, since the interpolation is not able to capture all the complexity of the beating heart.

Random augmentations were used during training to reduce overfitting. The following augmentations were used:

- Gamma intensity transformation.
- Rotation - Maximum angle: 10 degrees.
- Gaussian shadows - Dark shadows applied to the image at random locations and with random sizes.
- Depth - Cuts the image bottom randomly up until the LV.
- JPEG compression - Compresses the image with a random quality setting.
- Blackout - Sets all pixels in a random rectangle to all zeros.

For the temporal NNs an additional batch augmentation was applied which selects at random a subsequence of N frames with random frame step for each ultrasound sequence sample. The temporal NNs were trained in non-stateful manner with N frames for each sample in a batch. Ideally, the size of N should be as high as possible, but with a limited GPU memory, the batch size and N must be adjusted accordingly. The training parameters used for training the temporal NNs on a GPU with 16 GB memory was batch size 2, $N = 20$, and randomly selected frame step of 1 or 2.

D. Comparison metrics

Segmentation accuracy in deep learning is usually reported using the overlap measure **Dice** which is 0 for no overlap and 1 for perfect overlap. Although this metric captures the overall accuracy, it is not necessarily able to capture the temporal issues we seek to resolve. Since not all frames are annotated, if the temporal error occurs on a frame which is not annotated, it will not be measured. To solve this, we have counted **large temporal errors (#LTE)** as the number of recordings where the Hausdorff distance between a frame and the closest annotated frame was above a high threshold of 30 mm. Although not a perfect measure, it seems to be able to capture large temporal errors occurring on non-annotated frames. Segmentation flickering was measured with the **mean flickering image pixels (mFIP)** measure introduced in [8] which simply measures how many pixels changed their label from one frame to the other:

$$\mathbf{mFIP} = \frac{1}{WH} \sum_{t=1}^{T-1} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} |\text{sign}(S_t(x,y) - S_{t-1}(x,y))| \quad (2)$$

where S is the segmentation result image at timestep t with size $W \times H$. In addition, we also measured the **Hausdorff distance** in millimeters which is the maximal distance between the closest points on the segmentation and ground truth contours. This metric can give an impression of size of the largest segmentation errors.

E. Comparison study

To evaluate the effect of the proposed multi-view ConvLSTM segmentation NN, the performance of several alternative approaches were measured. Using the base NN from [7] trained on 1) ED/ES frames only, 2) all 7 frames, and 3) all 7 frames with temporal smoothing post processing. The temporal smoothing technique used was weighted moving average (WMA). A window size of 6 frames was used for WMA. In summary, the following four approaches were trained and tested on the same dataset using 10-fold cross validation:

- **Non-temporal NN** trained on **ED/ES only**.
- **Non-temporal NN** trained on all **7 frames**.

TABLE I
CROSS VALIDATION RESULTS OF MEAN FLICKERING IMAGE PIXELS (MFIP), NUMBER OF LARGE TEMPORAL ERRORS (#LTE), MEAN DICE SCORE AND HAUSDORFF DISTANCE IN MILLIMETERS FOR EACH STRUCTURE.

Experiment	mFIP	#LTE	Dice LV	Dice Myoc.	Dice LA	Hausd. LV	Hausd. Myoc.	Hausd. LA
Non-temporal ED/ES frames	0.013	11	0.93 ± 0.04	$\mathbf{0.80} \pm 0.08$	0.89 ± 0.09	5.78 ± 4.08	6.38 ± 4.25	5.07 ± 4.44
Non-temporal 7 frames	0.011	4	$\mathbf{0.94} \pm 0.03$	$\mathbf{0.80} \pm 0.08$	$\mathbf{0.91} \pm 0.05$	$\mathbf{4.94} \pm 2.13$	$\mathbf{5.60} \pm 2.71$	$\mathbf{4.31} \pm 2.46$
Non-temporal 7 frames + WMA	**0.007**	3	0.93 ± 0.03	$\mathbf{0.80} \pm 0.08$	$\mathbf{0.91} \pm 0.05$	4.95 ± 2.06	5.71 ± 3.24	4.39 ± 2.33
Temporal ConvLSTM	**0.007**	**1**	0.93 ± 0.03	0.79 ± 0.07	0.88 ± 0.07	5.55 ± 2.09	6.26 ± 2.65	5.66 ± 2.91

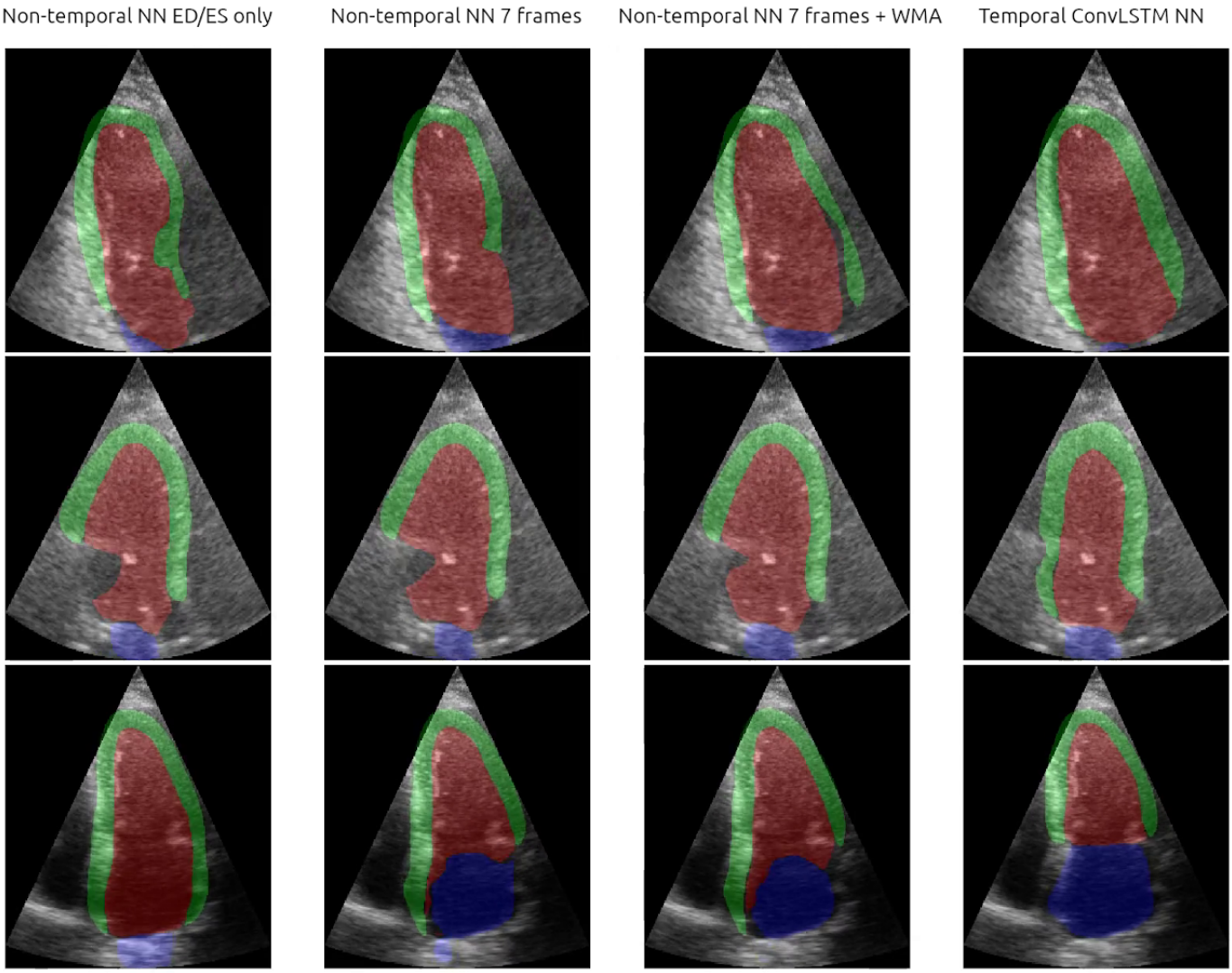

Fig. 1. Three examples of large temporal errors (**LTE**) which were only resolved using the proposed temporal ConvLSTM NN method.

- **Non-temporal NN** trained on all **7 frames** and applying **WMA** post processing.
- **Temporal ConvLSTM NN** trained on interpolated annotations.

III. RESULTS

Table I shows the results of the four tested methods. Note that Dice and Hausdorff were only calculated on the 7 frames annotated by an expert, while mFIP and #LTE were calculated using the entire cardiac cycle. Using ConvLSTM layers only in the encoder of the temporal NN gave the best results, while at the same keeping the number of parameters at a reasonable level ($\sim$ 2.7 million). Comparing non-temporal NNs trained with only ED and ES frames, and with all 7 frames, the results showed that mFIP and #LTE were reduced by including all 7 frames (0.013$\rightarrow$0.011, 11$\rightarrow$4). The temporal errors were reduced further by applying WMA (0.007, 3). The temporal ConvLSTM NN achieved the best temporal results (0.007, 1). Dice accuracy was however slightly reduced with the temporal

NN (0.94→0.93 and 0.80→0.79 for LV and myocardium). Fig. 1 shows three examples of large temporal errors which were only solved using the temporal ConvLSTM NN.

By converting the ConvLSTM NN to a stateful version, we were able to use it in real-time by feeding the network one frame at a time using TensorFlow. The NN inference runtime was 13 ms on an NVIDIA RTX 2080 GPU when used in a real-time application for automatic ejection fraction [9].

IV. Discussion

In this work, we have shown that convolutional LSTMs layers can be used in a fully convolutional segmentation neural network to efficiently produce more temporally coherent LV segmentations of the entire cardiac cycle. While temporal flickering can be resolved using simple post processing techniques such as WMA, the temporal ConvLSTM network was able to eliminate more large temporal errors than using WMA. Still, we observe that this comes at the cost of over-smoothing and more stiff temporal segmentation, which is reflected in the slightly lower Dice and Hausdorff scores in Table I. Also, the temporal ConvLSTM is not able to eliminate all large temporal errors, thus there is room for improvement.

One drawback of this work is the use of interpolation to annotate the entire cardiac cycle and use this to train the temporal network. The interpolation is not able to capture the complex motion of the LV from just seven frames. Training without interpolated annotations was tested, but gave worse results. In this approach, only the seven annotated frames were used, and the Dice loss was not calculated for the frames lacking annotations, while the temporal loss was used for all frames. An alternative could be to use advanced speckle tracking methods such as Echo-PWC-Net [10], to create more accurate temporal annotations of the entire cardiac cycle.

V. Conclusion

Temporal coherent segmentation of the left ventricle from apical 2-, 3- and 4-chamber recordings using neural networks was investigated. The results indicate that increasing the number of frames annotated from the cardiac cycle helps, while using weighted moving average post processing can reduce segmentation flickering. Still, large temporal errors were best reduced using a neural network with convolutional LSTM layers which has the ability to remember. This network was found to be very efficient enabling real-time usage.

References

[1] S. Leclerc, E. Smistad, J. Pedrosa, A. Ostvik, F. Cervenansky, F. Espinosa, T. Espeland, E. A. R. Berg, P.-M. Jodoin, T. Grenier, C. Lartizien, J. Drhooge, L. Lovstakken, and O. Bernard, "Deep Learning for Segmentation using an Open Large-Scale Dataset in 2D Echocardiography," *IEEE Transactions on Medical Imaging*, pp. 1–1, 2019.

[2] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning Video Object Segmentation from Static Images," in *CVPR*, 12 2016. [Online]. Available: http://arxiv.org/abs/1612.02646

[3] H. Wei, H. Cao, Y. Cao, Y. Zhou, W. Xue, D. Ni, and S. Li, "Temporal-Consistent Segmentation of Echocardiography with Co-learning," in *MICCAI*. Springer International Publishing, 2020, pp. 623–632.

[4] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," pp. 1–9, 2015. [Online]. Available: http://arxiv.org/abs/1506.04214

[5] E. Smistad, A. Østvik, and L. Lovstakken, "Annotation Web - An open-source web-based annotation tool for ultrasound images," in *IEEE International Ultrasonics Symposium, IUS*, 2021.

[6] R. M. Lang, L. P. Badano, V. Mor-Avi, J. Afilalo, A. Armstrong, L. Ernande, F. A. Flachskampf, E. Foster, S. A. Goldstein, T. Kuznetsova, P. Lancellotti, D. Muraru, M. H. Picard, E. R. Rietzschel, L. Rudski, K. T. Spencer, W. Tsang, and J.-U. Voigt, "Recommendations for Cardiac Chamber Quantification by Echocardiography in Adults: An Update from the American Society of Echocardiography and the European Association of Cardiovascular Imaging," *Journal of the American Society of Echocardiography*, vol. 28, no. 1, pp. 1–39, 1 2015.

[7] E. Smistad, I. Salte, A. Ostvik, S. Leclerc, O. Bernard, and L. Lovstakken, "Segmentation of apical long axis, four- and two-chamber views using deep neural networks," in *IEEE International Ultrasonics Symposium, IUS*, vol. 2019-Octob, 2019.

[8] A. Pfeuffer and K. Dietmayer, "Separable Convolutional LSTMs for Faster Video Segmentation," *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, pp. 1072–1078, 2019.

[9] E. Smistad, A. Ostvik, I. M. Salte, D. Melichova, T. M. Nguyen, K. Haugaa, H. Brunvand, T. Edvardsen, S. Leclerc, O. Bernard, B. Grenne, and L. Lovstakken, "Real-time automatic ejection fraction and foreshortening detection using deep learning," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, pp. 1–1, 2020.

[10] A. Ostvik, I. M. Salte, E. Smistad, T. M. Nguyen, D. Melichova, H. Brunvand, K. Haugaa, T. Edvardsen, B. Grenne, and L. Lovstakken, "Myocardial Function Imaging in Echocardiography Using Deep Learning," *IEEE Transactions on Medical Imaging*, vol. 40, no. 5, pp. 1340–1351, 5 2021.