TWO-STREAM ATTENTION SPATIO-TEMPORAL NETWORK FOR CLASSIFICATION OF ECHOCARDIOGRAPHY VIDEOS

Zishun Feng[1], *Joseph A. Sivak*[2] *and Ashok K. Krishnamurthy*[1,3]

[1]Department of Computer Science, [2]Division of Cardiology, [3]Renaissance Computing Institute,
University of North Carolina, Chapel Hill

ABSTRACT

There is considerable interest in AI systems that can assist a cardiologist to diagnose echocardiograms, and can also be used to train residents in classifying echocardiograms. Prior work has focused on the analysis of a single frame. Classifying echocardiograms at the video-level is challenging due to intra-frame and inter-frame noise. We propose a two-stream deep network which learns from the spatial context and optical flow for the classification of echocardiography videos. Each stream contains two parts: a Convolutional Neural Network (CNN) for spatial features and a bi-directional Long Short-Term Memory (LSTM) network with Attention for temporal. The features from these two streams are fused for classification. We verify our experimental results on a dataset of 170 (80 normal and 90 abnormal) videos that have been manually labeled by trained cardiologists. Our method provides an overall accuracy of 91.18%, with a sensitivity of 94.11% and a specificity of 88.24%.

***Index Terms*—** Echocardiography, Classification, Deep learning

1. INTRODUCTION

Echocardiography presents unique challenges to machine learning algorithms compared to other medical imaging applications. Unlike CT and MRI, where each patient is carefully positioned within the scanner and static images are generated, echocardiography generates video loops which are subject to variations in technique (e.g., patient position, probe position and angulation, patient respiration), and machine settings (e.g., probe selection, depth, gain, and compression settings). Not only does the machine learning method need to recognize the anatomical findings, but it needs to learn the image variables caused by technical variations vs. true pathology.

Due to the versatility and cost-effectiveness of echocardiography, it is typically the first-line imaging study for most cardiac diagnoses. As such, the ability to instantaneously and automatically detect echocardiogram findings could have broadly impactful clinical applications. In particular, rapidly identifying a normal echocardiogram would rule out several clinical findings, enabling providers to focus on other causes of the patient's presentation.

Deep neural networks (DNNs) have recently achieved state-of-the-art results in computer vision[1, 2] and medical image analysis[3]. Deep learning-based approaches also play an increasing role in automatic echocardiography analysis. There have been numbers of prior works on image-based echocardiogram analysis: 1) viewpoint classification[4, 5], 2) identification of certain diseases[5, 6] (for example, left ventricular hypertrophy and hypertrophic cardiomyopathy), and 3) chamber identification and segmentation[5, 7, 8]. Most of these applications take a single frame as input and only consider spatial information. Videos contain not only spatial information but also temporal (motion) information, which will better guide echo analysis. Moreover, video-based classification problems, like abnormality detection and disease identification, only require video-level labels, which are easier to get than frame-level labels in image-based methods. Lee *et al.*[9] proposed a video-based method to determine the fetal cardiac cycle in ultrasound automatically. However, while their method only uses video frames as input, it does not use any explicit motion information.

In this paper, we propose a two-stream attention spatial-temporal network for classification of parasternal long view echocardiography videos. First, our two-stream network takes video frames and optical flow as input to exploit spatial context and motion information. Second, in each stream, a pre-trained auto-encoder, which will be fine-tuned with LSTM and classifier, extracts visual features fed into the temporal network. Third, we employ the Attention module to let the LSTM focus on task-related time steps.

2. ECHOCARDIOGRAPHY DATASET

Consecutive patients aged 18-65 without prior cardiac disease undergoing routine outpatient echocardiography at UNC Hospital and clinics were included. 121 echocardiography videos were included, and the parasternal long 2D loop for each study was anonymized and independently evaluated by 3 expert (COCATS Level III) cardiologists. Of the 121 echocardiography videos evaluated, 54 videos had a consensus read of normal. 35 echocardiography videos were split

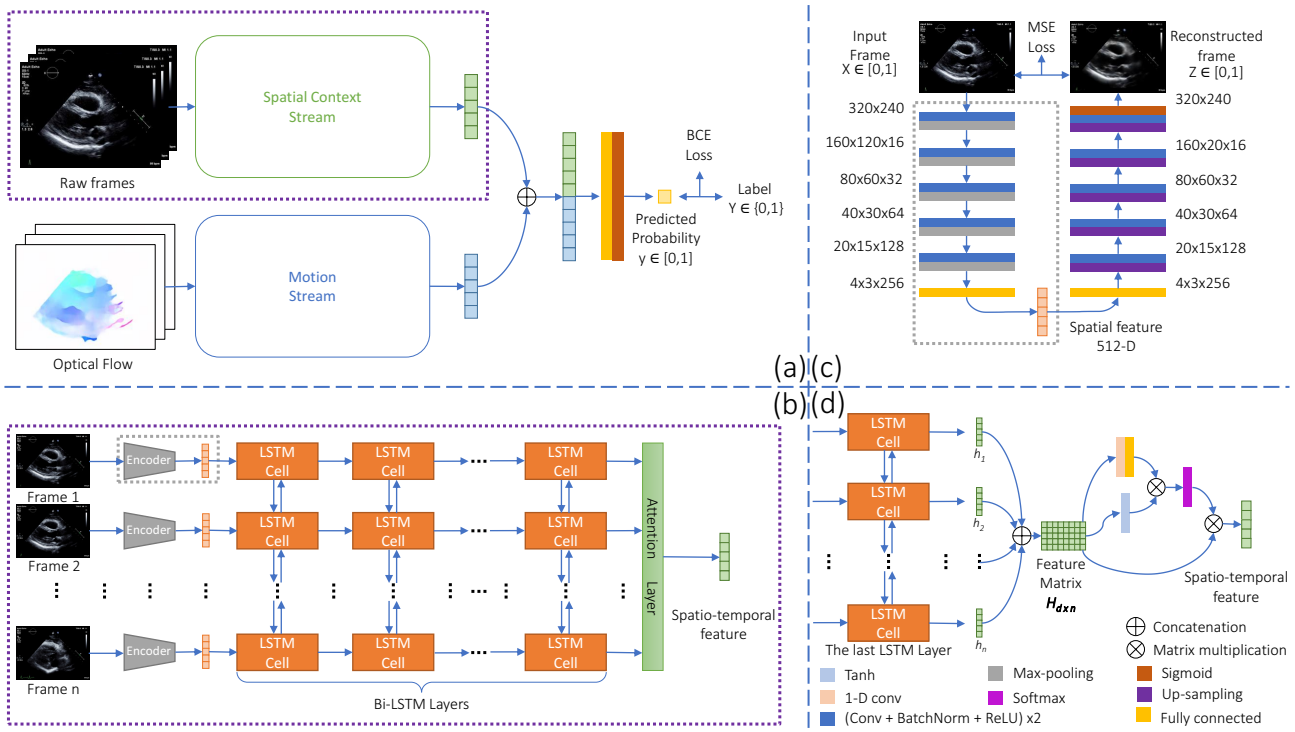

Fig. 1: Architecture of proposed network. (a): Architecture of overall two-stream network. (b): Architecture of each stream, taking the spatial context stream for example. (c): Architecture of CNN auto-encoder. (d): Architecture of Attention layer. The details of the purple dotted box in part (a) are shown in part (b), and the details of the gray dotted box in part (b) are shown in part (c).

into two or more 30-frame segments without any overlap, only 30 frames from each video were kept for processing. In total, our dataset consists of 170 (80 normal and 90 abnormal) 30-frame echocardiography video segments from 121 (54 normal and 67 abnormal) subjects.

The original resolution of each frame is 640×480. In this study, we followed [6] and resized all videos to 320×240 per frame and used 30 frames (1 second).

3. METHODS

3.1. Network Architecture

Our proposed network architecture is shown in Figure 1. The network consists of two streams: a spatial context stream that takes echo frames as input and a motion stream that takes optical flow as input. These two streams use the same architecture that consists of two sub-networks: a convolutional encoder and an Attention LSTM. The learned features from these two streams are concatenated and fed into the top layer for class prediction.

3.1.1. Optical Flow

Optical flow plays an important role in video analysis. Optical flow is computed from successive frames and provides the pattern of apparent motion of objects in the visual scene caused by the relative motion between the observer and a scene. Dense optical flow can be considered as a displacement vector field, which moves the point in the video frame at time t to the corresponding point in frame $t + 1$. Optical flow is usually computed by conventional algorithms, such as the Lucas–Kanade (L-K) method[10], which requires several hyperparameters to be tuned.

Here, we use a pre-trained FlowNet2 model[11], a CNN-based method, for optical flow computation. A pre-trained FlowNet2 model is hyperparameter-free and only takes two frames as input to compute the optical flow. FlowNet2, compared with the conventional algorithm, gives a smoother displacement field when there is noise in the input frames.

3.1.2. Convolutional Encoder

We pre-train an auto-encoder on each of the two streams independently and use the encoding part of the auto-encoder to provide the spatial features. The convolutional auto-encoder is shown in Figure 1(c). The encoder with 5 down-sampling steps and a fully convolutional layer maps the input video

frame (or optical flow) to a 512-D vector. In each down-sampling step, two convolution layers and one max-pooling layer are used. The number of convolution filters is set to 16 at the first stage and doubled after each max-pooling. Furthermore, the decoder is the inverse of the same architecture and reconstructs the input video frame (or optical flow) from the embedded 512-D vector.

During the pre-training, the auto-encoder is optimized with mean square error (MSE) loss:

$$L_{MSE}(\theta_{AE}) = -\sum_{i} (X_i - Z_i)^2 \quad (1)$$

where θ_{AE} denotes the parameters of auto-encoder, i denotes the index of pixels, X and Z are input and reconstructed echo frame (or optical flow). Only the encoder is used in the following training and inference stages.

3.1.3. Bi-directional LSTM with Attention

To temporally aggregate features learned from the convolutional encoder, we employ a 4-layer bi-directional LSTM network. The first LSTM layer takes features from the CNN encoder as input, returns the hidden state at each time step, and feeds these hidden states into the next LSTM layer. The hidden size of each layer is set to 256, and the dropout rate is 0.3. One problem of the LSTM network is that the performance deteriorates as the length of the input sequence increases. The Attention mechanism is adopted to address this challenge and make the network 'focus' on task-related time steps.

The Attention mechanism, introduced by Bahdanau *et al* [12] to solve machine translation problems, is designed to retrieve information from a set of features $\{h_j\}$ related to a query vector c. Let H be the matrix that consists of output features $[h_1, h_2, ..., h_n]$, where n is the length of input videos. The computation of the Attention layer is as follows:

$$U = \tanh(H) \quad (2)$$

$$e = c^T U \quad (3)$$

$$\alpha_j = \frac{exp(e_j)}{\sum_j exp(e_j)} \quad (4)$$

$$Attention(c, \{h_j\}) = \sum_j \alpha_j h_j \quad (5)$$

where e_j, the j^{th} element in vector e, is the matching score of c and h_j, α_j is the softmax score of e_j. The dimensions of $\{H, c, e, \alpha\}$ are $\{d{\times}n, d{\times}1, 1{\times}n, 1{\times}n\}$ respectively, where d is the hidden size. The query vector c is obtained from H after a 1-D convolutional layer and a fully connected layer. The output of the Attention layer is the weighted average of $\{h_j\}$, described by Equation 5.

3.1.4. Feature Fusion and Loss Function

In the final stage of the network, the features from the two streams (spatial context and motion) are concatenated and used to predict the final class using a fully connected layer with sigmoid activation. The overall network is trained using a binary cross-entropy (BCE) loss as the task is a binary classification problem:

$$L_{BCE}(\theta) = -\sum_{k} Y_k \log y_k - (1 - Y_k) \log(1 - y_k) \quad (6)$$

where θ denotes all parameters in the network, k denotes the index of videos, Y_k is the training label for the video, and y_k is the predicted probability.

3.2. Network training

Our proposed network is trained in three stages: 1) pre-training of the CNN auto-encoder; 2) pre-training the entire network but with fixed CNN weights; and 3) fine-tuning of the entire network. In the first stage, the auto-encoders in both streams are pre-trained for 30 epochs. In the second stage, we fix the convolutional encoders' weights and train the rest of the network for 30 epochs. In the third stage, we train all parameters in the network for 30 epochs. The learning rates of these three stages are set to $1e^{-4}$, $1e^{-4}$, $1e^{-5}$ respectively.

Our proposed network was implemented in Pytorch and optimized using the Adam optimizer in all three stages with parameters: beta1=0.9, beta2=0.999, and the learning rates mentioned above. We trained and tested our network on a Tesla V100 GPU. And the number of total parameters is 23M.

4. TESTING AND EXPERIMENTAL RESULTS

We evaluated our proposed network architecture from three aspect: 1) two-stream; 2) Attention mechanism; and 3) training strategy. The training and testing set splits are shown in Table 1. Note that the entire network including the CNN auto-encoder were trained on the training set only. We augmented our data by randomly rotating the input by angles in $[-30°, 30°]$ during training. We also used 4-fold cross-validation on the training set to select the best parameter for testing. In the experiments, accuracy, sensitivity ,and specificity are adopted to evaluate our classification performance.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (7)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (8)$$

$$Specificity = \frac{TN}{TN + FP} \quad (9)$$

where {TP, FP, TN, FN} denote the numbers of {true positive, false positive, true negative, false negative} cases respectively.

Table 1: Training and testing splits of the dataset.

	Training		Testing	
	#subjects	#videos	#subjects	#videos
Abnormal (P)	50	73	17	17
Normal (N)	37	63	17	17

4.1. Impact of two-stream design

As mentioned in Section 3.1, our proposed network takes both spatial context and optical flow as input. To investigate the impact of the two-stream design, we trained the network with different settings: 1) spatial context stream only; 2) motion stream only; and 3) using two-stream. Moreover, we also compared different optical flow computation methods: FlowNet2 and the L-K method. Note that we trained all these models following the training details in Section 3.2. The performance of above four different settings is shown in Table 2.

Table 2: Comparison of one- and two-stream networks.The bold font denotes the best performance. Accuracy (Acc), Sensitivity (Sen) and Specificity (Spe).

Method	Acc(%)	Sen(%)	Spe(%)
Spatial context stream only	82.35	88.24	76.47
Motion stream only (FlowNet2)	73.53	82.35	64.71
Two-stream (L-K method)	85.29	88.24	82.35
Two-stream (FlowNet2)	**91.18**	**94.11**	**88.24**

4.2. Impact of the Attention mechanism

As mentioned in Section 3.1.3, the Attention module is added to the original LSTM to ensure focus on task-related time steps. To evaluate its impact, we compared the networks with and without the Attention module. In the latter case, The Attention layer was replaced by the unweighted average of the LSTM outputs. The results are shown in Table 3.

Figure 2 shows the Attention weights (α in Eq. 4) aligned to the electrocardiogram (ECG) corresponding to the echo video. The figure shows that, the Attention layer focuses on different phases of a cardiac cycle when predicting normal vs. abnormal echo videos.

Table 3: Comparison of networks with and without the Attention module. The bold font denotes the best performance.

Method	Acc(%)	Sen(%)	Spe(%)
Without Attention module	88.24	88.24	**88.24**
With Attention module	**91.18**	**94.11**	**88.24**

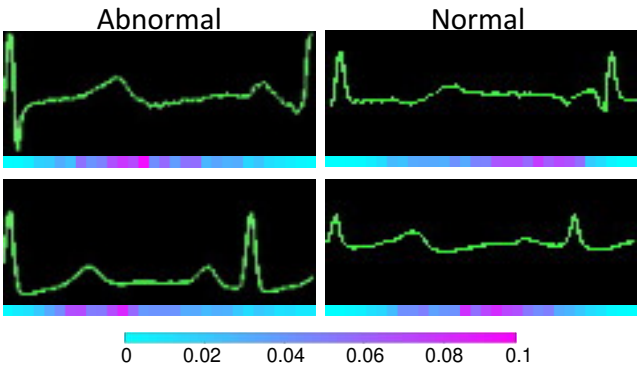

Fig. 2: Visualization of Attention weights.

4.3. Impact of network training

In Section 3.2, we have introduced the training strategy of our network. To explore the impact of network training, we trained three networks with different strategies. The first network was trained entire network *from scratch* (i.e., without pre-training and fine-tuning) with learning rates of $1e^{-4}$ for 60 epochs followed by a learning rate of $1e^{-5}$ for 30 epochs. The second network pre-trained the convolutional encoder in the first stage with learning rate $1e^{-4}$ for 30 epochs and only pre-trained and fine-tuned the entire network but with fixed CNN weights in the following *two stages* with learning rates of $\{1e^{-4}, 1e^{-5}\}$ for $\{30, 30\}$ epochs. The third network was trained following the strategy in Section 3.2. The results comparing these training strategies are shown in Table 4.

Table 4: Comparison of network training strategies.

Method	Acc(%)	Sen(%)	Spe(%)
No pre-training	50	52.94	47.06
Only CNN pre-trained	85.29	88.24	82.35
Proposed training	**91.18**	**94.11**	**88.24**

5. CONCLUSIONS

In this paper, we have described a two-stream attention spatio-temporal network to recognize normal/abnormal echocardiography videos. The two-stream network utilizes spatial context and motion information by taking raw frames and optical flow as input. In each stream, spatio-temporal features are extracted and aggregated by the CNN-LSTM network. Also, the Attention module helps the network give a better prediction. The results show that the proposed method is able to achieve good performance on the datasets used. We are in the process of increasing the number of datasets and understanding the clinical significance of the mis-classifications.

6. ACKNOWLEDGMENTS

No funding was received for conducting this study. The authors have no relevant financial or non-financial interests to disclose.

7. COMPLIANCE WITH ETHICAL STANDARDS

This research was conducted under IRB 18-2345 approved by the University of North Carolina, Chapel Hill IRB Board, Joseph Sivak, PI.

8. REFERENCES

[1] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations*, 2015.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[4] Ali Madani, Ramy Arnaout, Mohammad Mofrad, and Rima Arnaout, "Fast and accurate view classification of echocardiograms using deep learning," *NPJ digital medicine*, vol. 1, no. 1, pp. 1–8, 2018.

[5] Jeffrey Zhang, Sravani Gajjala, Pulkit Agrawal, Geoffrey H Tison, Laura A Hallock, Lauren Beussink-Nelson, Mats H Lassen, Eugene Fan, Mandar A Aras, ChaRandle Jordan, et al., "Fully automated echocardiogram interpretation in clinical practice: feasibility and diagnostic accuracy," *Circulation*, vol. 138, no. 16, pp. 1623–1635, 2018.

[6] Ali Madani, Jia Rui Ong, Anshul Tibrewal, and Mohammad RK Mofrad, "Deep echocardiography: data-efficient supervised and semi-supervised deep learning towards automated diagnosis of cardiac disease," *NPJ digital medicine*, vol. 1, no. 1, pp. 1–11, 2018.

[7] Mohammad H Jafari, Hany Girgis, Amir H Abdi, Zhibin Liao, Mehran Pesteie, Robert Rohling, Ken Gin, Terasa Tsang, and Purang Abolmaesumi, "Semi-supervised learning for cardiac left ventricle segmentation using conditional deep generative models as prior," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 2019, pp. 649–652.

[8] Tingyang Yang, Jiancheng Han, Haogang Zhu, Tiantian Li, Xiaowei Liu, Xiaoyan Gu, Xiangyu Liu, Shan An, Yingying Zhang, Ye Zhang, et al., "Segmentation of five components in four chamber view of fetal echocardiography," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2020, pp. 1962–1965.

[9] Lok Hin Lee and J Alison Noble, "Automatic determination of the fetal cardiac cycle in ultrasound using spatio-temporal neural networks," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2020, pp. 1937–1940.

[10] Bruce D Lucas, Takeo Kanade, et al., "An iterative image registration technique with an application to stereo vision," 1981.

[11] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[12] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *International Conference on Learning Representations*, 2015.