HHS Public Access

Author manuscript

Med Image Anal. Author manuscript; available in PMC 2022 August 01.

Published in final edited form as:

Med Image Anal. 2022 August ; 80: 102438. doi:10.1016/j.media.2022.102438.

Real-time echocardiography image analysis and quantification of cardiac indices

Ghada Zamzmi[a,*], **Sivaramakrishnan Rajaraman**[a], **Li-Yueh Hsu**[b], **Vandana Sachdev**[c], **Sameer Antani**[a]

[a]Computational Health Research Branch, National Library of Medicine, National Institutes of Health, Bethesda, MD, USA

[b]Radiology and Imaging Sciences, Clinical Center, National Institutes of Health, Bethesda, MD, USA

[c]Echocardiography Laboratory, National Heart, Lung, and Blood Institute, National Institutes of Health, Bethesda, MD, USA

Abstract

Deep learning has a huge potential to transform echocardiography in clinical practice and point of care ultrasound testing by providing real-time analysis of cardiac structure and function. Automated echocardiography analysis is benefited through use of machine learning for tasks such as image quality assessment, view classification, cardiac region segmentation, and quantification of diagnostic indices. By taking advantage of high-performing deep neural networks, we propose a novel and eicient real-time system for echocardiography analysis and quantification. Our system uses a self-supervised modality-specific representation trained using a publicly available large-scale dataset. The trained representation is used to enhance the learning of target echo tasks with relatively small datasets. We also present a novel Trilateral Attention Network (TaNet) for real-time cardiac region segmentation. The proposed network uses a module for region localization and three lightweight pathways for encoding rich low-level, textural, and high-level features. Feature embeddings from these individual pathways are then aggregated for cardiac region segmentation. This network is fine-tuned using a joint loss function and training strategy. We extensively evaluate the proposed system and its components, which are echo view retrieval, cardiac segmentation, and quantification, using four echocardiography datasets. Our experimental results show a consistent improvement in the performance of echocardiography analysis tasks with enhanced computational eiciency that charts a path toward its adoption in clinical practice. Specifically, our results show superior real-time performance in retrieving good quality echo from individual cardiac view, segmenting cardiac chambers with complex overlaps, and extracting cardiac indices that highly agree with the experts' values. The source code of our implementation can be found in the project ' s GitHub page.

[*]Corresponding author. alzamzmiga@nih.gov (G. Zamzmi). .

Declaration of Competing Interest

The authors declare no conflict of interests.

Keywords

Echocardiography; Classification; Segmentation; Real-time analysis; Cardiac quantification; B-mode; Doppler; Quality assessment

1. Introduction

Echocardiography, henceforth echo, is a safe and low-cost imaging test that is widely used for the analysis of cardiac structure and function. It is clinically used for disease diagnosis based on the interpretation of clinical biomarkers such as the ejection fraction and chamber volume. In clinical routine, the process of extracting these biomarkers involves visually selecting good-quality end-diastole (ED) and end-systole (ES) frames from a specific cardiac view followed by manually, or semi-automatically, delineating specific cardiac regions for quantification (Oh et al., 2006). This process is tedious, error-prone, often prohibitively expensive for analyzing large studies, and is limited by inter- and intra-observer variability (Oh et al., 2006). To mitigate these issues, quantitative imaging has attracted increasing attention in recent years (Zamzmi et al., 2020) as it can provide low-cost and scalable analysis, facilitate the monitoring of disease characteristics and functions, expedite clinical workflow, standardize echo analysis, and help democratize it in rural settings with limited resources. Further, automated echo analysis can lead to the detection of new patterns aiding in gaining new understandings and discoveries (Wong et al., 2020).

Recent advances in machine learning (ML) techniques have allowed the development of automated methods for echo image quality assessment, view classification, cardiac region segmentation, and disease diagnosis (Zamzmi et al., 2020; Wong et al., 2020). However, the performance of these methods is often limited by the size of the labeled training datasets and the required computational resources (Zamzmi et al., 2020). As many echo datasets are limited in size, these methods often adopt transfer learning from large models pre-trained on a large-scale collection of natural images (e.g., ImageNet). However, several studies (Torrey and Shavlik, 2010; Rosenstein et al., 2005; Jiang et al., 2020) reported the negative impacts of transferring the knowledge from the natural image domain to the medical image domain due to the differences between these domains in terms of visual characteristics (e.g., shape, color, texture), spatial resolution, data and noise distributions. In addition to the impact of negative transfer, current automated methods are designed with clinical performance in mind and little consideration is given to issues such as speed, computational time, model size, and power/energy consumption.

In this work, we present a fully automated system for real-time echo retrieval, region-based cardiac segmentation, and quantification of cardiac indices. Our system uses a self-supervised echo-specific representation to enhance the learning of the target echo tasks (i.e., image retrieval and segmentation prior to quantification). To obtain real-time performance, we use a lightweight multi-head network for image retrieval, and present a novel network, named Trilateral Attention Network (TaNet), for fast region-based segmentation. We demonstrate the eiciency of the proposed framework using four echo imaging datasets and evaluate its performance against expert-based classification,

segmentation, and quantification. Our experimental results show a consistent improvement in the performance of the echo tasks with enhanced computational eiciency that charts a path toward adoption in clinical practice.

1.1. Related work

We briefly present the state-of-the-art methods for echo classification and segmentation. A comprehensive review of existing methods can be found in Zamzmi et al. (2020).

1.1.1. Automated echo view classification—Echo view classification categorizes the acquired echo into different views such as the parasternal long axis view (PLAX), apical four-chamber view (A4C), and subcostal view (SV). Broadly, existing methods for view classification can be divided into conventional ML-based methods and deep learning-based methods.

Wu et al. (2013) presented one of the first conventional methods for classifying echo images into eight views including PLAX, A4C, and SV. Their method extracts spectral energy features from the images using a GIST descriptor. These features are then used to train a support vector machine (SVM) for classification. Other methods used SVM with handcrafted descriptors such as Scale-invariant feature transform (SIFT) (Qian et al., 2012), histogram of oriented gradients (HOG) Agarwal et al. (2013), and bag of visual words (BoWs) (Penatti et al., 2015).

Recent works for echo view classification use state-of-the-arts deep convolutional neural networks (CNNs) such as VGG (Simonyan and Zisserman, 2014), DenseNet (Iandola et al., 2014), and ResNet (He et al., 2016). For example, Zhang et al. (2017) used a VGG-based model to classify echo images into 23 views including PLAX, A4C, and SV. Similarly, Madani et al. (2018) used a VGG-based method for view classification. Their method classifies echo images into 3 modes: B-mode (12 views), M-mode, and Doppler (2 views). The networks in both studies (Zhang et al., 2017; Madani et al., 2018) are trained using random weights or ImageNet weights. Instead of using the deep VGG architecture, Vaseli et al. (2019) used shallow versions of VGG to classify echo images into 12 views. The shallow models have corresponding large teacher models, which are used to transfer the learned knowledge to the three lightweight student models. The lightweight models have only 1% of the three teacher models parameters, and hence, they are significantly faster. Other deep learning-based methods for view classification can be found in Østvik et al. (2019) ; Smistad et al. (2020).

1.1.2. Automated echo quality assessment—Echo quality assessment task involves detecting and eliminating the low-quality echoes. Automating this task facilitates the analysis of subsequent tasks because it automatically removes unusable or unmeasurable cases. Previously reported works in echo quality assessment use traditional techniques to generate a quality score. For example, Snare et al. (2012) developed an algorithm based on a parametric multi-chamber model of A4C view and an extended Kalman filter to calculate a goodness-of-fit score for determining acceptability of the recorded echo. Pavani et al. (2012) used a Generalized Hough Transform (GHT) for delineating incoming echo images in PLAX view and comparing them with a representative atlas to generate a quality score.

Both studies achieved acceptable performance in assessing the quality of A4C and PLAX views.

Several deep learning-based methods have also been proposed for quality assessment. For example, Abdi et al. (2017) proposed a multi-stream network architecture that consists of five regression models for five cardiac views. The proposed method achieved 85% ± 12% accuracy and outperformed previous traditional techniques. Dong et al. (2019) proposed a deep learning framework that consists of three networks; a basic network that roughly detects A4C view, a deeper CNN that performs classification refinement, and finally an aggregated residual visual block network that automates key anatomical structures detection. The proposed network achieved 93.5% mean average precision (mAP). Other deep learning-based approaches for echo quality assessment can be found in Nolan and Thavendiranathan (2019); Liao et al. (2019); Zamzmi et al. (2019b); Vrettos et al. (2020).

Previous deep learning-based methods for echo view classification and quality assessment, except Dong et al. (2019) ; Vaseli et al. (2019), focused mainly on the performance while ignoring the speed and computational complexity of the models. Further, these methods were initialized with random weights or those learned from natural images. In this work, we propose a customized lightweight multi-head model with echo-specific representation for real-time echo retrieval, which consists of view classification (head 1) and quality assessment (head 2). We hypothesize that the lightweight model with the echo modality-specific weights would result in a generalized positive knowledge transfer and faster convergence leading to improved accuracy and real-time performance.

1.1.3. Automated echo segmentation—Segmentation is performed to delineate the boundary of a desired cardiac region for quantifying biomarkers. These biomarkers can be subsequently used for cardiac disease monitoring and diagnosis.

Previously reported methods for echo segmentation used low-level image processing-based methods such as watershed (Cheng et al., 2005; Lacerda et al., 2008) and Otsu thresholding (Santos et al., 2007), deformable model-based methods such as active contour (Chen et al., 2007), B-spline snake (Marsousi et al., 2010; Oktay and Akgul, 2009) and level set (Nandagopalan et al., 2010), and statistical model-based methods such as active shape (Guo et al., 2014; Beymer et al., 2009) and active appearance models (Belous et al., 2013).

Instead of using the traditional methods, deep learning-based methods, such as Fully Convolutional Networks (FCN) and UNET, have also been widely used to achieve state-of-the-art performance in segmenting various cardiac chambers. FCN is one of the first and most widely used deep learning architectures for semantic segmentation in several medical imaging domains including echo (Chen et al., 2016; Dong et al., 2018; Yang et al., 2019). Although FCN outperformed the traditional methods, it reduces the resolution of the input image resulting in sub-optimal prediction with fuzzy object boundaries. To address this issue, advanced deep learning architectures have been proposed. These architectures can be subdivided into encoder-decoder architecture and dilation architecture.

In the encoder-decoder architecture, the encoder is used to extract spatial features at different levels of abstraction while the decoder is used to upsample the output of the encoder to the original image resolution. Typically, this architecture uses skip connections to recover high-resolution details of the predicted output image by concatenating the feature maps from the encoder with their corresponding upsampled feature maps in the decoder. Examples of encoder-decoder segmentation architectures include UNET (Ronneberger et al., 2015) and SegNet (Badrinarayanan et al., 2017). These models have been used for left ventricle (LV) segmentation in echo images. For example, Veni et al. (2018) combined a UNET model with a shape-driven deformable model (level set) for LV segmentation. Similarly, Zhang et al. (2017) used four UNET models to segment the cardiac chambers from four different views. Azarmehr et al. (2019) used both UNET and SegNet to segment the LV endocardium from the PLAX view. Their experimental results suggest the superiority of UNET as compared to SegNet for LV segmentation. Jafari et al. (2018) integrated shape information and motion (i.e., optical flow) with UNET for segmenting LV in A4C view. Other works that use encoder-decoder architecture for echo LV segmentation include (Chen et al., 2020; Zamzmi et al., 2020; Moradi et al., 2019; Leclerc et al., 2019).

The dilation segmentation architecture removes the down-sampling operations and up-samples the corresponding convolutional filters (dilated or atrous convolutions) to obtain high-resolution feature maps. Examples of dilation architectures include DeeplabV3 ((Yurtkulu et al., 2019), dilated convolutions) PSPNet (Zhao et al. (2017), pyramid pooling), and Global Convolutional Network ((Peng et al., 2017), large kernels). Several works used dilation architecture to perform cardiac segmentation in echo images. For example, Chen et al. (2019) and Teng et al. (2020) investigated and compared the performance of PSPNet along with other networks for LV segmentation.

Although the encoder-decoder and dilation architectures achieved excellent segmentation performance, the dilation convolutions and skip connections in these architectures increase the computational cost and memory overhead leading to a slow inference speed. To solve this issue, a novel segmentation architecture, known as Bilateral Segmentation Network (BiSeNet), has been proposed to achieve a good trade-off between accuracy and speed (Yu et al., 2018). BiSeNet uses a dual-pathway to concurrently generate low-detail and high-level semantic information. The detail path is shallow and only has three convolution layers with large spatial size while the semantic path is lightweight for rapid down-sampling. As these paths concurrently generate feature maps, the efficiency increases significantly without impacting the performance (Yu et al., 2018).

To summarize, our literature review reveals that current echo segmentation methods use an encoder-decoder architecture (e.g., UNET) initialized randomly or with ImageNet weights. This architecture runs at a slow inference speed due to computational complexity and memory overhead caused by the skip connections. Our review also reveals that most current methods (1) use a separate model for cardiac region localization prior to segmentation or (2) apply segmentation to the entire image. In medical images, it is common that the target region occupies a relatively small portion of the image (Hesamian et al., 2019; Badshah et al., 2020). Hence, considering the entire image would add noise caused by irrelevant portions and might lead to the segmentation network being biased toward the background.

1.2. Contributions

Our specific contributions can be summarized as follows.

- We devise an efficient system for real-time echo image retrieval (view classification and quality assessment), segmentation, and quantification based on a self-supervised echo-specific representation (see Fig. 1, a). Specifically, the echo tasks are fine- tuned by transferring relevant knowledge from an echo-specific representation constructed using self-supervised learning. Our results show that fine-tuning based on the echo-specific representation enhances the generalization of the target echo tasks and leads to faster convergence.
- We propose a lightweight multi-head model for image retrieval (Fig. 1, b). This model is based on MobileNetV2 (Sandler et al., 2018) but has fewer layers and a fuzzy pooling layer instead of the average pooling layer. The fuzzy pooling operation can handle the uncertainties presented in the images due to the speckle noise. The proposed model achieved superior performance, in terms of accuracy and computations, as compared to the state-of-the-art models.
- We present a Trilateral Attention Network (TaNet) for real- time cardiac region segmentation (Fig. 1, c). TaNet, inspired by BiSeNet (Yu et al., 2018), contains a Spatial Transformer Network (STN) to localize specific regions of interest (ROIs) while learning the context relationship among them. It then uses three pathways for capturing textural (local binary patterns [LBP]) features, low-level details, and high-level semantic information. The proposed TaNet is trained end-to-end to jointly optimize the localization and segmentation tasks for region-based segmentation.
- We present temporal quantitative curves of different cardiac re- gions (Fig. 1, d). Although the estimation of cardiac indices in ED or ES frames represents the clinical standard, we believe quantifying indices and plotting them over time would provide a better understanding of cardiac pattern and function.

To enable other researchers building upon our system, we made the source code available in the project ‘s Github page. The rest of this paper is organized as follows. Section 2 lists the datasets used in our work followed by the system description in Section 3. We provide the experimental setup and results in Section 4, which is followed by conclusions.

2. Materials

We used four datasets in this work: EchoNet-Dynamic dataset (Ouyang et al., 2020), NIH IVC dataset, NIH PLAX dataset, and NIH Doppler dataset. The first dataset is primarily used to train the self-supervised echo-specific representation while the others are used to fine-tune the echo tasks.

In all datasets, each echo study has three ground truth (GT) labels. The first GT label determines the view of the recorded echo as A4C, inferior vena cava (IVC), Doppler, etc. The second GT label determines the quality of the recorded echo as low quality or moderate/good quality. When determining the quality label, the following criteria were considered:

(1) the cardiac region is clearly present and (2) the edges or boundaries of the cardiac regions are clear enough to separate them. Both the view and quality labels are used for the echo retrieval task (Fig. 1, b). The third type of GT is the segmentation masks, which are used to learn the segmentation of cardiac regions (Fig. 1, c) from the background. In addition to these labels, each echo clip or image has cardiac biomarkers provided by a senior echocardiographer and further verified by a certified cardiologist.

2.1. EchoNet-dynamic dataset

EchoNet-Dynamic dataset (Ouyang et al., 2020) contains videos collected in real clinical practice. It contains 10,036 B-mode (A4C) videos collected from 10,036 random patients who underwent an echocardiography exam between 2006 and 2018. The videos were acquired using iE33, Sonos, Acuson SC2000, Epiq 5G, or Epiq 7C ultrasound machines. The number of video frames ranges from 24 to 1002 with a mean acquisition rate of 51 frames per second (FPS). In the processing stage, the videos were cropped, masked to remove protected health information (PHI), and resized to 112 × 112 pixel resolution. Each video in this dataset has an ejection fraction (EF) biomarker provided by human experts. Clinically, EF is calculated as follows:

$$EF = \frac{EDV - ESV}{EDV} \tag{1}$$

where EDV (ED volume) and ESV (ES volume) are computed, based on LV tracings, using the modified Simpson's rule.

EchoNet-Dynamic is one of the few publicly available echo datasets that can be used to train deep learning models. In this work, we used 80% of EchoNet-Dynamic to train the shared echo-specific representation (Fig. 1, a). We used the remaining 20% for fine-tuning and evaluating the echo target tasks. Precisely, we used 1600 patients of the fine-tuning dataset (≈2000 patients) for training and validation (10-fold cross validation). The remaining 400 patients are used as an independent test set.

2.2. NIH IVC dataset

This dataset contains 268 IVC videos collected from 264 patients. The videos were acquired using iE33, GE E9, GE Vivid 7, GE Vivid E95, and Acuson Sequoia C512 ultrasound machines, and have a spatial resolution of 800 × 600 pixels. Each video is associated with the following biomarkers: the IVC diameter and the estimated right atrium pressure (RAP) based on the IVC collapsibility during inspiration. These values are provided by an expert sonographers;, and verified further by an expert cardiologist. Clinically, the IVC thickness is measured perpendicular to the long axis of the IVC within ≈ 2.0 cm (cm) of the right atrium (RA) as shown in Fig. 2. Then, the RAP can be estimated based on IVC as follows:

$$RAP = \begin{cases} \text{if IVC } \leq 21mm \text{ and collapse } > 50\% \Rightarrow \text{RAP} = 5 \\ \text{if IVC } \leq 21mm \text{ and collapse } < 50\% \Rightarrow \text{RAP} = 10 \\ \text{if IVC } > 21mm \text{ and collapse } < 50\% \Rightarrow \text{RAP} = 15 \\ \text{if IVC } > 21mm \text{ and without collapse } \Rightarrow \text{RAP} = 20 \end{cases} \quad (2)$$

The collapsibility indicates that IVC's diameter (measured in millimeters, mm) increases or decreases by 50%. RAP is measured in millimeters of mercury (mmHg). In addition to IVC and RAP values, the quality of each video was assessed for acceptable quality, and the binary mask of IVC region is provided. The images of IVC dataset were resized to 512 × 512 pixels using bicubic interpolation. We also performed mean normalization to ensure that the images have a similar distribution. We used 80% (patient-level) of IVC dataset to fine-tune the echo tasks (Fig. 1, b,c). This fine-tuning set is further divided into training and validation using 10-fold cross validation. To evaluate the generalization of the fine-tuned models, we used the remaining 20% of IVC dataset as an independent testing set.

2.3. NIH PLAX dataset

This dataset contains 68 PLAX videos collected from 60 patients. The videos were acquired using iE33, GE E9, GE Vivid 7, GE Vivid E95, and Acuson Sequoia C512 ultrasound machines, and have a spatial resolution of 800 × 1024 pixels. Echo video with PLAX view shows the following cardiac regions: LV, right ventricle (RV), left atrium (LA), septal wall (SW), posterior wall (PW), and aorta. Each video has two biomarker values provided by an expert sonographer and were further verified by an expert cardiologist. These values are PW thickness (PWT) and SW thickness (SWT). Clinically, these values are measured perpendicular to LV's long axis at the level of mitral valve (MV) leaflet tips as shown in Fig. 3. In addition to SWT and PWT, the quality of each video was assessed for acceptable quality, and the masks are provided for the following cardiac regions: LV, SW, PW, RA, and LA.

The images of this dataset were resized to 512 × 512 pixels using bicubic interpolation. We also performed mean normalization to ensure that the images have a similar distribution. We used 80% (patient-level) of this dataset to fine-tune the echo tasks (Fig. 1, b, c). We then used 10-fold cross validation to divide this fine-tuning set (80%) into training and validation. As an independent testing set, we used the remaining 20% set.

2.4. NIH Doppler dataset

This dataset contains images showing continuous wave and pulsed wave Doppler flows collected from patients who were referred for echocardiographic examination in the Clinical Center at NIH.

The Doppler traces of the mitral valve flow (MV), mitral annular flow (MA), and tricuspid regurgitation flow (TR) were acquired using different commercial echocardiography systems including Philips iE33, GE Vivid 95, and GE Vivid E9. Each Doppler image has a flow type label (TR, MV, or MA) and a segmentation mask provided by an expert technician, which separates the spectral envelope from the background. Besides, the expert technician assessed the quality of images as low- or good-quality and measured the maximum velocities for TR,

MV (E and A), and MA (E'). Clinically, the maximum velocities are measured by finding the peaks of Doppler envelopes. The images of the dataset were resized using bicubic interpolation, and normalized to ensure a similar distribution. We used 80% (patient-level) of this dataset to fine- tune the target echo tasks. This fine-tuning set is further divided into training and validation using 10-fold cross validation. For evaluation, we used the remaining 20% (patient-level) as a testing set.

NIH datasets (IVC, PLAX, Doppler) follow NIH policy for the use of specimens/data (OHSRP determination #18-NHLBI-00686). The use of the de-identified videos was excluded from IRB review per 45 CFR 46.

3. Real-time echocardiography analysis & quantification

Fig. 1 presents the main components of the proposed system, which are self supervised echo-specific representation, image retrieval (view and quality classification), cardiac segmentation, and quantification.

3.1. Self-supervised echo representation

We trained a self-supervised denoising autoencoder to learn echo-specific representation (Fig. 1, a). In the encoder part of the autoencoder, we used a smaller version of MobileNetV2 (Sandler et al., 2018), which we call MobileNetV2-s. As compared to MobileNetV2 (Sandler et al., 2018), MobileNetV2-s has only 5 inverted residual bottleneck blocks and a final fuzzy pooling layer (Diamantis and Iakovidis, 2020) instead of average pooling layer. It has 55,620 parameters and is less than ≈ 2 megabytes (MB). This makes MobileNetV2-s more suitable for analysis in embedded real-time systems. Further, the replacement of traditional pooling with fuzzy pooling allows better handling of the uncertainty caused by the speckle noise. The decoder part of the autoencoder is the reverse of MobileNetV2-s encoder.

The entire MobileNetV2-s-based autoencoder is trained, via self-supervised learning, using the large-scale EchoNet-Dynamic dataset (80%) to learn echo-specific features. Using self-supervised learning to build echo-specific representation allows exploiting available large-scale datasets for creating better initialization and transferring relevant knowledge (i.e., echo weights) to target tasks, which have relatively small datasets. This can lead to better generalizability and faster convergence as discussed thoroughly in Rajaraman et al. (2021) ; Zamzmi et al. (2021). The autoencoder is trained to minimize the mean square error (MSE) with a batch size of 64 for 128 epochs, root mean square propagation (RMSprop), and initial learning rate of 1×10^{-3}.

3.2. Real-time echo image retrieval

The image retrieval component retrieves a specific view with acceptable quality. As shown in Fig. 4, this component is a lightweight model with a shared echo modality-specific encoder and two heads: (1) view classification head to identify the cardiac view and (2) quality assessment head to assess the quality of the identified view.

For the shared echo-specific encoder, we used the encoder part of the MobileNetV2-s autoencoder. As mentioned above, MobileNetV2-s encoder has only 5 inverted residual bottleneck blocks and a final fuzzy pooling layer (Diamantis and Iakovidis, 2020) instead of average pooling layer. We then attached two heads, namely view classification head and quality assessment head, to MobileNetV2-s encoder as shown in Fig. 4. Each head has the following layers: global average pooling (GAP), dropout, fully connected, and softmax layers. We fine-tuned each head along with its encoder as follows. First, we initialized MobileNetV2-s encoder with the echo-specific weights followed by fine-tuning the echo-specific encoder and the view classification layers using the datasets presented in Section 2. The view classification head is fine tuned to minimize the categorical cross entropy (CCE) loss using stochastic gradient descent (SGD) optimizer. We used a batch size of 32, for 32 epochs, and an initial learning rate of 1×10^{-3}. This head classifies a given echo as A4C, IVC, PLAX, or Doppler. Similar to the view classification head, the quality assessment head is fine-tuned to minimize binary cross entropy (BCE) loss using SGD optimizer with a batch size of 16, for 32 epochs, and an initial learning rate of 1×10^{-3}. This head classifies a given echo view as good quality or bad quality. In clinical practice, echocardiographers visually identify echo views and manually exclude low-quality echoes as they lead to inaccurate measurements. Since our image retrieval component is lightweight, it enables real-time echo view classification and quality assessment in clinical practice.

3.3. Real-time echo segmentation

Fig. 5 depicts our TaNet for cardiac region segmentation. TaNet localizes ROIs using a localization component and then uses three pathways for learning rich textural, low-level, and context features. The entire network is fine-tuned end-to-end to learn localization and segmentation. The joint learning of localization and segmentation within the same network prevents unnecessary repetitions of training individual models in isolation and allows the network to focus on specific ROIs (i.e., cardiac regions) while learning the relationships among them.

3.3.1. Localization—CNNs operate on the whole image and are limited by the spatial invariance of input data. The traditional approach for handling these issues involves using separate models for spatial transformation and localization. Jaderberg et al. (2015) proposed a more efficient transformation network, called STN, for applying spatial transformations (e.g., scaling, translation, attention) to the input image or feature map without additional training supervision. STN is a plug-and-play module that can be easily inserted into existing CNNs. It is also differentiable in the sense that it computes the derivative of the transformations within the module, which allows learning the loss gradients with respect to the module parameters.

In medical images, it is common that the target ROI occupies a relatively small portion of the image. Hence, considering the entire image for segmentation would add noise caused by irrelevant regions. Inspired by Yin et al. (2021), we use STN for focusing the attention of the segmentation on a specific ROI while suppressing irrelevant regions. We explain next the main components of the STN module, which are the localization network (L), grid generator (G), and sampler (S).

Localization network (L): this network takes an image (z) with different ROIs as input and creates, for each ROI, the spatial trans- formation parameters (θ):

$$\theta = L(z) \tag{3}$$

where z is a rough segmentation mask with coarsely labeled cardiac ROIs and $\theta \in \mathbb{R}^{N \times 2 \times 3}$, N represents the number of cardiac region(s) in the input image. In case of A4C, IVC, PLAX, and Doppler views, the number of ROIs (N) equal to 4 (LV, LA, RV, and RA), 1 (IVC), 5 (LV, RV, LA, SW, and PW), and 1 (Doppler envelope), respectively.

To generate the rough segmentation mask (z) with coarsely labeled ROIs, we used the MobileNetV2-s-based autoencoder with its weights and fine-tuned it for coarse segmentation. It is important to note that the coarse segmentation is performed only once as a pre-training step to get a rough location for each ROI and estimate its transformation matrix (θ). As discussed in Yin et al. (2021), providing a coarse segmentation of different ROIs allows the localization network to (1) generate the transformation parameters (θ) for these regions and (2) learn the context relationship among them. The output of the coarse segmentation is an image (z) with coarse semantic labels corresponding to different ROIs.

Given an input image $I \in \mathbb{R}^{C_I \times H_I \times W_I}$, where C_I, H_I, and W_I represent the image channels, height, and width, respectively, the output of the coarse segmentation model $\left(z \in \mathbb{R}^{H_z \times W_z}\right)$ can be expressed as:

$$z = CSM(I) \tag{4}$$

where CSM stands for the coarse segmentation model (MobileNetV2-s based autoencoder). The predicted rough mask (z) is then sent to the localization network (L) to generate the trans- former parameter matrix $\left(\theta \in \mathbb{R}^{N \times 2 \times 3}\right)$ for each region (Eq. (3)). Our localization network (L), which is used to estimate θ for each ROI, has eight convolutional layers and a final regression layer to generate the $N \times 2 \times 3$ spatial transformation matrix (θ). For each ROI, θ is defined as follows:

$$\theta = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \end{bmatrix} \tag{5}$$

where s_x, s_y, t_x, and t_y parameters, which are learned by L, allow cropping, translation, and isotropic scaling.

Grid generator (G): given $\theta \in \mathbb{R}^{N \times 2 \times 3}$, the relevant parts of the image (i.e., $ROI \in \{1, 2, .., N\}$) are sampled into a sampling grid G of pixels $G_i = \left(x_i^t, y_i^t\right)$. Specifically, the pointwise affine transformation is computed for each ROI as follows:

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \theta(G_i) = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{6}$$

where $\left(\chi_i^S, y_i^S\right)$ are the source coordinates in the input feature map that define the sample points, $\left(\chi_i^t, y_i^t\right)$ are the target coordinates of the grid in the output feature map, and θ is the affine transformation matrix.

Bilinear sampler (S): To perform the spatial transformation, θ (for each ROI) and the sampling grid are sent to a bilinear sampling kernel to produce N output maps $V_{1:N}$, $V \in \mathbb{R}^{C \times H' \times W'}$ corresponding to N ROIs; C, H', and W' are the grid's number of channels, height, and width, which is the same in the input and output.

Each $\left(x_i^S, y_i^S\right)$ coordinate in the sampling grid ($\theta(G_i)$) defines the spatial location in the input where the bilinear sampler is applied to get the value at a particular pixel in the output V. This can be written as:

$$V_i^c = \sum_n^H \sum_m^W ROI_{nm}^c max(0,\ 1 - |x_i^s - m|) max(0, 1 - |y_i^s - n|) \tag{7}$$

where ROI_{nm}^c is the value at location (n, m) in channel c of the input ROI, $ROI \in \{1, 2, .., N\}$, and V_i^c is the output value for pixel i at location $\left(\chi_i^t, y_i^t\right)$ in channel c. Since the bilinear sampling (Eq. (7)) is differentiable, it allows the gradient loss to flow back to the input feature map, sampling grid coordinates and, therefore, to the transformation parameters θ and the localization network (L).

After localizing the relevant ROIs in the input image, they are sent to the pathways for pixel-wise prediction as shown in Fig. 5. Finally, the segmented ROIs can be remapped (Yin et al., 2021) to their original positions using a reverse grid transformer (G^{-1}).

3.3.2. Segmentation—The segmentation is performed using three pathways: spatial or detail pathway (*SP*), handcrafted pathway (*HP*), and context pathway (*CP*). Each of these pathways extract a unique set of features as described next.

Spatial Pathway (SP): To extract rich low-level details at a low computational cost, a shallow pathway that has three convolutional layers with high channel capacity is adopted. Specifically, we used three blocks, each containing a 3 × 3 convolutional layer with stride of 2 followed by batch normalization and ReLU activation. The number of filters in the first, second, and third blocks are 64, 64, and 128, respectively. This pathway outputs feature maps that are $\frac{1}{8}$ of the input image size as shown in Fig. 5.

Handcrafted pathway (HP): Depending on the medical imaging modality and the application, the standard convolutional kernels can be replaced by handcrafted-based kernels to extract a unique set of statistical, geometrical, or textural features.

As compared to the handcrafted-based methods, the main strength of deep learning is its ability to learn features at different levels of abstraction, which allows learning complex functions that map the input to the output. However, these complex functions may be

generic. On the other hand, hand-crafted descriptors or kernels are designed to extract specific features (e.g., textural, geometric) that may be different from the ones extracted by deep learning models. For example, the textural features (e.g., LBP) have strong ability to differentiate small differences in texture and topography especially at the boundaries between complex regions with challenging separations. Due to this, handcrafted descriptors are still widely and successfully used in different medical imaging modalities including ultrasound and Chest X ray (CXR).

The main limitations of handcrafted features are that they are computed at a single level of abstraction contrary to the deep features which are extracted at different levels; their performance relies on a set of parameters (usually determined manually based on experience); and, therefore, they exhibit limited generalizability. In this work, we adopted the work proposed by JuefeiXu et al. (2017) to address these issues and integrated handcrafted kernels into CNN learning. Specifically, being able to extract, at different levels of abstraction, a unique set of textural features that are complementary to the deep features while generalizing the parameters of handcrafted kernels in a learnable framework is the main motivation of creating a custom handcrafted-encoded CNN pathway for segmentation.

Similar to the spatial pathway (*SP*), we add a handcrafted pathway (*HP*) with three convolutional blocks, but replace the standard convolutional filters with LBP filters. These LBP-encoded convolutional kernels are used to extract rich texture features from the echo images. The mathematical formulation of these LBP-encoded convolutional kernels is presented in detail in Appendix A. Each LBP block has a layer with fixed anchor weights (m) followed by a second layer with learnable convolutional filters of size 1×1. We generated the anchor weights stochastically with different ranges of sparsity. As similar to propagating gradients through layers with learnable and unlearnable parameters (e.g., ReLU, Max Pooling), the entire path can be trained by back propagating the gradients through the anchor weights as well as the learnable weights. In other words, we leave the anchor weights unaffected and only update the weights of the learnable filters.

One might argue the unnecessity of the handcrafted path (HP) as the traditional CNNs can theoretically approximate any signal. While this may be true, it is important to note that the traditional pathway in our network (SP) is shallow with only three convolutional layers, limiting its ability to extract diverse textural features in different orientations. Recall that we used a shallow spatial path to decrease the latency and computational burden. Hence, the handcrafted path is used to augment the spatial path, and extract rich textural features in different orientations, without increasing the computational burden, while the spatial path is used to extract general low-level details from the image; finally, the context path (CP) is used for fast-downsampling of the feature map to obtain a sufficient receptive field. Our experimental results demonstrate that the use of LBP (HP) along with low-level features (SP) provides a stronger, more diverse and representative feature descriptors. These results are consistent with previous works in the literature (e.g., Tang et al. (2020)) that discuss the discriminative power of LBP and suggest the use of ensemble or fusion of both deep and LBP features to boost the performance of classification in medical (Liu et al., 2019; Zamzmi et al., 2019a; Francis and Pandian, 2021; Yasar and Ceylan, 2021) and non-medical images (Wang et al., 2019; Yang et al., 2020; Tang et al., 2020).

Context Pathway (CP): Since the size of the receptive field highly impacts the performance of segmentation, several methods are proposed (e.g., pyramid pooling He et al. (2015)) to obtain sufficient receptive fields. However, most of these methods have high computational complexity and memory consumption. Inspired by Yu et al. (2018), we used a lightweight model (i.e., *Xception*) for fast-downsampling of the feature map to obtain a sufficient receptive field and encode high level context information. Then, a global average pooling is added to the top of the lightweight model to provide the maximum receptive field with global context information. Finally, the output of the global average pooling is up-sampled as shown in Fig. 5.

In summary, TaNet has three pathways for extracting unique sets of low-level (SP), textural (HP), and context (CP) information. As SP and HP have only three layers, they are not computationally intensive. The CP uses a lightweight model for rapid down-sampling. All the three pathways extract complementary features concurrently, which further increases the network efficiency.

Pathways Fusion: As the aforementioned pathways extract different feature embeddings, the simple summation of these representations or embeddings can degrade the performance and complicate the network optimization. To efficiently com- bine these pathways, we adopted the approach proposed in Yu et al. (2018) and fused the features as shown in Fig. 6. First, we concatenate the pathways' outputs and then use batch normalization to balance the different scales of the features. Then, the concatenated features are combined into a single feature vector (O_{concat}). This feature vector is sent to a global pooling followed by a convolutional layer (1 × 1), ReLU activation, convolutional layer (1 × 1), and finally Sigmoid function to generate the weight vector v_{concat}. This weight vector is used to re-weight the concatenated feature vector (O_{concat}) as follows:

$$O_{out\ put} = O_{concat\ .} \times v_{concat\ .} + O_{concat\ .} \tag{8}$$

3.3.3. Loss function—The loss function of the entire TaNet network can be defined as follows:

$$L = \frac{1}{N}\sum_{i}^{N} L_{Seg}\left(ROI_i, ROI_i^{gt}\right) \tag{9}$$

where ROI_i, $i \in \{1, 2, ..., N\}$ represents the pixel-wise prediction for each ROI, ROI_i^{gt} represents the corresponding ground truth, and N represents the total number of ROIs. Similar to Yu et al. (2018), our segmentation loss (L_{Seg}) consists of principal and auxiliary loss functions. The principal loss function are used to supervise the output of the whole network while the auxiliary functions are used to supervise the output of the context pathway. Mathematically, the segmentation loss function (L_{Seg}) can be defined as:

$$L_{seg}(Y, W) = L_p(Y, W) + \alpha_1 L_{aux_1}(Y_1, W) + \alpha_2 L_{aux_2}(Y_2, W) \tag{10}$$

where L_p is the principal loss (softmax) function, Y is the final segmentation for each ROI, W is the learnable parameters, L_{aux1} and L_{aux1} are the auxiliary loss functions (softmax) for the context pathway, Y_1 and Y_2 represent the output features from CP. To balance the weight of the loss functions, we empirically set α_1 and α_2 to 1.

3.3.4. Training—

We trained TaNet in two stages: pre-training and fine-tuning.

Pre-training: As shown in Fig. 7, this stage has two steps: (1) pre-training the coarse segmentation model (*CSM*) and (2) pre-training the localization network (*L*). We trained a coarse segmentation model (*CSM*) to get rough predictions of different ROIs. Then, we trained the localization network (*L*) to generate estimations of $\theta_{1:N}$. It is important to note that this stage is performed only to estimate the approximate location of different ROIs.

To generate rough ROIs, the coarse segmentation model (*CSM*) is trained with 32 batch size and a learning rate of 1×10^{-3}. We used Adam optimizer to minimize the loss between GT masks and the predicted coarse segmentation masks. Then, we used the out- put of the coarse segmentation (*z*) as input to the localization network (*L*). The localization network (*L*) is trained with 32 batch size and 1×10^{-3} learning rate to optimize the Smooth L1 loss. The smooth L1 loss, which is commonly used for box regression, is less sensitive to outliers Wang et al. (2020). The localization network aims to minimize the smooth L1 loss between predicted θ and ground truth θ_r^{gt}:

$$L1_{\text{smooth}} = \begin{cases} \text{if } \left|\theta_r - \theta_r^{gt}\right| < 1 \Rightarrow 0.5\left(\theta_r - \theta_r^{gt}\right)^2 \\ \text{otherwise } \Rightarrow \left|\theta_r - \theta_r^{gt}\right| - 0.5 \end{cases} \tag{11}$$

where θ_r and θ_r^{gt} are the predicted and ground truth transformation matrices for a specific region $r \in \{1, 2,..N\}$. The ground truth transformation matrix is calculated for each region $\left(\theta_r^{gt}, r \in \{1, 2,..N\}\right)$ as follows. We calculated the central coordinates (x, y) for each coarsely segmented ROI_r $(r \in \{1, ..N\})$ and estimated θ_r^{gt} as:

$$\theta_r^{gt} = \begin{bmatrix} S_x & 0 & t_x \\ 0 & S_y & t_y \end{bmatrix} = \begin{bmatrix} \frac{\bar{W}}{W_z} & 0 & -1 + \frac{2x}{W_z} \\ 0 & \frac{\bar{H}}{H_z} & -1 + \frac{2y}{H_z} \end{bmatrix} \tag{12}$$

where H_z and W_z represent the height and width of the labeled image (*z*) while $\bar{H}$ and $\bar{W}$ represent the height and width of a fixed window size. The ground truth transformation matrix $\left(\theta_r^{gt}\right)$ is generated for each region based on Eq. (12).

End-to-end fine-tuning: With the pre-trained parameters (stage 1) loaded, we fine-tuned the entire network end-to-end. We used Adam optimizer and an initial learning rate of 1×10^{-3}. The optimization goal is to minimize the loss (Eq. (9)) between ROIs prediction (ROI_i) and ROIs ground truth labels $\left(ROI_i^{gt}\right)$.

3.4. Real-time echo quantification

Instead of extracting cardiac biomarkers in specific frames (e.g., ED and ES), the temporal analysis of cardiac indices over frames has great applicability in clinical cardiology practice and research. It provides information about the mechanics of cardiac chambers and captures how they change over time. We present next the steps for computing cardiac biomarkers from the segmented regions.

3.4.1. Post processing—Prior to the delineation of the cardiac boundaries, we perform morphological cleaning to remove any isolated pixels and only keep the closed region of interest. Next, we compute the contour of the clean regions using Moore-Neighbor tracing algorithm modified by Jacob's stopping criteria Reddy et al. (2012). Next, the de- lineated region is divided into equal segments (or sectors). These segments are then used to compute several cardiac indices.

3.4.2. Biomarkers estimation—We estimate the following biomarkers: IVC diameter (IVCD), right atrium pressure (RAP), LV internal diameter (LVID), septal wall thickness (SWT), posterior wall thickness (PWT), maximum velocities, and ejection fraction (EF). IVCD and RAP biomarkers are computed from IVC dataset, LVID, SWT, and PWT are computed from PLAX dataset, maximum velocities are computed from Doppler dataset, and EF is computed from A4C view (EchoNet dataset). After computing these biomarkers, we compare them with the manual human-based measures.

IVCD and RAP Calculation: To compute IVCD, we find the major axis of the sub region that is located approximately 2 cm proximal to the ostium of RA. We then compute the Euclidean distance between the endpoints of the major axis. Finally, we convert the computed pixel distance into millimeters (mm) as follows:

$$Pixel = (25.4 \div dpi)\text{mm} \tag{13}$$

$$IVCD = IVCD_{distance} \times Pixel \tag{14}$$

where dpi is the dots per inch in a given clip. After computing $IVCD$, we construct the $IVCD$ curve by plotting $IV\,CD$ values over frames. Next, we use the Savitzky-Golay filter to obtain a smoothed $IV\,CD$ curve. The smoothed curve is then used to compute RAP as follows. First, we compute IVC collapsibility based on the differ- ence between the absolute maximum peak and minimum valley in the $IV\,CD$ curve:

$$Collapsibility = \frac{IVC_{peak} - IVC_{valley}}{IVC_{peak}} \times 100\% \tag{15}$$

Finally, the RAP value is computed by plugging the IVC diameter and collapsibility values into Eq. (2).

LVID, SWT, and PWT Calculation: To compute LVID, the sub region or segment where the leaflet tips of MV are located should be determined. Therefore, we track these points

within the segmented LV region. We then create a line passing through the major axis of that sub region and extending in both directions as shown in Fig. 8. We compute LVID, SWT, and PWT by finding the intersection points between the line and cardiac boundaries followed by computing the Euclidean distances between these points. For example, SWT is calculated by computing the Euclidean distance between P_1 and P_2 in Fig. 8. Finally, we convert the computed distances into mm unit using Eqs. (13) and (14). Similar to IVC analysis, we construct the curves for LVID, SWT, and PWT by plotting their values over frames and smooth the signals using the Savitzky-Golay filter. These signals show the values at ED and ES and provide information about the mechanics of cardiac chambers.

Doppler Calculation: To compute the maximum velocities from the Doppler flows, we detect the peaks of the delineated envelopes as shown in Fig. 8. We first convert the delineated boundary into 1D signal followed by smoothing the signal. Next, we find all local maxima and select the k maximum values of all detected local maxima as our maximum velocities.

Ejection Fraction: After segmenting the LV from A4C view, LV tracings are used to calculate the volumes. These volumes are then plugged into Eq. (1) to compute EF. The computed EF values are then compared with the manual EF obtained by human experts.

4. Results and discussion

We evaluate the performance of the proposed system, which consists of echo retrieval, cardiac region segmentation, and quantification. The performance of classification is reported using accuracy, precision, recall, F-score, Area Under the Curve (AUC), and the Matthews correlation coefficient (MCC), while the performance of segmentation is reported using intersection over union (IoU) and Dice (F1) scores; finally, the performance of indices quantification is reported using Pearson correlation coefficient (CC) and Bland-Altman plot. We also reported the computational complexity and frames per second (FPS).

For all models, the Talos toolbox[1] is used for selecting the model hyperparameters. We conducted all experiments using Py-torch and performed training and inference on NVIDIA GTX1080Ti GPU.

4.1. Real-time echo retrieval

Prior to training for echo retrieval, we used the self-supervised echo-specific representation to learn low-level features that are common among all echo tasks. Then, the shared representation was truncated at the deepest convolutional layer and appended with two heads, one for view classification and another for quality assessment. We then initialized the model (encoder + two heads) with the pretrained (echo-specific) weights and fine-tuned it using the datasets presented in Section 2.

Specifically, the set that is used for fine-tuning the retrieval model contains 80% of IVC dataset, 80% of PLAX dataset, 80% of Doppler dataset, and 16% of EchoNet-Dynamic

[1] https://github.com/autonomio/talos

dataset. This fine-tuning set is further divided into training and validation using 10-fold cross validation. To enlarge the training set and introduce diversity, we applied the following operations: random rotation (−15° to +15°), horizontal and vertical shift (−0.25, 0. 25), scale [0.75, 1, 1.25], and horizontal and vertical flip. Finally, the fine-tuned model is evaluated using an independent test set that contains the remaining 20% of IVC, PLAX, and Doppler datasets as well as 4% of EchoNet-Dynamic dataset.

To evaluate the performance of our echo retrieval model, we compared the performance of the view classification head and the quality assessment head with the state-of-the-art models (i.e., VGG16 and ResNet18). We performed two ablation experiments to validate the impact of fuzzy pooling and echo-specific weights on the performance.

Ablation for fuzzy pooling: To evaluate the impact of the fuzzy operation on the classification, we performed an ablation experiment in which we used: (1) classification models (proposed, VGG16, and ResNet18) with average pooling, (2) classification models (proposed, VGG16, and ResNet18) with max pooling, and (3) classification models (proposed, VGG16, and ResNet18) with fuzzy pooling. Table 1 showsthe performance of these three cases in terms of accuracy, precision, and sensitivity. The results in Table 1 demonstrate that the use of fuzzy pooling out- performs other pooling operations in both tasks (view classification and quality assessment). We refer the interested reader to Diamantis and Iakovidis (2020) for detailed theoretical description of how the proposed fuzzy operation tackles the uncertainty (caused by speckle noise) naturally propagated from the input to the feature maps.

Ablation for echo-specific representation: In the second ablation experiment, we wanted to evaluate the impact of the echo-specific representation on classification. To do this, we initialized all models with echo-specific, random, and ImageNet weights. We reported the performance for all cases in Tables 2 and 3 for view classification and quality assessment, respectively. From this experiment, we have two observations. First, the echo-specific weights increase the classification performance in most models as compared to random and ImageNet weights. This improvement is attributed to the positive transfer of relevant knowledge from the echo-specific representation to the target echo tasks. Second, the proposed MobileNetV2-s based classification achieved comparable, if not better, performance as compared to the state-of-the- arts while having significantly lower computational complexity. As shown in Table 4, our MobileNetV2-s based model has the lowest training and parameters as well as the smallest memory size.

These results demonstrate the superiority, in terms of performance and computations, of the proposed echo retrieval model, which contains a shared encoder and two heads for view classification and quality assessment. Note that our retrieval model only uses the first frames (chosen empirically as an odd number to pre- vent a tie vote) to determine the input view and quality. After classifying the input echo, the bad quality echoes are excluded from further analysis and the good quality echoes are sent to the segmentation network.

4.2. Real-time echo segmentation

We fine-tuned four TaNet networks, which are ($TaNet_{IVC}$) for IVC dataset, ($TaNet_{PLAX}$) for PLAX dataset, ($TaNet_{Doppler}$) for Doppler dataset, and ($TaNet_{A4C}$) for EchoNet dataset. Prior to fine-tuning these networks, we initialize them with the echo-specific weights. We then fine-tuned them using 80% of IVC dataset, 80% PLAX dataset, 80% of Doppler dataset, and 16% of EchoNet dataset (subject-wise). The sets are further divided into a training set for coarse segmentation (20%) and a training set for fine segmentation (60%). To enlarge the training sets and add diversity, we applied the following operations: random rotation (−15° to +15°), horizontal and vertical shift (−0.25, 0.25), scale [0.75, 1, 1.25], and horizontal and vertical flip.

The fine-tuned TaNet networks are then evaluated using the remaining of the datasets. Note that we did not provide comparison, in terms of semantic segmentation metrics, for $TaNet_{A4C}$ because EchoNet-Dynamic dataset does not provide ground truth masks for LV. However, we gauge $TaNet_{A4C}$ performance by comparing the automated EF extracted based on the segmented LV with the manual EF. We performed two ablation experiments to validate the impact of TaNet components, namely STN for localization and HP for textural representation. We also compared the performance of TaNet with the state-of-the-art models for segmentation including the baseline BiSeNet (Yu et al., 2018), FCN (Long et al., 2015), and UNET (Ronneberger et al., 2015).

Ablation for localization module: To evaluate the impact of STN on the segmentation performance and speed, we integrated STN module to the baseline Bisenet (Yu et al., 2018), FCN (Long et al., 2015), and UNET (Ronneberger et al., 2015) and reported the segmentation results for each cardiac region as shown in Table 5 (IVC dataset), Table 6 (PLAX dataset), and Table 7 (Doppler dataset). We reported the results using the average IoU and F1 score, which are averaged over the test samples.

As observed from Table 5 (IVC), Table 6 (PLAX), and Table 7 (Doppler), integrating STN into segmentation models slightly decreases the inference speed. It, however, improves cardiac region segmentation in most cases. Note that the improvement in segmentation is especially higher in case of the PLAX dataset (Table 6). This can be attributed to the simpler structure of IVC and Doppler images, which contain a single region and the background, as compared to PLAX images with relatively overlapping cardiac regions. Nonetheless, these results demonstrate STN's ability to increase the segmentation performance by focusing the attention of segmentation on the desired region while learning the relationship of different regions in the image.

Ablation for LBP-encoded layers: To evaluate the impact of integrating LBP-encoded kernels on segmentation, we replaced the classical kernels in FCN8, UNET, and BiSeNet with LBP-encoded convolutional kernels. Table 5 (IVC), Table 6 (PLAX), and Table 7 (Doppler) show that using LBP-encoded layers achieved comparable, if not better, performance as compared to the classical convolutional kernels while increasing the inference speed. Note how using LBP-encoded layers improves the performance of the wall (SW and PW) regions (Table 6). Due to the similarity between LV anterior and posterior

walls, the accurate segmentation of LV walls is challenging. These results suggest that (1) LBP-encoded kernels extract a different set of features than the traditional kernels and (2) the ability of LBP-encoded kernels to better differentiate small differences in texture at the boundaries between complex regions with challenging separation. These results are consistent with previous works (e.g., Tang et al. (2020)) that discuss the discriminative power of LBP and suggest to fuse its features with the traditional CNN features to boost the classification performance.

The last rows of Tables 5–7 present the performance of TaNet (*STN*, *SP*, *HP*, *CP*) as compared to the baseline BiSeNet, FCN-8, and UNET (Ronneberger et al., 2015). As shown in the last rows of these tables, the use of STN-based localization and the combination of the textural and low-level features improved the performance even further. The green cells indicate that the proposed TaNet achieved significantly ($p < 0.05$) higher performance as compared to the baseline models. Note that TaNet has a slightly lower speed as compared to BiSeNet (Yu et al., 2018) due to the integration of the localization module (STN) and handcrafted pathway along with the spatial and context pathways. However, this speed is still efficient for real-time medical image analysis. Fig. 9 shows the GT masks and predicted masks (TaNet) for images from IVC, PLAX, and Doppler datasets. Fig. 10 shows segmentation examples generated by the baseline models and TaNet for IVC (1*st* row), PLAX (2*nd* − 4*th* rows), and Doppler (5*th* − 7*th* rows). From the figures, we can visually observe that TaNet outperforms all models in segmenting all cardiac regions.

4.3. Comparison with the state-of-the-arts

This work presents a system that performs echo retrieval and segmentation tasks, and hence, the performance of each task with the state-of-the-art methods need to be reported. Most current methods for echo classification rely on VGG16 (Zhang et al., 2017; Madani et al., 2018; Vaseli et al., 2019; Ghorbani et al., 2020), ResNet18 (Vaseli et al., 2019), and DenseNet161 (Vaseli et al., 2019) while the majority of segmentation methods rely on different extensions of UNET (Zhang et al., 2017; Leclerc et al., 2019), FCN (Chen et al., 2020), and Res-U (Ali et al., 2021).

Table 8 provides a performance summary of the state-of-the- art classification models on our datasets. From the table, we can observe that our MobileNetV2-s model has significantly faster inference time, and it achieved high classification performance with the lowest memory size. Similarly, we compared TaNet to the state-of-the-art segmentation models and reported the results in Table 9. From the table, we can observe that our TaNet achieved the best average IoU for IVC, PLAX, and Doppler datasets with relatively small memory footprint and fast inference time. From Tables 8 and 9, we can conclude that the proposed approach outperforms current ones for echo classification and segmentation, and achieves excellent performance with fast inference speed and relatively small memory size.

4.4. Real-time echo quantification

After the segmented regions are cleaned and delineated, we extract different cardiac biomarkers. The extracted biomarkers are then used to create cardiac curves to provide information about the mechanics of cardiac chambers as shown in Fig. 11. From such

curves, we can estimate the absolute maximum value (highest peak) of a biomarker and the absolute minimum value (lowest Valley). We can also estimate the average maximum and average minimum by averaging the curve's peaks and valleys.

To assess the agreement between the automated values and those estimated by experts, we used Pearson correlation coefficient and Bland-Altman analysis. The first row of Fig. 12 shows correlation and Bland-Altman plots for automated IVC as compared to the manual IVC. From the figure, we can observe that the IVC extracted by the system is highly correlated with the values calculated by human experts. To estimate RAP, we first measured the collapsibility of IVC using Eq. (15). Next, we plugged the absolute IVC value and the percentage of collapsibility into Eq. (2) to estimate RAP. To assess the agreement between the automated RAP values and those estimated by experts, we presented the confusion matrix in the second row of Fig. 12. From the confusion matrix, we can conclude the ability of the proposed system to accurately estimate RAP values based on the segmented IVC region.

In addition to IVC, we measured the agreement between the automated and manual SWT and PWT using correlation and Bland-Altman plots. As shown in Fig. 13, the values extracted by our system highly correlate ($r_{SWT} = 0.99$ and $r_{PWT} = 0.908$) with the wall thickness values estimated by the experts. The lower correlation value of PWT can be attributed to the substantial overlapping be- tween PW and surrounding regions as compared to SW region. In case of echo Doppler, we also measured the agreement between the automated and manual Doppler velocities. The first, second, third, and fourth rows of Fig. 14 show these plots for TR velocity, MV E velocity, MV A velocity, and MA E' velocity, respectively. These plots show strong agreement between the experts' velocity values and the ones extracted by our system. Finally, we measured the agreement between the manual and automated EF and reported the results in Fig. 15. As shown in the figure, the EF values extracted by our system are highly correlated with the GT values provided by experts.

5. Conclusion

The automated interpretation of echo has the potential to change clinical practice through fast, low-cost, portable, and accurate assessment of cardiac structure and function. This work proposed a novel end-to-end system for robust real-time echo retrieval, segmentation, and quantification. The proposed system, which was evaluated using four echo datasets, achieved superior performance in retrieving good quality echo from individual views, segmenting cardiac chambers with complex overlaps, and extracting cardiac measures/ indices that highly agree with expert's scores. Further, the proposed system significantly enhanced computational eiciency as compared to the state-of-the-arts. The high efficiency and performance of our framework would facilitate its deployment and production in real-world applications for bedside echo examination and point-of-care ultrasound.

Acknowledgment

This work was supported by the Intramural Research Program of the National Library of Medicine (NLM), the National Heart, Lung, and Blood Institute (NHLBI), and the Clinical Center, all parts of the National Institutes of Health (NIH). We would like to acknowledge the contributions of Wen Li from NHLBI in the process of

data collection and annotations. We would also like to thank anonymous reviewers whose insightful feedback and constructive suggestions helped in shaping this article into its present form.

Appendix A.: LBP-encoded Convolutional Kernels

LBP (Pietikäinen, 2010) is a theoretically simple and computationally efficient method for summarizing the texture of an image. This method computes the texture pattern around a central pixel in a local neighborhood by comparing the intensity of the neighborhood pixels (p_i) with the central pixel (p_c), and assigning a value of 1 if $p_i > p_c$ and 0 otherwise. Then, LBP code is computed by mapping the binary digits to a decimal number using a base of 2. These aggregated LBP codes characterize the image's texture. Mathematically, the standard LBP is formulated as follows:

$$y_{map} = \sum_{i=1}^{8} \sigma(b_i * \chi_{vec}) \tag{A.1}$$

where x_{vec} is the vectorized input image, b_i are the 2-sparse filters, σ is the non-linear binarization function (e.g., Heaviside step), and y_{map} is the resulting LBP feature map. This formulation has all the components of the standard convolutional layers. Hence, it can be used to formulate a LBP block with two convolutional layers (Juefei-Xu et al., 2017). The first layer has m fixed convolutional filters with non-learnable anchor weights while the second layer has learnable convolutional filters of size 1 × 1 to compute the weighted sum of the activations from the first layer.

The first layer is used to generate LBP feature maps as follows. First, the input image (x_{vec}) is processed by m predefined convolutional filters (anchor weights) b_i, $i \in [m]$ to generate m difference maps; i.e., m represents the number of LBC filters. Then, these maps are activated using differentiable and non-linear activation functions (e.g., ReLU) to generate m bit maps. Finally, the generated bit maps are linearly combined to generate the final LBP feature map. Mathematically, this can be expressed as:

$$y_{map} = \sum_{i=1}^{m} \left(\sum (b_i * x_{vec}) \right) \cdot v_i \tag{A.2}$$

where y_{map} and x_{vec} are the output and input images or feature maps, respectively; b_i is the 2-sparse convolutional filters, m is the number of predefined convolutional filters, σ is the non-linear activation function (ReLU), and V_i is the learnable weights. To compute the weighted sum of the activations in Eq. (A.2), we used a convolution operation with filters of size 1 × 1 in the second layer; i.e., this convolution layer has a learnable weights and it is used to compute the weighted sum of the activations from the first layer.

Therefore, each LBP block has two layers, where the first layer has m unlearnable convolutional filters followed by learnable filters (1 × 1) in the second layer. Each LBP block has a smaller number of learnable parameters as compared to the standard convolutional layer (Juefei-Xu et al., 2017). Specifically, the number of learnable parameters in the LBP layer (with 1 × 1 convolutions) are significantly less than those of a standard convolutional layer for the same size of the convolutional kernel and number of input and

output channels. As shown in Juefei-Xu et al. (2017), the number of learnable parameters in LBP encoded layers are reduced by a factor of 9, 25, 49, 81, 121, and 169 for 3 × 3, 5 × 5, 7 × 7, 9 × 9, 11 × 11, and 13 × 13 convolutional filters, respectively.

References

Abdi AH, Luong C, Tsang T, Allan G, Nouranian S, Jue J, Hawley D, Fleming S, Gin K, Swift J, et al., 2017. Automatic quality assessment of apical four-chamber echocardiograms using deep convolutional neural networks. In: Medical Imaging 2017: Image Processing, Vol. 10133. International Society for Optics and Photonics, p. 101330S.

Agarwal D, Shriram K, Subramanian N, 2013. Automatic view classification of echocardiograms using histogram of oriented gradients. In: 2013 IEEE 10th International Symposium on Biomedical Imaging. IEEE, pp. 1368–1371.

Ali Y, Janabi-Sharifi F, Beheshti S, 2021. Echocardiographic image segmentation using deep res-u network. Biomedical Signal Processing and Control 64, 102248.

Azarmehr N, Ye X, Sacchi S, Howard JP, Francis DP, Zolgharni M, 2019. Segmentation of left ventricle in 2d echocardiography using deep learning. In: Annual Conference on Medical Image Understanding and Analysis. Springer, pp. 497–504.

Badrinarayanan V, Kendall A, Cipolla R, 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence 39 (12), 2481–2495. [PubMed: 28060704]

Badshah N, Atta H, Shah SIA, Attaullah S, Minallah N, Ullah M, 2020. New local region based model for the segmentation of medical images. IEEE Access 8, 175035–175053.

Belous G, Busch A, Rowlands D, 2013. Segmentation of the left ventricle from ultrasound using random forest with active shape model. In: 2013 1st International Conference on Artificial Intelligence, Modelling and Simulation. IEEE, pp. 315–319.

Beymer D, Syeda-Mahmood T, Amir A, Wang F, Adelman S, 2009. Automatic estimation of left ventricular dysfunction from echocardiogram videos. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE, pp. 164–171.

Chen C, Qin C, Qiu H, Tarroni G, Duan J, Bai W, Rueckert D, 2020. Deep learning for cardiac image segmentation: A review. Frontiers in Cardiovascular Medicine 7, 25. [PubMed: 32195270]

Chen H, Zheng Y, Park J-H, Heng P-A, Zhou SK, 2016. Iterative multi-domain regularized deep learning for anatomical structure detection and segmentation from ultrasound images. In: International Conference on Medical image computing and computer-assisted intervention. Springer, pp. 487–495.

Chen S, Ma K, Zheng Y, 2019. Tan: temporal affine network for real-time left ventricle anatomical structure analysis based on 2d ultrasound videos. arXiv preprint arXiv:1904.00631.

Chen Y, Huang F, Tagare HD, Rao M, 2007. A coupled minimization problem for medical image segmentation with priors. International journal of computer vision 71 (3), 259–272.

Cheng J, Foo SW, Krishnan S, 2005. Automatic detection of region of interest and center point of left ventricle using watershed segmentation. In: 2005 IEEE International Symposium on Circuits and Systems. IEEE, pp. 149–151.

Diamantis D, Iakovidis D, 2020. Fuzzy pooling. IEEE Transactions on Fuzzy Systems.

Dong J, Liu S, Liao Y, Wen H, Lei B, Li S, Wang T, 2019. A generic quality control framework for fetal ultrasound cardiac four-chamber planes. IEEE journal of biomedical and health informatics 24 (4), 931–942. [PubMed: 31634851]

Dong S, Luo G, Wang K, Cao S, Li Q, Zhang H, 2018. A combined fully convolutional networks and deformable model for automatic left ventricle segmentation based on 3d echocardiography. BioMed research international 2018.

Francis A, Pandian IA, 2021. Early detection of alzheimers disease using local binary pattern and convolutional neural network. Multimedia Tools and Applications 80 (19), 29585–29600.

Ghorbani A, Ouyang D, Abid A, He B, Chen JH, Harrington RA, Liang DH, Ashley EA, Zou JY, 2020. Deep learning interpretation of echocardiograms. NPJ digital medicine 3 (1), 1–10. [PubMed: 31934645]

Guo Y, Wang Y, Nie S, Yu J, Chen P, 2014. Automatic segmentation of a fetal echocardiogram using modified active appearance models and sparse representation. IEEE Transactions on Biomedical Engineering 61 (4), 1121–1133. [PubMed: 24658237]

He K, Zhang X, Ren S, Sun J, 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence 37 (9), 1904–1916. [PubMed: 26353135]

He K, Zhang X, Ren S, Sun J, 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

Hesamian MH, Jia W, He X, Kennedy P, 2019. Deep learning techniques for medical image segmentation: achievements and challenges. Journal of digital imaging 32 (4), 582–596. [PubMed: 31144149]

Iandola F, Moskewicz M, Karayev S, Girshick R, Darrell T, Keutzer K, 2014. Densenet: Implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869.

Jaderberg M, Simonyan K, Zisserman A, Kavukcuoglu K, 2015. Spatial transformer networks. arXiv preprint arXiv:1506.02025.

Jafari MH, Girgis H, Liao Z, Behnami D, Abdi A, Vaseli H, Luong C, Rohling R, Gin K, Tsang T, et al., 2018. A unified framework integrating recurrent fully–convolutional networks and optical flow for segmentation of the left ventricle in echocardiography data. In: Deep learning in medical image analysis and multimodal learning for clinical decision support. Springer, pp. 29–37.

Jiang Y, Gu X, Wu D, Hang W, Xue J, Qiu S, Chin-Teng L, 2020. A novel negative-transfer-resistant fuzzy clustering model with a shared cross-domain transfer latent space and its application to brain ct image segmentation. IEEE/ACM transactions on computational biology and bioinformatics.

Juefei-Xu F, Naresh Boddeti V, Savvides M, 2017. Local binary convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 19–28.

Lacerda SG, da Rocha AF, Vasconcelos DF, de Carvalho JL, Sene IG, Camapum JF, 2008. Left ventricle segmentation in echocardiography using a radial-search-based image processing algorithm. In: 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, pp. 222–225.

Leclerc S, Smistad E, Pedrosa J, Østvik A, Cervenansky F, Espinosa F, Espeland T, Berg EAR, Jodoin P-M, Grenier T, et al. , 2019. Deep learning for segmentation using an open large-scale dataset in 2d echocardiography. IEEE transactions on medical imaging 38 (9), 2198–2210. [PubMed: 30802851]

Liao Z, Girgis H, Abdi A, Vaseli H, Hetherington J, Rohling R, Gin K, Tsang T, Abolmaesumi P, 2019. On modelling label uncertainty in deep neural networks: automatic estimation of intra-observer variability in 2d echocardiography quality assessment. IEEE transactions on medical imaging 39 (6), 1868–1883. [PubMed: 31841401]

Liu D, Liu Y, Li S, Li W, Wang L, 2019. Fusion of handcrafted and deep features for medical image classification. In: Journal of Physics: Conference Series, Vol. 1345. IOP Publishing, p. 022052.

Long J, Shelhamer E, Darrell T, 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440.

Madani A, Arnaout R, Mofrad M, Arnaout R, 2018. Fast and accurate view clas- sification of echocardiograms using deep learning. NPJ digital medicine 1 (1), 6. [PubMed: 30828647]

Marsousi M, Eftekhari A., Kocharian A., Alirezaie J., 2010. Endocardial boundary extraction in left ventricular echocardiographic images using fast and adaptive b-spline snake algorithm. International journal of computer assisted radiology and surgery 5 (5), 501–513. [PubMed: 20232263]

Moradi S, Oghli MG, Alizadehasl A, Shiri I, Oveisi N, Oveisi M, Maleki M, Dhooge J, 2019. Mfp-unet: A novel deep learning based approach for left ven- tricle segmentation in echocardiography. Physica Medica 67, 58–69. [PubMed: 31671333]

Nandagopalan S, Adiga B, Dhanalakshmi C, Deepak N, 2010. Automatic segmentation and ventricular border detection of 2d echocardiographic images combining k-means clustering and active contour

model. In: 2010 Second International Conference on Computer and Network Technology. IEEE, pp. 447–451.

Nolan MT, Thavendiranathan P, 2019. Automated quantification in echocardiography. JACC: Cardiovascular Imaging 12 (6), 1073–1092. [PubMed: 31171260]

Oh JK, Seward JB, Tajik AJ, 2006. The echo manual. Lippincott Williams & Wilkins.

Oktay AB, Akgul YS, 2009. Echocardiographic contour extraction with local and global priors through boosting and level sets. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE, pp. 46–51.

Østvik A, Smistad E, Aase SA, Haugen BO, Lovstakken L, 2019. Real-time standard view classification in transthoracic echocardiography using convolutional neural networks. Ultrasound in medicine & biology 45 (2), 374–384. [PubMed: 30470606]

Ouyang D, He B, Ghorbani A, Yuan N, Ebinger J, Langlotz CP, Heidenreich PA, Harrington RA, Liang DH, Ashley EA, et al. , 2020. Video-based ai for beat-to-beat assessment of cardiac function. Nature 580 (7802), 252–256. [PubMed: 32269341]

Pavani S-K, Subramanian N, Gupta MD, Annangi P, Govind SC, Young B, 2012. Quality metric for parasternal long axis b-mode echocardiograms. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 478–485.

Penatti OA, Werneck R.d.O., de Almeida WR., Stein BV., Pazinato DV., Júnior PRM., Torres R.d.S, Rocha A., 2015. Mid-level image representations for real-time heart view plane classification of echocardiograms. Computers in biology and medicine 66, 66–81. [PubMed: 26386547]

Peng C, Zhang X, Yu G, Luo G, Sun J, 2017. Large kernel matters–improve semantic segmentation by global convolutional network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4353–4361.

Pietikäinen M, 2010. Local binary patterns. Scholarpedia 5 (3), 9775.

Qian Y, Wang L, Wang C, Gao X, 2012. The synergy of 3d sift and sparse codes for classification of viewpoints from echocardiogram videos. In: MICCAI International Workshop on Medical Content-Based Retrieval for Clinical Decision Support. Springer, pp. 68–79.

Rajaraman S, Zamzmi G, Antani SK, 2021. Novel loss functions for ensemble-based medical image classification. Plos one 16 (12), e0261307.

Reddy PR, Amarnadh V, Bhaskar M, 2012. Evaluation of stopping criterion in contour tracing algorithms. International Journal of Computer Science and Information Technologies 3 (3), 3888–3894.

Ronneberger O, Fischer P, Brox T, 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer, pp. 234–241.

Rosenstein MT, Marx Z, Kaelbling LP, Dietterich TG, 2005. To transfer or not to transfer. In: NIPS 2005 workshop on transfer learning, Vol. 898, pp. 1–4.

Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C, 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520.

Santos J, Celorico D, Varandas J, Dias J, 2007. Automatic segmentation of echocardiographic left ventricular images by windows adaptive thresholds. In: Proceedings of the International Congress on Ultrasonics, Vienna, April, pp. 9–13.

Simonyan K, Zisserman A, 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Smistad E, Østvik A, Salte IM, Melichova D, Nguyen TM, Haugaa K, Brunvand H, Edvardsen T, Leclerc S, Bernard O, et al. , 2020. Real-time automatic ejection fraction and foreshortening detection using deep learning. IEEE transactions on ultrasonics, ferroelectrics, and frequency control 67 (12), 2595–2604. [PubMed: 32175861]

Snare SR, Torp H, Orderud F, Haugen BO, 2012. Real-time scan assistant for echocardiography. IEEE transactions on ultrasonics, ferroelectrics, and frequency control 59 (3), 583–589. [PubMed: 22481796]

Tang J, Su Q, Su B, Fong S, Cao W, Gong X, 2020. Parallel ensemble learning of convolutional neural networks and local binary patterns for face recognition. Computer Methods and Programs in Biomedicine 197, 105622.

Teng L, Fu Z, Yao Y, 2020. Interactive translation in echocardiography training system with enhanced cycle-gan. IEEE Access 8, 106147–106156.

Torrey L, Shavlik J, 2010. Transfer learning. In: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI global, pp. 242–264.

Vaseli H, Liao Z, Abdi AH, Girgis H, Behnami D, Luong C, Dezaki FT, Dhungel N, Rohling R, Gin K, et al. , 2019. Designing lightweight deep learning models for echocardiography view classification. In: Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling, Vol. 10951. Inter- national Society for Optics and Photonics, p. 109510F.

Veni G, Moradi M, Bulu H, Narayan G, Syeda-Mahmood T, 2018. Echocardiography segmentation based on a shape-guided deformable model driven by a fully convolutional network prior. In: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018). IEEE, pp. 898–902.

Vrettos A, Azarmehr N, Howard J, Shun-shin M, Cole G, Francis D, Zolgharni M, et al., 2020. Automated assessment of image quality in 2d echocardiography using deep learning.

Wang C, Li D, Li Z, Wang D, Dey N, Biswas A, Moraru L, Sherratt R, Shi F, 2019. An efficient local binary pattern based plantar pressure optical sensor image classification using convolutional neural networks. Optik 185, 543–557.

Wang Q, Ma Y, Zhao K, Tian Y, 2020. A comprehensive survey of loss functions in machine learning. Annals of Data Science 1–26.

Wong KK, Fortino G, Abbott D, 2020. Deep learning-based cardiovascular image diagnosis: a promising challenge. Future Generation Computer Systems 110, 802–811.

Wu H, Bowers DM, Huynh TT, Souvenir R, 2013. Echocardiogram view classification using low-level features. In: 2013 IEEE 10th International Symposium on Biomedical Imaging. IEEE, pp. 752–755.

Yang H, Shan C, Kolen AF, de With PH, 2019. Efficient catheter segmentation in 3d cardiac ultrasound using slice-based fcn with deep supervision and f-score loss. In: 2019 IEEE International Conference on Image Processing (ICIP). IEEE, pp. 260–264.

Yang Z, Ge W, Zhang Z, 2020. Face recognition based on mtcnn and integrated application of facenet and lbp method In: 2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM). IEEE, pp. 95–98.

Yasar H, Ceylan M, 2021. A new deep learning pipeline to detect covid-19 on chest x-ray images using local binary pattern, dual tree complex wavelet transform and convolutional neural networks. Applied Intelligence 51 (5), 2740–2763. [PubMed: 34764560]

Yin Z, Yiu V, Hu X, Tang L, 2021. End-to-end face parsing via interlinked convolutional neural networks. Cognitive Neurodynamics 15 (1), 169–179. [PubMed: 33786087]

Yu C, Wang J, Peng C, Gao C, Yu G, Sang N, 2018. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV), pp. 325–341.

Yurtkulu SC, Şahin YH, Unal G, 2019. Semantic segmentation with extended deeplabv3 architecture. In: 2019 27th Signal Processing and Communications Applications Conference (SIU). IEEE, pp. 1–4.

Zamzmi G, Chih-Yun P, Goldgof D, Kasturi R, Ashmeade T, Sun Y, 2019. A comprehensive and context-sensitive neonatal pain assessment using computer vision. IEEE Transactions on Affective Computing (01). 1–1

Zamzmi G, Hsu L-Y, Li W, Sachdev V, Antani S, 2019. Echo doppler flow classification and goodness assessment with convolutional neural networks. In: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). IEEE, pp. 1744–1749.

Zamzmi G, Hsu L-Y, Li W, Sachdev V, Antani S, 2020. Harnessing machine intelligence in automatic echocardiogram analysis: Current status, limitations, and future directions. IEEE reviews in biomedical engineering.

Zamzmi G, Rajaraman S, Antani S, 2021. Ums-rep: Unified modality-specific representation for efficient medical image analysis. Informatics in Medicine Unlocked 100571.

Zhang J, Gajjala S, Agrawal P, Tison GH, Hallock LA, Beussink-Nelson L, Fan E, Aras MA, Jordan C, Fleischmann KE, et al. , 2017. A computer vision pipeline for automated determination of cardiac structure and function and detection of disease by two-dimensional echocardiography. arXiv preprint arXiv:1706.07342.

Zhao H, Shi J, Qi X, Wang X, Jia J, 2017. Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2881–2890.

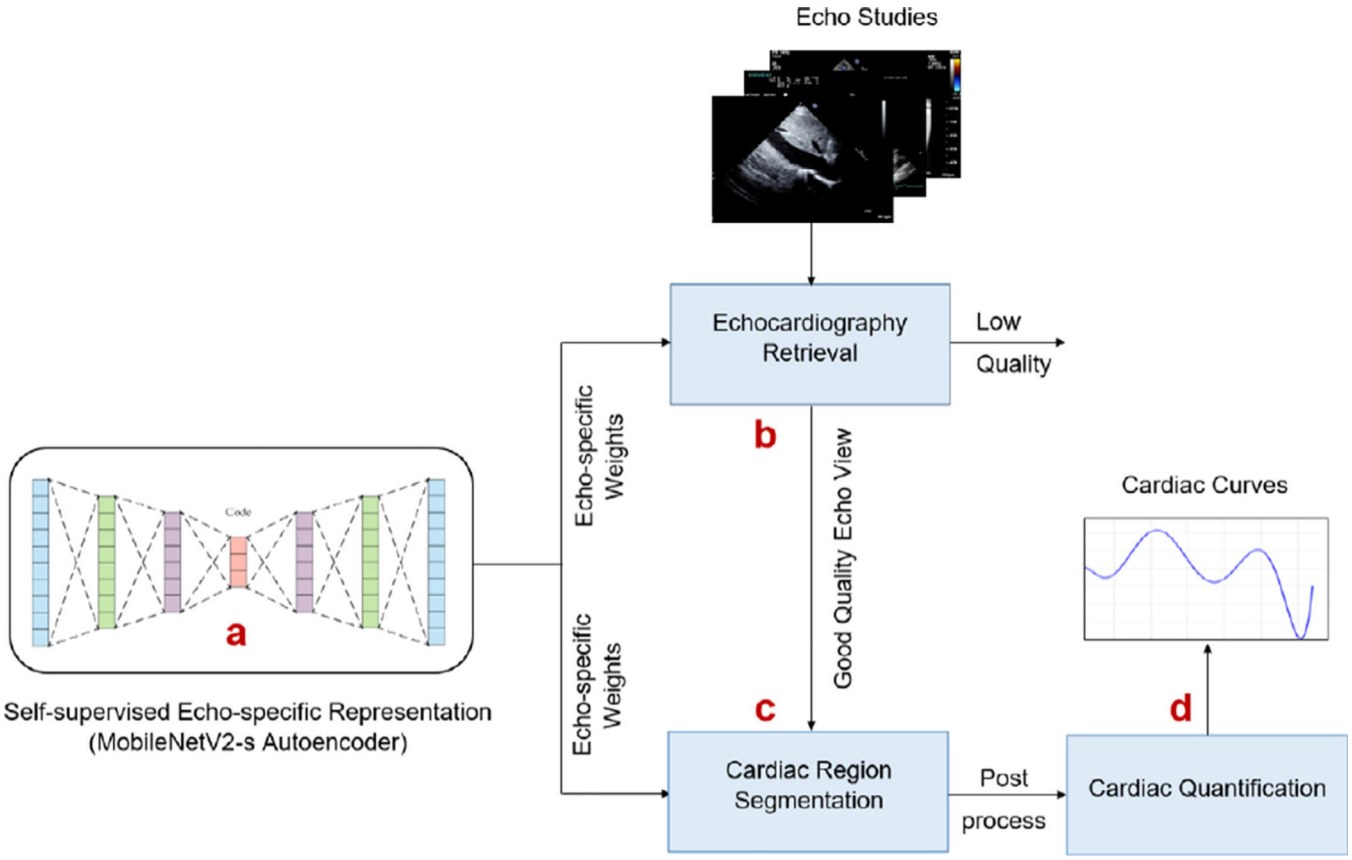

Fig. 1.

Proposed pipeline for echo analysis. (a) echo-specific representation or MobileNetV2-s based autoencoder trained to learn low-level echo features using the publicly available EchoNet-Dynamic dataset, (b) lightweight model to identify echo views and exclude low-quality cases, (c) novel segmentation model, and (d) quantification to extract biomarkers from the segmented regions. The models of both b and c are initialized with the echo-specific weights learned in a.

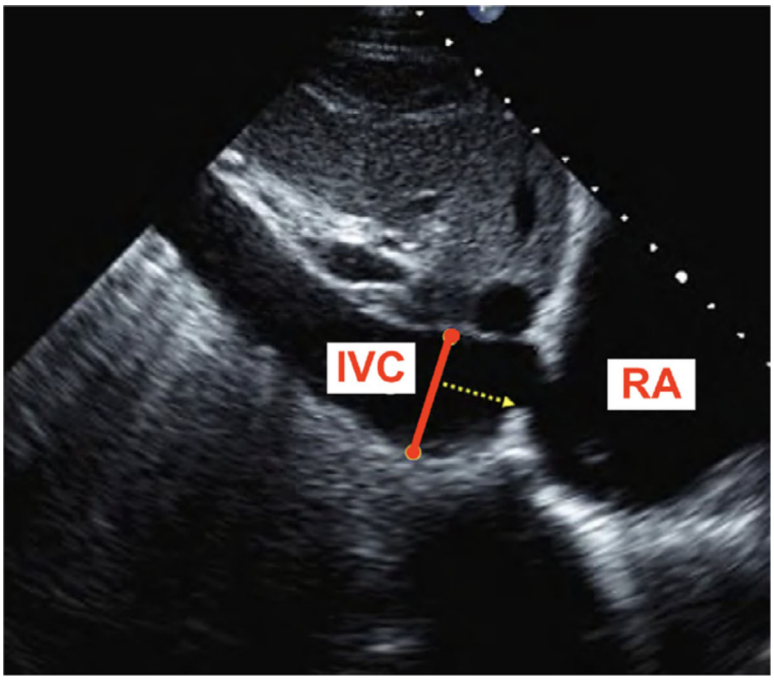

Fig. 2.
IVC subcostal view. The diameter of the IVC is measured perpendicular to IVC long axis approximately 2.0 cm from RA.

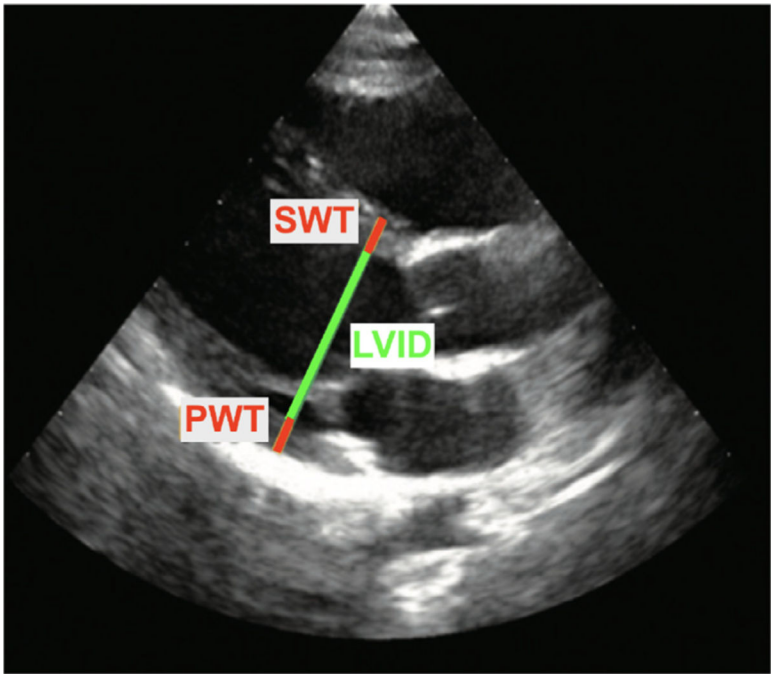

Fig. 3.

PLAX echo view. SWT and PWT are computed perpendicular to the LV long axis, at the level of mitral valve tips. LVID stands for LV internal diameter.

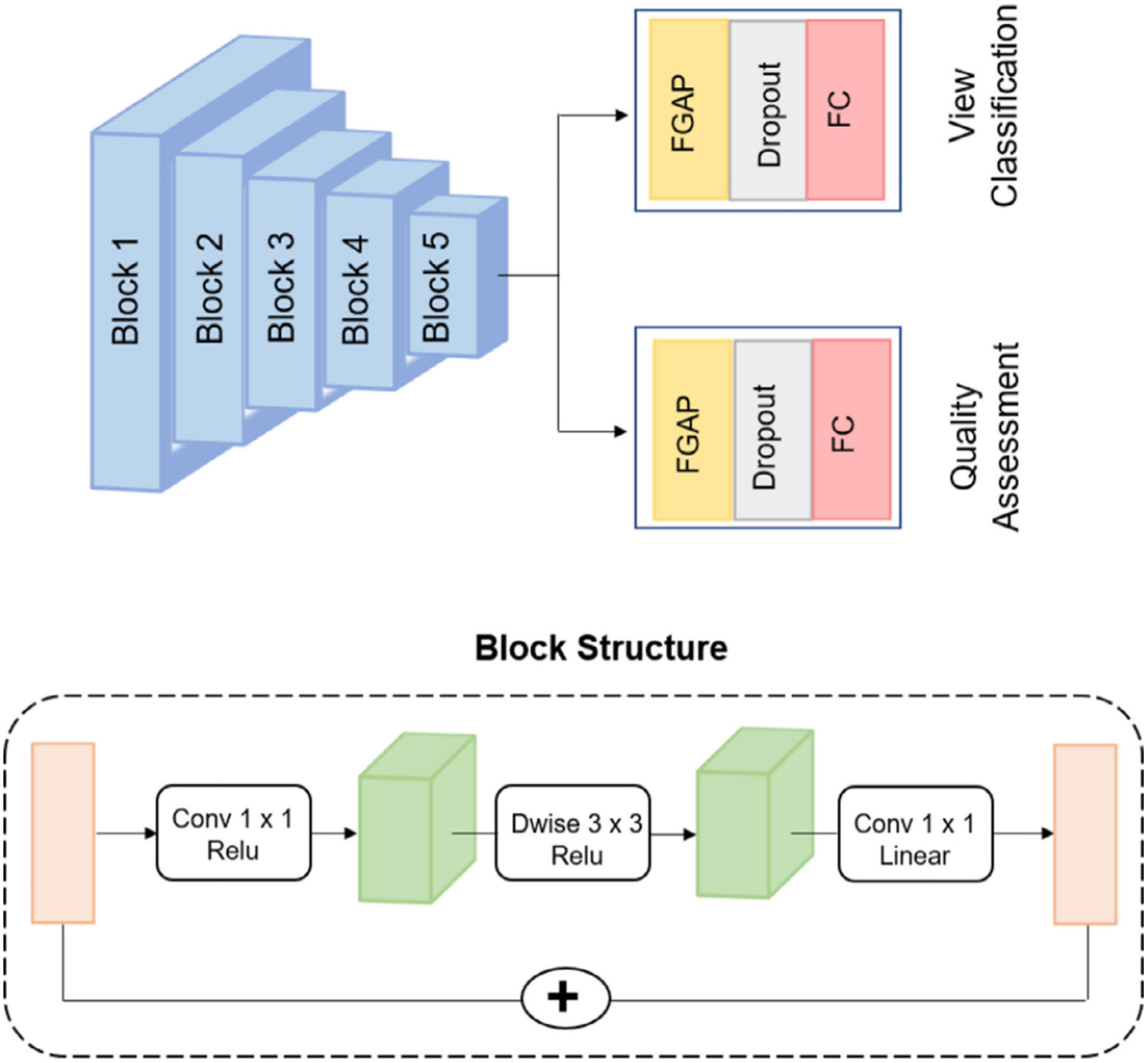

Fig. 4.

Echo retrieval. The blue boxes represent the encoder part of MobileNetV2-s-based autoencoder. The dashed box represents the structure for each block in the encoder. FGAP and FC are fuzzy global average pooling and fully connected layers, respectively.

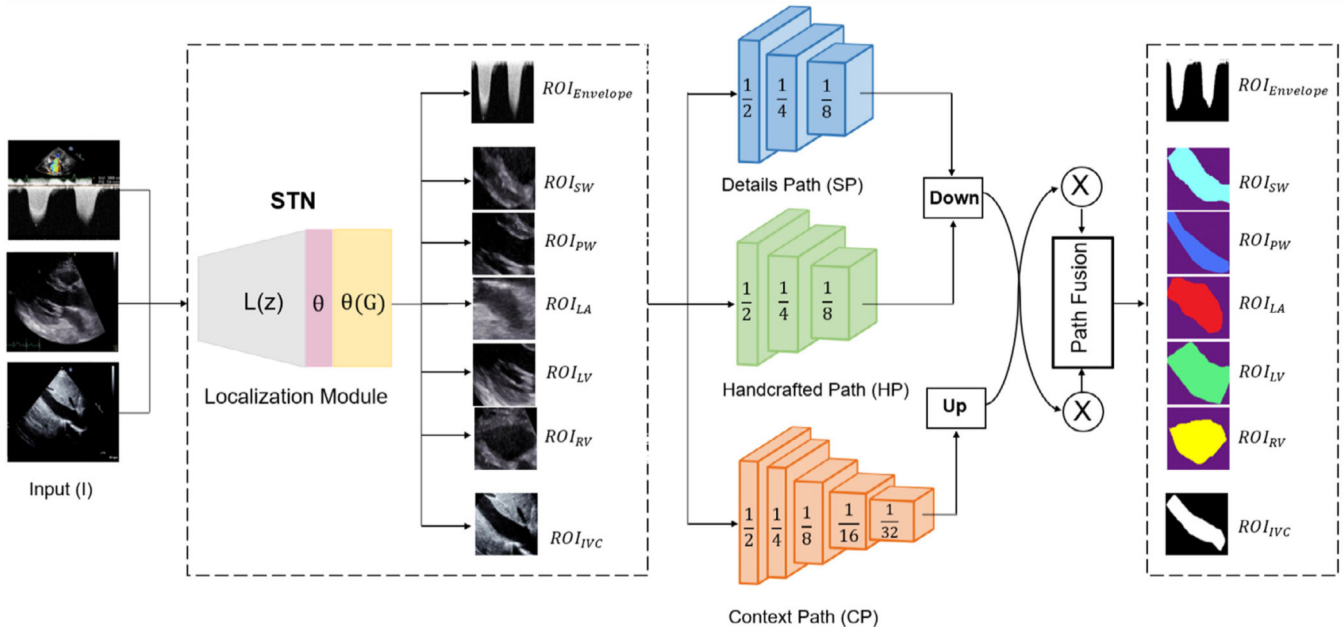

Fig. 5.

TaNet for cardiac region segmentation. TaNet has two main components: STN for ROIs localization and segmentation with 3 pathways, spatial (detail) path (*SP*), handcrafted path (*HP*), global or context path (*GP*). STN focuses the segmentation attention on different ROIs. The numbers in the segmentation cubes are the size ratios to the resolution of the input. In the fusion module, the feature maps from all pathways are aggregated as shown in Fig 6. Down, up, and ⊗ represent downsampling, upsampling, and element-wise product, respectively.

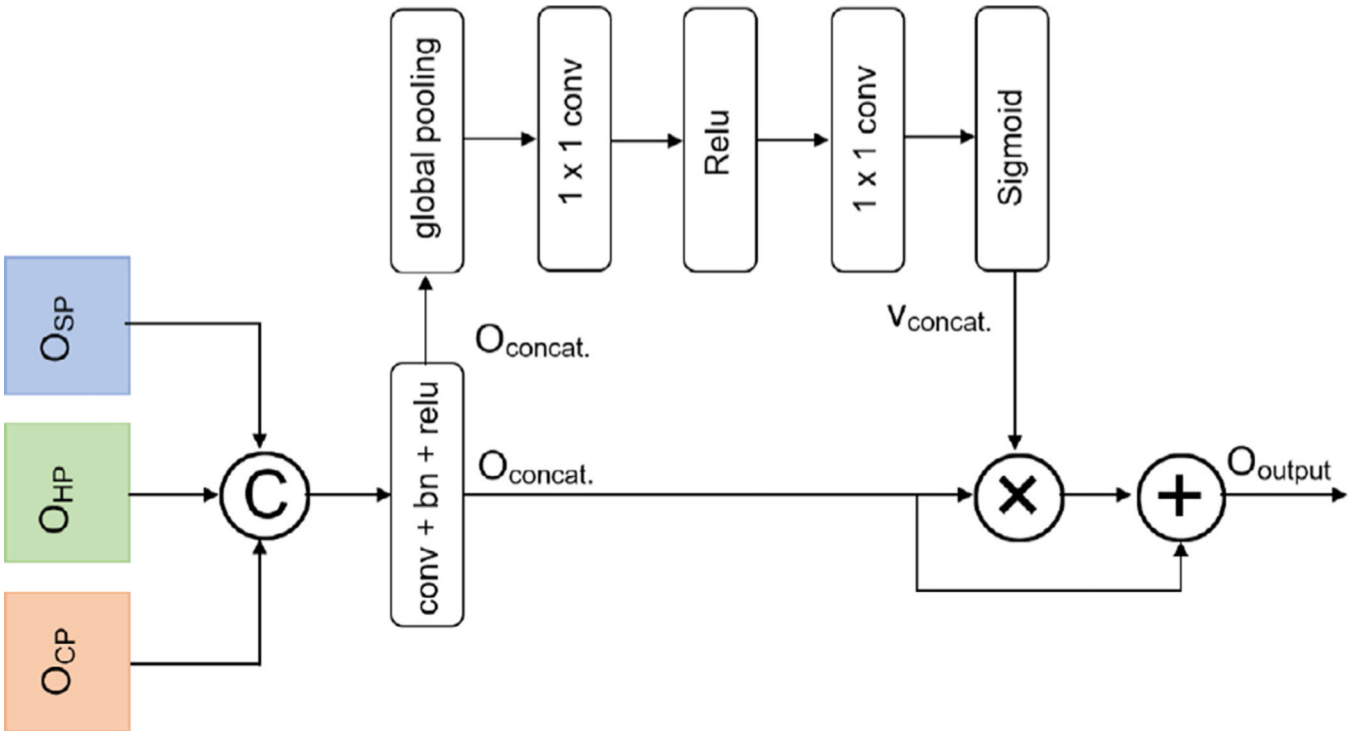

Fig. 6.

Pathways Fusion. O_{SP}, O_{HP}, O_{CP} are the output from the spatial pathway, handcrafted pathway, and context pathway, respectively. O_{concat} and v_{concat} represent the concatenation into a single vector and the weight vector, respectively. *bn*, ⊗, and + are batch normalization, element-wise product, and addition, respectively.

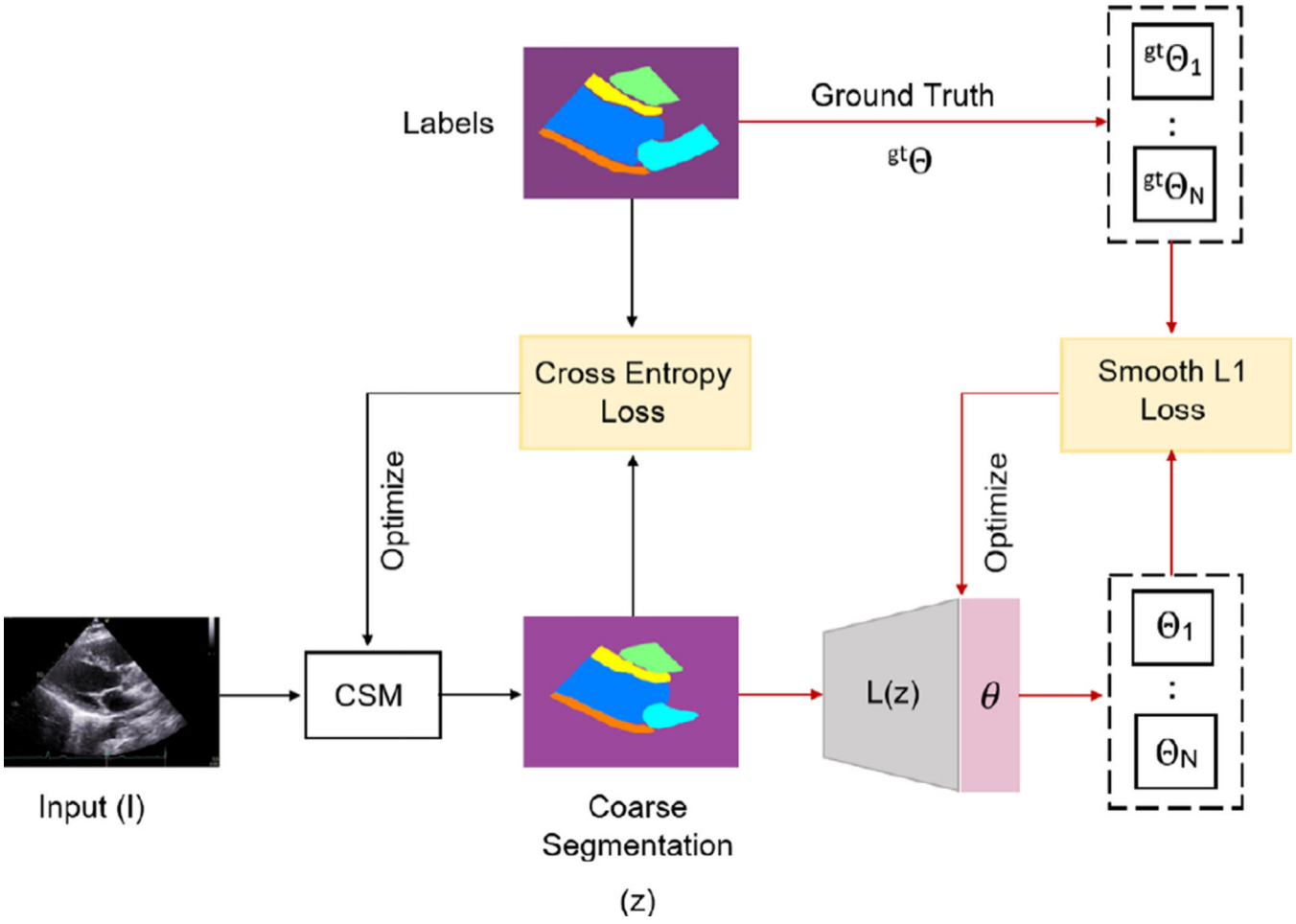

Fig. 7.

Pre-training steps. The black arrows indicate the first step of training a coarse segmentation model. The red arrows indicate the second pre-training step.

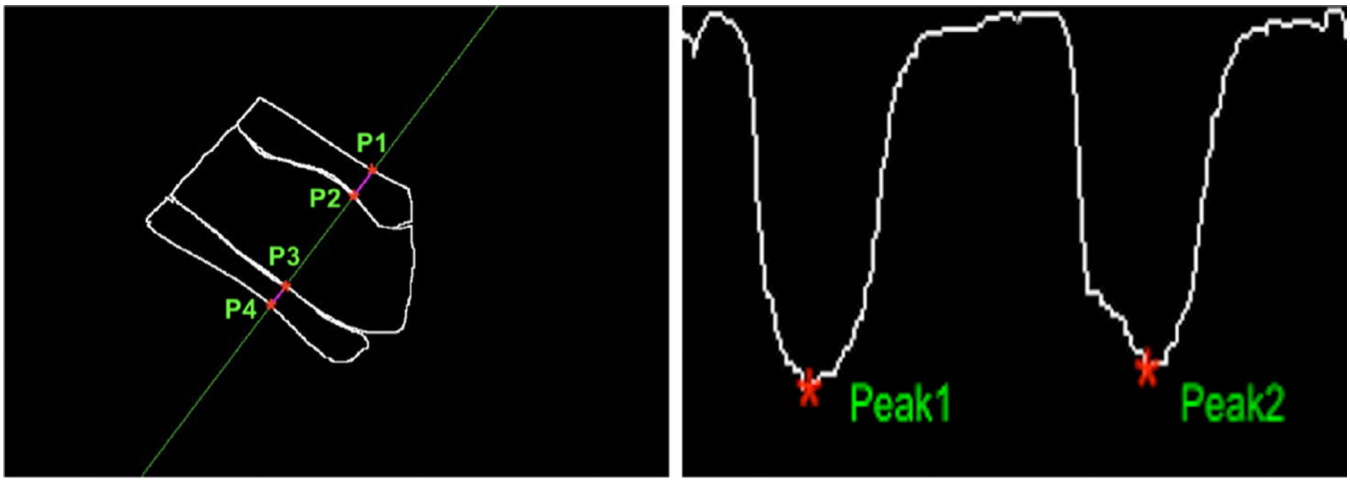

Fig. 8.

First column: estimation of LV walls thickness based on the delineated boundaries and the major axis. Second column: estimation of maximum velocities based on the peaks of the delineated boundaries.

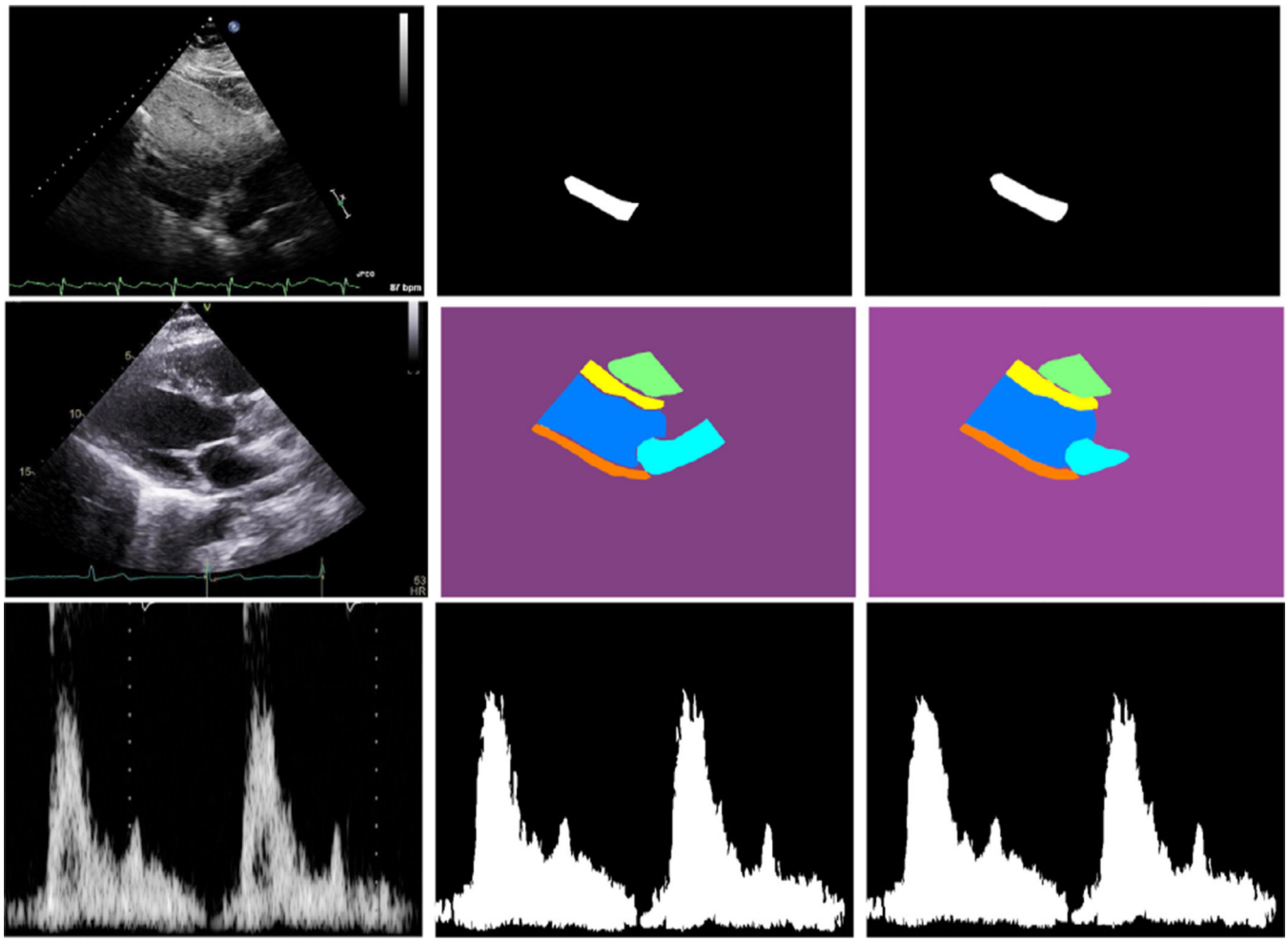

Fig. 9.

IVC (1*st* row), PLAX (2*nd* row), and Doppler (3*rd* row) segmentation: original image (1*st* column), ground truth mask (2*nd* column), and mask predicted by TaNet (3*rd* column).

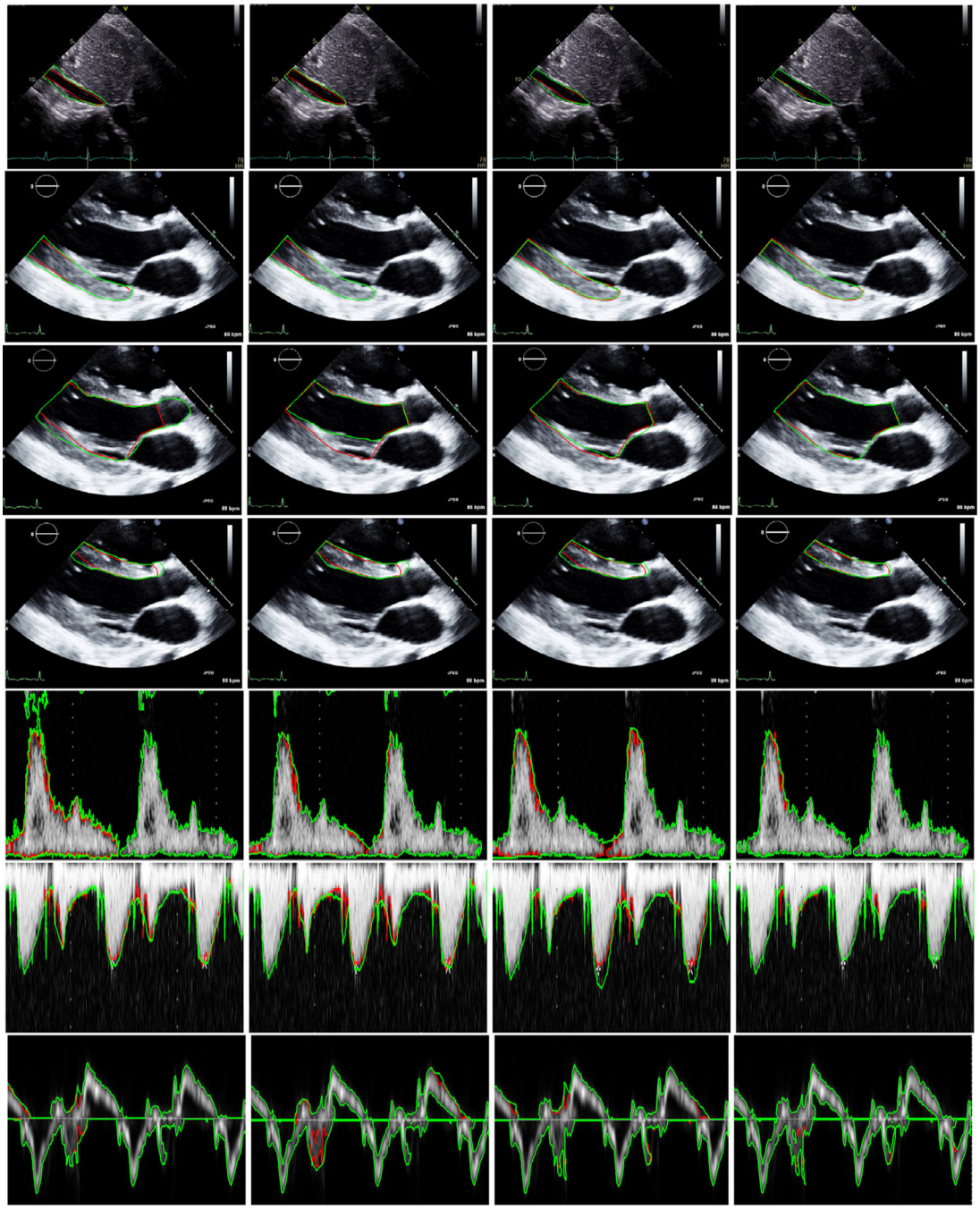

Fig. 10.

Segmentation for IVC (1*st* row), PLAX PW (2*nd* row), PLAX LV (3*rd* row), PLAX SW (4*th* row), Doppler MV (5*th* row), Doppler TR (6*th* row), and Doppler MA (7*th* row). Ground truth contour (red) and the automated contour (green) generated by FCN8 (1*st* column), UNET (2*nd* column), BiSeNet (3*rd* column), and TaNet (4*th* column).

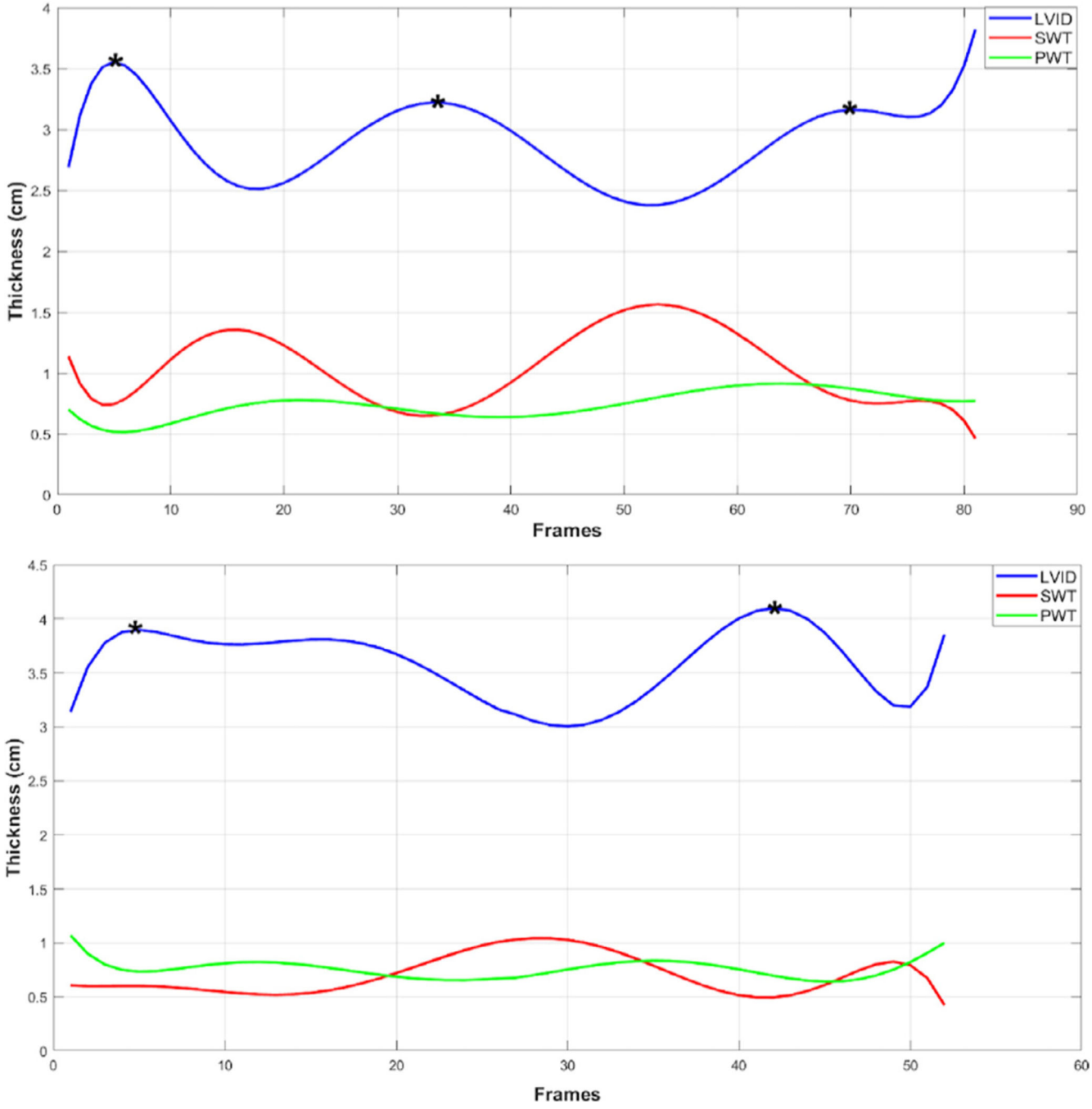

Fig. 11.

Wall thickness for IVC over frames. The curve shows the largest (peaks) and smallest (valleys) values. It can be used to find the value at a specific frame (e.g., ED) and monitor changes in the motions or patterns.

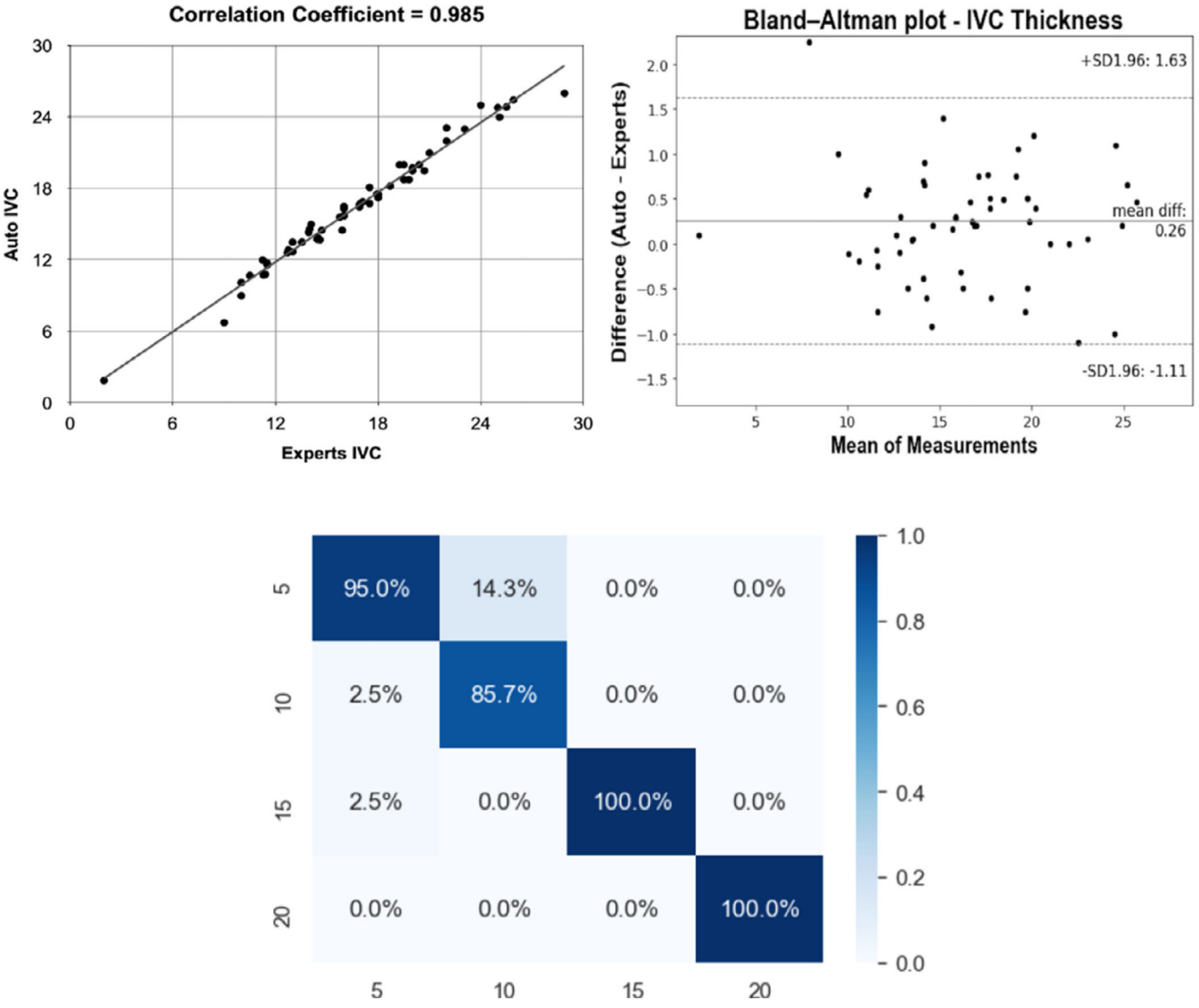

Fig. 12.

IVC dataset. First row: correlation and Bland-Altman plots for GT and automated IVC values. Second row: normalized confusion matrix for RAP; 5, 10, 15, and 20 represent different RAP scores.

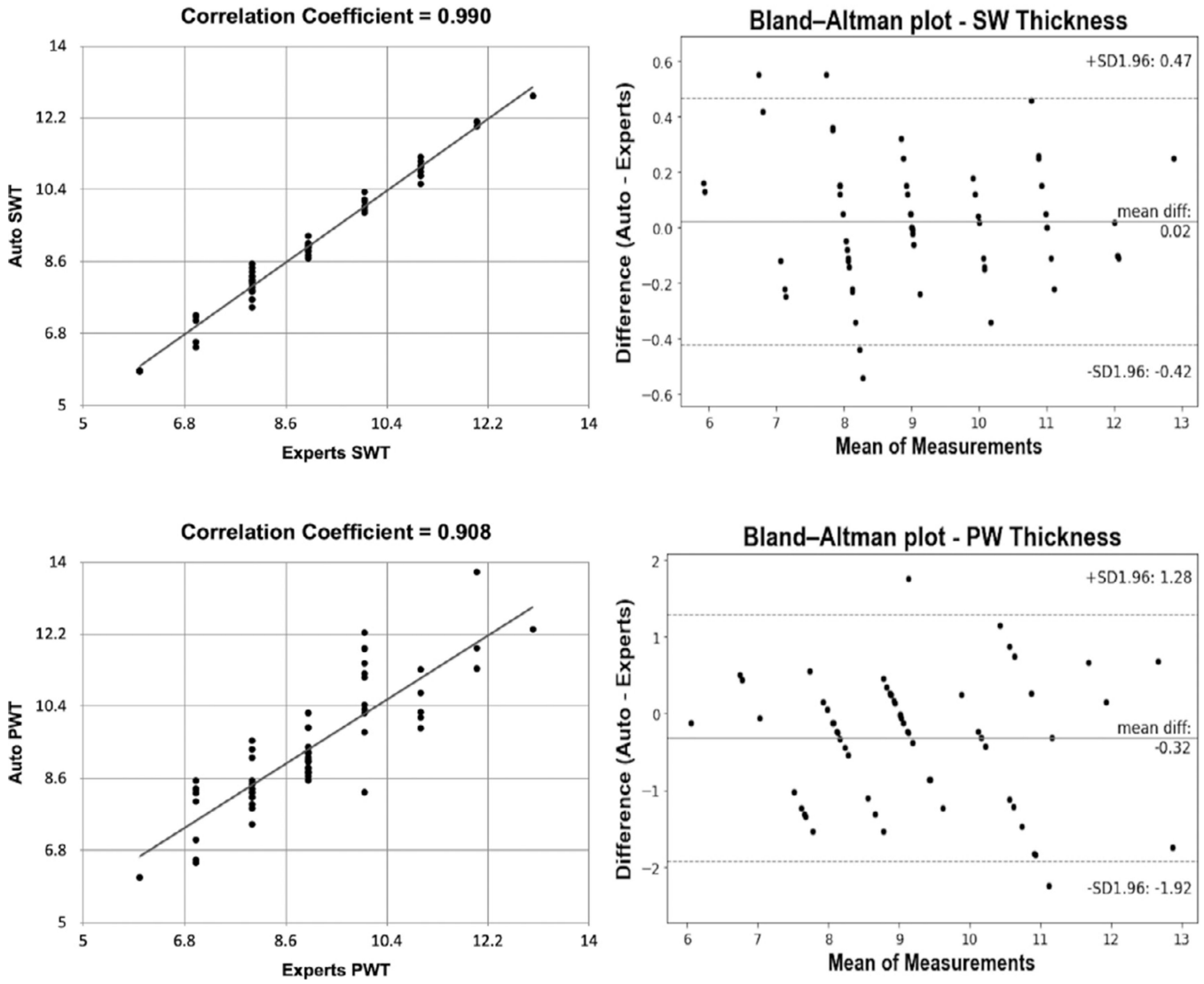

Fig. 13.

PLAX dataset. First row: correlation and Bland-Altman plots for GT and automated septal wall thickness (SWT). Second row: correlation and Bland-Altman plots for GT and automated posterior wall thickness (PWT).

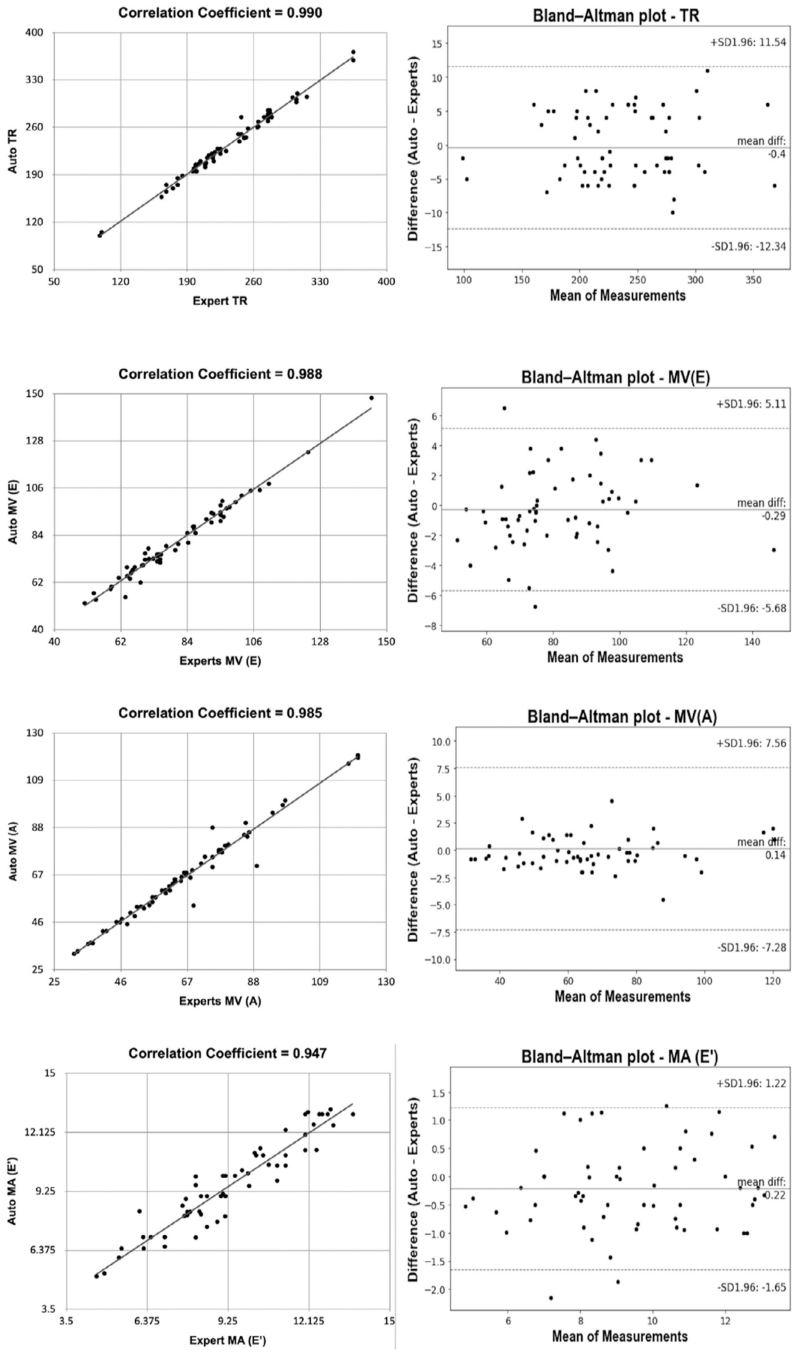

Fig. 14.

Doppler dataset. First row: correlation and Bland-Altman plots for GT and automated TR velocity values. Second row: correlation and Bland-Altman plots for GT and automated MV (E) velocity values. Third row: correlation and Bland-Altman plots for GT and automated MV (A) velocity values. Fourth row: correlation and Bland-Altman plots for GT and automated MA (E') velocity values.

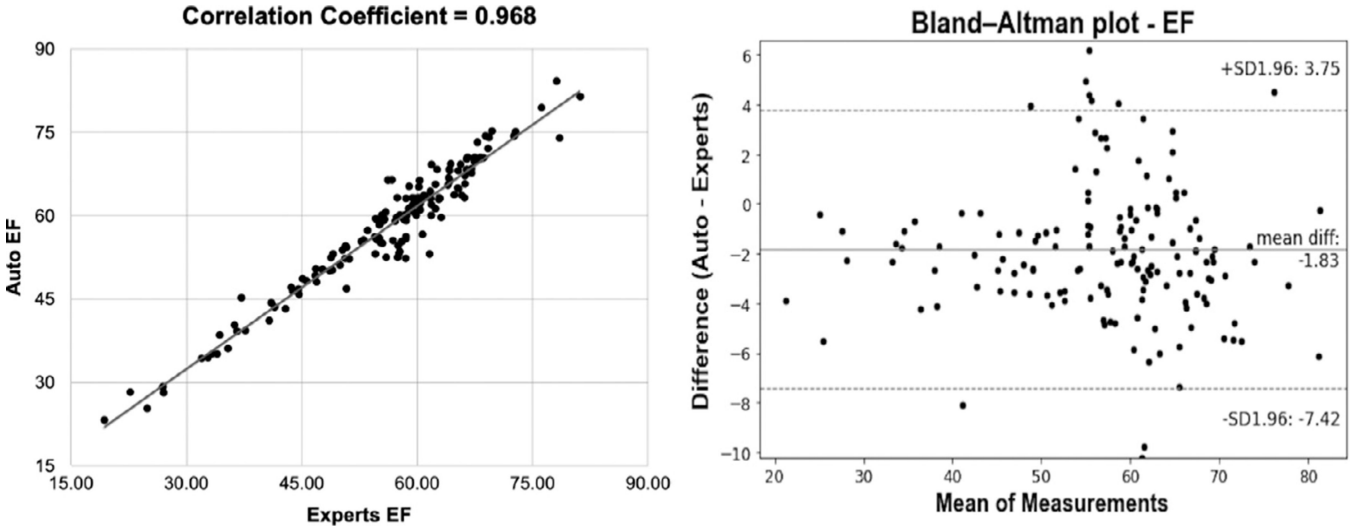

Fig. 15.

EchoNet-Dynamic dataset. Correlation and Bland-Altman plots for GT and automated EF values.

Table 1

Performance of view classification and quality assessment using conventional and fuzzy pooling operations. Acc., Prec., and Sens. stand for accuracy, precision, and sensitivity, respectively. The green cells indicate that the performance of fuzzy pooling integrated to our MobileNetV2-s based model (proposed) performs significantly ($P < 0.05$) higher than max and average pooling.

Task	**Model**	**Max Pooling**			**Average Pooling**			**Fuzzy Pooling**		
		Acc.	Prec.	Sens.	Acc.	Prec.	Sens.	Acc.	Prec.	Sens.
View Classification	Proposed	0.93	0.94	0.95	0.95	0.96	0.97	0.98	0.98	0.97
	VGG16	0.93	0.95	0.92	0.92	0.96	0.94	0.95	0.95	0.95
	ResNet18	0.90	0.90	0.94	0.91	0.95	0.91	0.94	0.94	0.94
Quality Assessment	Proposed	0.93	0.95	0.94	0.92	0.96	0.92	0.94	0.94	0.95
	VGG16	0.90	0.92	0.93	0.92	0.98	0.90	0.93	0.93	0.93
	ResNet18	0.89	0.91	0.91	0.89	0.91	0.93	0.91	0.91	0.92

Table 2

Performance (Mean±SD) of view classification head and state-of-the-art models using echo-specific, random, and ImageNet weights. Bold numerical values denote superior performance of echo-specific weights. The green cells indicate that the performance of MobileNetV2s-Based (echo-specific) is significantly ($P < 0.05$) higher than VGG16 (echo-specific) and ResNet18 (echo-specific).

Model	Weights	Accuracy	AUC	Sensitivity	Precision	F-score	MCC
MobileNetV2-s Based	Echo-specific	0.977 ±0.01	0.985 ±0.02	0.978±0.07	0.977±0.06	0.974±0.06	0.959±0.07
	Random	0.782±0.36	0.889 ±0.06	0.785±0.11	0.783±0.09	0.787±0.04	0.711±0.05
	ImageNet	0.928±0.04	0.933 ±0.14	0.925±0.09	0.927±0.07	0.926±0.10	0.903±0.03
VGG16	Echo-specific	0.953±0.02	0.958 ± 0.04	0.954±0.02	0.953±0.03	0.954±0.02	0.937±0.04
	Random	0.766±0.36	0.847 ±0.40	0.768±0.29	0.767±0.22	0.769±0.22	0.688±0.03
	ImageNet	0.940±0.22	0.949 ±0.31	0.938±0.15	0.941±0.15	0.939±0.12	0.925±0.10
ResNet18	Echo-specific	0.936±0.11	0.951 ±0.03	0.938 ±0.10	0.935±0.12	0.936 ±0.11	0.914±0.16
	Random	0.777±0.02	0.904 ±0.02	0.784±0.03	0.767±0.02	0.774±0.04	0.686±0.02
	ImageNet	0.903±0.05	0.931 ±0.10	0.908 ±0.02	0.902±0.08	0.906±0.05	0.878±0.18

Table 3

Performance (Mean±SD) of quality assessment head and state-of-the-art models using echo-specific, random, and ImageNet weights. Bold numerical values denote superior performance of echo-specific weights. The green cells indicate that the performance of MobileNetV2s- Based (echo-specific) is significantly ($P < 0.05$) higher than VGG16 (echo-specific) and ResNet18 (echo-specific)

Model	Weights	Accuracy	AUC	Sensitivity	Precision	F-score	MCC
MobileNetV2-s Based	Echo-specific	0.943±0.10	0.962 ±0.22	0.946±0.09	0.944±0.04	0.945± 0.06	0.891±0.01
	Random	0.868±0.13	0.894 ±0.06	0.869±0.11	0.868±0.09	0.869±0.04	0.735±0.05
	ImageNet	0.928±0.04	0.936 ±0.14	0.925±0.09	0.927±0.07	0.926±0.10	0.854±0.03
VGG16	Echo-specific	0.933±0.15	0.951 ±0.43	0.933±0.34	0.932±0.29	0.933±0.30	0.865±0.03
	Random	0.873±0.36	0.899 ±0.31	0.872±0.29	0.876±0.22	0.874±0.18	0.745±0.03
	ImageNet	0.920±0.22	0.943 ±0.31	0.920±0.15	0.921±0.10	0.921±0.12	0.843±0.10
ResNet18	Echo-specific	0.913±0.02	0.933 ±0.05	0.914 ±0.03	0.912±0.02	0.912 ±0.05	0.834±0.11
	Random	0.835±0.12	0.854 ±0.10	0.837±0.08	0.835±0.05	0.836±0.08	0.676±0.09
	ImageNet	0.903±0.14	0.924 ±0.09	0.905±0.10	0.902±0.12	0.904±0.10	0.808±0.18

Table 4

Comparison of computational complexity.

Model	Param.	≈ Size	≈ Training
MobileNetV2-s Based	55,620	1.2 MB	12.45
VGG16	138 MM	528 MB	110.64
ResNet18	11 MM	44 MB	45.20

Table 5

Results of ablation experiments to evaluate the impact of STN and LBP-encoded convolutional kernels on IVC segmentation. The results are reported on the testing set of IVC dataset. The green cells mean that TaNet performance is significantly ($p < 0.05$) higher than the baseline FCN8, UNET, and BiSeNet models.

Model		**IVC**		**FPS**
		IoU	F1	
	Baseline	0.83±0.01	0.91±0.07	9
FCN8	w/ STN	0.86±0.17	0.93±0.22	7
	w/ LBP	0.84±0.25	0.92±0.16	12
	Baseline	0.86±0.12	0.93±0.05	10
UNET	w/ STN	0.88±0.08	0.93±0.15	7
	w/ LBP	0.87±0.09	0.93±0.12	13
	Baseline	0.92±0.11	0.95±0.17	105
BiSeNet	w/ STN	0.94±0.05	0.97±0.07	98
	w/ LBP	0.94±0.09	0.96±0.11	124
TaNet	STN & LBP	0.96±0.03	0.98±0.05	94

Table 6

Results of ablation experiments to evaluate the impact of STN and LBP-encoded kernels on PLAX regions segmentation (LV, RV, LA, SW, and PW). The results are reported on the testing set of PLAX dataset. The green cells mean that TaNet performance is significantly ($p < 0.05$) higher than the baseline FCN8, UNET, and BiSeNet models.

Model		**LV**		**RV**		**LA**		**SW**		**PW**		**FPS**
		IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	
	Baseline	0.87±0.03	0.93±0.05	0.82±0.09	0.91±0.19	0.83±0.20	0.91±0.17	0.86±0.13	0.93±0.03	0.76±0.12	0.87±0.16	6
FCN8	w/STN	0.93±0.09	0.96±0.08	0.80±0.13	0.89±0.06	0.89±0.16	0.94±0.08	0.86±0.10	0.93±0.02	0.77±0.14	0.87±0.03	4
	w/ LBP	0.89±0.14	0.93±0.20	0.80±0.04	0.89±0.10	0.85±0.22	0.92±0.13	0.89±0.07	0.94±0.09	0.83±0.16	0.91±0.18	9
	Baseline	0.85±0.18	0.91±0.08	0.81±0.13	0.89±0.02	0.88±0.13	0.93±0.11	0.87±0.04	0.93±0.03	0.79±0.02	0.88±0.01	5
UNET	w/STN	0.90±0.03	0.95±0.05	0.86±0.09	0.92±0.10	0.91±0.17	0.95±0.13	0.90±0.05	0.94±0.03	0.81±0.19	0.89±0.16	3
	w/ LBP	0.87±0.02	0.93±0.07	0.88±0.11	0.94±0.13	0.87±0.20	0.93±0.09	0.89±0.15	0.94±0.15	0.82± 0.01	0.90±0.13	7
	Baseline	0.89±0.01	0.94±0.09	0.85±0.03	0.92±0.01	0.89±0.05	0.94±0.02	0.85±0.10	0.92±0.16	0.77±0.23	0.87±0.20	100
BiSeNet	w/STN	0.94±0.11	0.97±0.05	0.90±0.12	0.94±0.09	0.92±0.17	0.96±0.10	0.90±0.04	0.95±0.01	0.78±0.15	0.87±0.19	92
	w/ LBP	0.91±0.09	0.95±0.11	0.86±0.22	0.92±0.17	0.93±0.08	0.96±0.02	0.90±0.15	0.94±0.10	0.86±0.24	0.92±0.21	108
TaNet	STN & LBP	0.95±0.01	0.98±0.03	0.91±0.05	0.95±0.02	0.93±0.04	0.97±0.06	0.93±0.09	0.96±0.01	0.88±0.08	0.93±0.05	85

Table 7

Results of ablation experiments to evaluate the impact of STN and LBP-encoded kernels on Doppler segmentation. The results are reported on the testing set of Doppler (TR, MV, MA) dataset. The green cells in the last row mean that TaNet performance is significantly ($p < 0.05$) higher than the baseline FCN8, UNET, and BiSeNet models.

Model		**Doppler TR**		**Doppler MV**		**Doppler MA**		**FPS**
		IoU	F1	IoU	F1	IoU	F1	
	Baseline	0.90±0.27	0.94±0.30	0.86±0.11	0.92 ±0.14	0.88±0.16	0.94±0.10	7
FCN8	w/ STN	0.93±0.16	0.96±0.19	0.89±0.07	0.94 ±0.05	0.90±0.11	0.95±0.09	6
	w/ LBP	0.91 ±0.21	0.95±0.17	0.87±0.09	0.93 ±0.08	0.89±0.11	0.94±0.14	10
	Baseline	0.88±0.16	0.93±0.10	0.89±0.12	0.94 ±0.14	0.91±0.10	0.95±0.09	9
UNET	w/ STN	0.93±0.09	0.96±0.07	0.90±0.04	0.96 ±0.06	0.94±0.07	0.97±0.09	6
	w/ LBP	0.90±0.10	0.95±0.08	0.90±0.11	0.94 ±0.10	0.90±0.04	0.95±0.06	11
	Baseline	0.91±0.13	0.95±0.16	0.87±0.09	0.93 ±0.11	0.93±0.04	0.96±0.07	113
BiSeNet	w/ STN	0.93±0.09	0.96±0.14	0.89±0.07	0.94 ±0.05	0.95±0.07	0.97±0.05	99
	w/ LBP	0.93±0.05	0.97±0.07	0.89±0.10	0.95 ±0.13	0.95±0.08	0.97±0.10	124
TaNet	STN & LBP	0.96±0.04	0.97±0.02	0.93±0.05	0.96 ±0.09	0.97±0.05	0.98±0.02	96

Table 8

Summary of the state-of-the-arts classification models. Inference time is reported on NVIDIA GTX1080Ti GPU.

Model	Task	AUC	F1	Size	Time
MobileNetV2-s	View	0.985	0.974	≈ 1.2 MB	≈2.11 ms
(our)	Quality	0.962	0.945		≈1.45 ms
VGG-16	View	0.958	0.954	≈ 528 MB	≈5.58 ms
	Quality	0.951	0.933		≈4.25 ms
ResNet18	View	0.951	0.936	≈ 44 MB	≈7.45 ms
	Quality	0.933	0.912		≈5.90 ms
DenseNet161	View	0.963	0.948	≈ 110 MB	≈10.62 ms
	Quality	0.947	0.925		≈6.75 ms

Table 9

Summary of the state-of-the-arts segmentation models. FPS range is reported on NVIDIA GTX1080Ti GPU.

Model	$mIoU_{IVC}$	$mIoU_{PLAX}$	$mIoU_{Doppler}$	FPS
TaNet	0.964	0.927	0.943	≈ 85–96
BiSeNet	0.923	0.854	0.906	≈ 100–113
Res-U	0.866	0.836	0.857	≈ 2–9
UNET	0.884	0.854	0.896	≈ 5–10
FCN	0.833	0.831	0.883	≈ 6–9