

Automatic Quality Assessment of Echocardiograms Using Convolutional Neural Networks: Feasibility on the Apical Four-Chamber View

Amir H. Abdi, *Student Member, IEEE*, Christina Luong, Teresa Tsang, Gregory Allan, Saman Nouranian, John Jue, Dale Hawley, Sarah Fleming, Ken Gin, Jody Swift, Robert Rohling, *Senior Member, IEEE*, and Purang Abolmaesumi,* *Senior Member, IEEE*

Abstract—Echocardiography (echo) is a skilled technical procedure that depends on the experience of the operator. The aim of this paper is to reduce user variability in data acquisition by automatically computing a score of echo quality for operator feedback. To do this, a deep convolutional neural network model, trained on a large set of samples, was developed for scoring apical four-chamber (A4C) echo. In this paper, 6,916 end-systolic echo images were manually studied by an expert cardiologist and were assigned a score between 0 (not acceptable) and 5 (excellent). The images were divided into two independent training-validation and test sets. The network architecture and its parameters were based on the stochastic approach of the particle swarm optimization on the training-validation data. The mean absolute error between the scores from the ultimately trained model and the expert's manual scores was 0.71 ± 0.58 . The reported error was comparable to the measured intra-rater reliability. The learned features of the network were visually interpretable and could be mapped to the anatomy of the heart in the A4C echo, giving confidence in the training result. The computation time for the proposed network architecture, running on a graphics processing unit, was less than 10 ms per frame, sufficient for real-time deployment. The proposed approach has the potential to facilitate the widespread use of echo at the point-of-care and enable early and timely diagnosis and treatment. Finally, the approach did not use any specific assumptions about the A4C echo, so it could be generalizable to other standard echo views.

Index Terms—Convolutional neural network, deep learning, quality assessment, echocardiography, apical four-chamber, swarm optimization.

I. INTRODUCTION

HEART disease is a leading cause of morbidity and premature death worldwide. 2D echocardiography (echo) is a non-invasive, low-cost, portable, and accessible imaging technology that allows diagnosis of various cardiac conditions, risk stratification, and prognostication with minimal risk. Echo provides an excellent assessment of structure and function of the heart. Standard echo studies include determination of both contractility and relaxation properties of the ventricles, atrial size and function, as well as valvular structure and function [1].

Echocardiography can be performed with several different techniques, among which transthoracic echocardiography (TTE) is the most common. In TTE, images are obtained from different probe positions, which can be grouped into four main categories, *i.e.* parasternal, apical, subcostal and suprasternal. One of the most informative yet challenging views to obtain is the apical four-chamber (A4C) view. This view contains cross-longitudinal sections of both ventricles and atria through the tricuspid and mitral valves; the cardiac apex is visualized closest to the transducer, the ventricles are in the near field and the atria are in the far field [2]. Fig. 1 displays a typical A4C view of the heart at its end-systolic state, which depicts the left and right atria, left and right ventricles, tricuspid valve and mitral valve. If the probe is oriented appropriately, the interatrial and interventricular septa are aligned vertically in the center of the image and the bullet-shaped apex is at the center-top. The A4C view is mainly used for quantification of cardiac chambers, evaluation of their contractility, and left ventricular ejection fraction [3], [4]. To best acquire the A4C view, the phased-array transducer is directed towards the cardiac apex through the apical impulse between the ribs and is aligned with the long axis of the left ventricle [5].

The accuracy of estimations of chamber volumes, function and ejection fraction in 2D echo views, such as the A4C view,

Manuscript received October 28, 2016; revised March 17, 2017; accepted March 29, 2017. Date of publication April 4, 2017; date of current version June 1, 2017. This work was supported in part by the Natural Sciences and Engineering Research Council and in part by the Canadian Institutes of Health Research. Asterisk indicates corresponding author.

A. H. Abdi, G. Alan, and S. Nouranian are with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada.

C. Luong, T. Tsang, J. Jue, D. Hawley, S. Fleming, K. Gin, and J. Swift are with Vancouver General Hospital's Cardiology Laboratory, Vancouver, BC V5Z 1M9, Canada.

R. Rohling is with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada, and also with the Department of Mechanical Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada.

*P. Abolmaesumi is with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: purang@ece.ubc.ca).

Digital Object Identifier 10.1109/TMI.2017.2690836

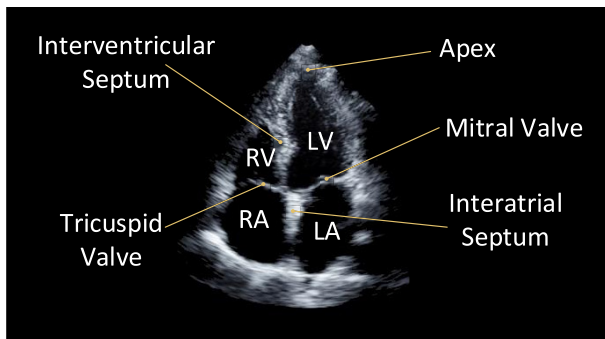


Fig. 1. A typical apical four-chamber echocardiogram depicting the four chambers, mitral valve, tricuspid valve, septum, and the lateral walls.

depend on the quality of the acquired cine. Therefore, they can be affected by various factors [6], [7], such as image position, exactness of geometric assumptions, and accuracy of boundary tracings [8]. Consequently, if suboptimal images are obtained, all measurements can be affected, resulting in a misclassification of the patient in terms of needs for specific treatments. Moreover, unlike other medical image modalities that benefit from a relatively automated image acquisition routine, the quality of echo highly relies on the experience and objectivity of the sonographer to obtain the most appropriate intersection of the imaging plane with the cardiac anatomy [9]. While it is expected from experienced technicians to find the correct acoustic window leading to high-quality images, it is common for less-experienced users to acquire data with suboptimal image quality.

To assist the sonographer in acquiring optimal views, several research groups have made notable efforts in producing real-time feedback to the operator regarding image quality. A set of studies have attempted to detect shadows and aperture blockage in echo images [10], [11]. These techniques recognize changes in the power spectrum or the coherency of the signal as an indicator of the aperture blockage. While, in general, these methods are applicable to all types of ultrasound imaging, they are limited to detection of aperture obstruction, which is only one of the many factors that affect the quality of an echo study.

Several groups have proposed content-based cardiac inter-view classification techniques using machine learning and statistical approaches [12], [13] as well as low-level features [14], [15]. However, intra-view quality analysis of echo is a much more challenging problem, as there is relatively higher correlation between the visual content of the different echo images that need scoring.

An important factor in determining echo quality scores is the presence and sharpness of desired anatomical structures for that particular echo view. Some studies have defined a goodness-of-fit based on mapping a parametric template model on the image [16] or searching for a binary atlas using a generalized Hough transform algorithm [17]. These intensity-based approaches are sensitive to noise and low contrast, and tend to fail on weak edges. Moreover, they only respond to the presence of low-level features, and have no mechanism implemented to detect higher-level structures. As a result,

they are expected to fail in realizing echo acquisition defects such as foreshortening and missing apex. Efforts have also been invested in assisting the operator with setting the optimal acquisition parameters of ultrasound imaging through manifold learning [18]; however, that work did not consider the effect of variability in ultrasound viewing angle and presence of desired anatomical structures in image.

Deep convolutional neural networks (DCNNs) [19] are deep learning architectures that possess the ability to extract hierarchical discriminative features, while preserving the locality of pixel dependencies. These models learn representations of data with multiple levels of abstraction, while the features are learned as part of an end-to-end supervised training, as opposed to hand-crafted alternatives [20]. DCNNs have recently gained significant attention and are now being considered as the state-of-the-art method for many challenging medical and non-medical tasks [21], [22]. As a result of DCNN's local connectivity and shared filter architecture, there are much fewer weights to adjust in comparison to the traditional neural network architectures; consequently, they are less prone to over-fit and are easier to train. With the advent of Graphics Processing Unit (GPU)-accelerated computation, it is now practical to train DCNNs with multiple layers for regression and classification purposes.

In our previous work [23], we investigated the feasibility of using convolutional neural networks to assess the quality of echo data. Here, we expand on that work and propose a framework for optimizing the deep learning architecture to generate an automatic echo score (AES) in real time. Our framework incorporates a regression model, based on hierarchical features extracted automatically from echo images, which relates images to a quality score determined by an expert cardiologist. We demonstrate the feasibility of our approach on the A4C echo view. In this study, data acquired from 6,916 patient studies were used to design, optimize, train and test the model. Using GPU-computing, the ultimate trained network is able to assess the quality of an echo image in real time. Since the design of the proposed DCNN architecture does not include any a priori assumptions on the A4C view, this approach could be extensible to other standard echo views.

II. METHODS

A. Convolutional Neural Networks

In this research, a regression model was developed to quantify the quality of echo images. The loss function of this model, trained via the stochastic gradient descent algorithm, was the ℓ_2 norm of the error, i.e. the Euclidean distance of the network's output to the manual echo score (MES) assigned by the expert (Section III-A).

The regression model was designed on a deep neural network architecture, structured in two main stages: a convolutional stage and a fully-connected stage. The first stage is composed of convolutional layers (conv) and pooling layers (pool); the second stage only contains fully-connected layers (fc).

1) *Convolutional Stage*: The conv layer, introduced by Cun *et al.* [24] and LeCun *et al.* [25], is the primary component

of this stage. It consists of 2D kernels that are convolved with the input signal resulting in the output feature-maps. Mathematically, kernels calculate the locally weighted sum of inputs or, as the name implies, perform a discrete convolution, as follows:

$$a_{i,jk}^l = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} w_{i,mn}^l a_{(j+m)(k+n)}^{l-1}. \quad (1)$$

Here, w_i^l is the weight matrix, and a_i^l is the output feature-map of the i th kernel of the `conv` layer l ; while a^{l-1} represents the input feature-map of the layer. Each kernel is convolved with all the feature-maps of the previous layer and generates a 2D output. Outputs of all kernels of a given layer are stacked together to create the 3D output feature-map of the `conv` layer.

The total number of parameters in the `conv` layer is equal to the number of kernels multiplied by the size of each kernel. Although it is theoretically acceptable to have kernels of different sizes in the same layer, this is rarely the case in practice [20], [21]. Since a single kernel is responsible for generating an output feature-map, the `conv` layer contains substantially fewer weights to adjust compared to other layers such as the `fc` layer. Each convolutional kernel models the spatial correlation of its input with subsequent layers representing an increased level of abstraction.

The output of the `conv` layer is passed into a non-linear activation function. We use the Rectified Linear Unit (ReLU), which resembles the function of biological neurons [26], because it converges faster than its alternatives such as sigmoid and hyperbolic tangent functions, while achieving the same performance [27]. ReLU is defined as

$$f_{\text{ReLU}}(a_i) = \max(0, a_i), \quad (2)$$

and is reported to demonstrate equal or higher performance compared to other activation functions of this family, *i.e.* PReLU and LReLU [28].

The `pool` layers reduce the spatial variance of feature-maps, and allow faster convergence and selection of superior invariant features [29]. They also reduce the computational cost of deeper layers by reducing the size of the feature-maps. Moreover, the `pool` layers encourage generalization by making the model invariant to small translations. Among pooling methods, max-pooling has demonstrated higher performance compared to its rivals [30] as it removes non-maximal values using max-filters. There is no weight associated with the max-pooling layer.

2) Fully-Connected Stage: Each neuron of an `fc` layer is fully connected to every activation in its previous layer. Mathematically, an `fc` layer can be represented with a matrix multiplication, followed by an offset summation, as follows:

$$f_{\text{FC}_i}^l(a^{l-1}) = \sum_{j=1}^n w_{ij}^l a_j^{l-1} + b_i^l, \quad (3)$$

where w_{ij}^l represents the j th weight of neuron i in layer l , and b_i^l is its bias value.

Similar to `conv` layers, the output of an `fc` layer is also passed into an activation function. The output of the last

`fc` layer is not filtered by an activation function as it produces the network's final output.

B. Hyper-Parameter Optimization

Deep convolutional neural networks, like most learning algorithms, rely on a set of hyper-parameters that can affect performance of a trained model. In practice, hyper-parameters are chosen in order to minimize the generalization error [31]. This task is substantially based on running trials with different hyper-parameter settings, comparing the results, and choosing the best setting. Many methods have been suggested on how to choose the trials, from grid searches and random searches [32], to sequential and tree-based searches [32]. Also, Garro and Vázquez used Particle Swarm Optimization (PSO) to tune feed-forward neural networks in terms of the activation function, number of neurons in each layer, and number of connections inside each layer [33].

PSO is an iterative stochastic approach based on swarm intelligence, inspired by the movements of bird flocks [34]. PSO is an iterative solution initiated by a set of particles in a n -dimensional space, where n is the number of hyper-parameters to optimize. In each PSO generation (iteration), the position of each particle, p_i , is updated with respect to its newly updated velocity, v_i^{t+1} , as follows:

$$p_i^{t+1} = p_i^t + v_i^{t+1}. \quad (4)$$

Prior to the position update, the velocity of the particle is updated based on the best solution found by the particle (S_i) and the best solution found by the whole population (S_g), as follows:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (S_i^t - p_i^t) + c_2 r_2 (S_g^t - p_i^t), \quad (5)$$

where ω , r_1 and r_2 are random variables taken from a uniform distribution on $[0,1]$ for each generation. ω is the inertia weight term, while c_1 and c_2 are acceleration variables. These three parameters determine the impact of momentum, local best (S_i), and the global best (S_g) solutions on the velocity. In our experiment, c_1 and c_2 were set to 2.05 based on literature [35].

III. EXPERIMENTAL SETUP

A. Dataset

Training computational models to represent medical images requires a large annotated dataset. In this research, we had the advantage of an echo database available on the Picture Archiving and Communication System of the Vancouver General Hospital (VGH). These echo images were acquired mostly by echo-technicians, with a small contribution from cardiology trainees and trainee technicians, during routine cardiac exams.

In an echo acquisition, the heart is imaged from at least seven standard (parasternal long and short axes, apical 2-, 3-, and 4-chamber, subcostal, and suprasternal) and atypical imaging views for which the sonographer places a transducer on the patients chest to obtain ultrasound frame stacks (cine clips) in a specific order from each of the standard echo views. The operator is instructed to acquire data from as many views as possible to visualize the pathology. In this research, we have focused on the apical four-chamber (A4C) view (Fig. 1).

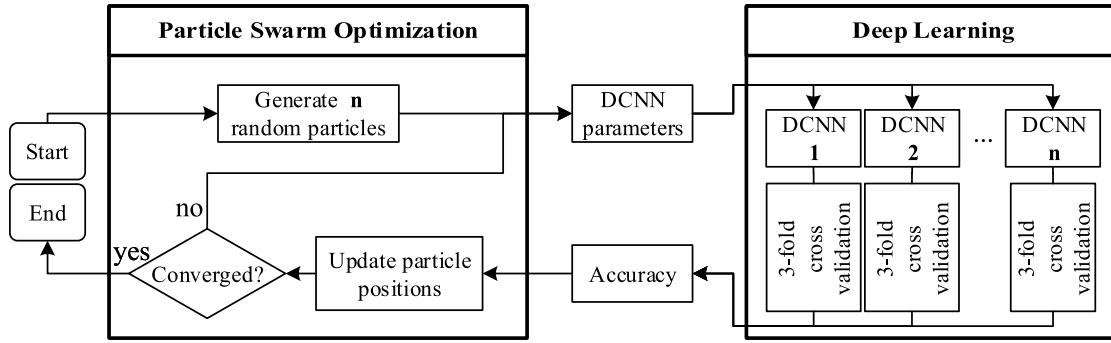


Fig. 2. Hyper-parameter optimization flowchart. In each PSO generation, a set of DCNN parameters are generated, and three-fold cross-validation results of their respective networks are sent back to the PSO routine. This loop continues until the PSO converges.

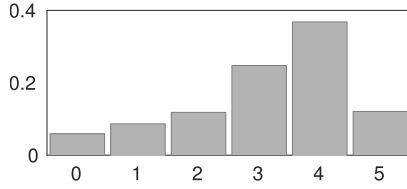


Fig. 3. Distribution of dataset, consisting of 6,916 samples, among six quality-levels based on manual echo scores. Since the data is obtained mostly by expert sonographers with several years of echocardiography examination experience, the distribution leans more towards higher quality scores.

For this project, 6,916 end-systolic A4C view frames were randomly fetched from the VGH echo database with ethics approval of the Clinical Medical Research Ethics Board of the Vancouver Coastal Health (VCH) (H13-02370). Based on the guidelines provided by the VCH Information Privacy Office, data was anonymized, de-identified and stored using encryption throughout the study. No new patients were scanned for this study and no special treatment was taken for handling pathological cases. All images used in this study were acquired using the Philips iE33 ultrasound machine.

B. Manual Quality Assessment

The dataset was examined by an expert cardiologist and an integer quality score of 0 (not acceptable) to 5 (excellent) was assigned to each image based on the following:

- 0) Clear presence of the aortic valve and/or only the interatrial or interventricular septums are visible;
- 1) Chamber boundaries of one or two chambers are visible;
- 2) Chamber boundaries of three chambers are visible;
- 3) Chamber boundaries of three or four chambers are visible, but not sharp enough for quantification of all chambers; heart is off-axis (crooked) or significantly foreshortened;
- 4) Boundaries of three or four chambers are clear for quantification, proper axis, mildly foreshortened;
- 5) Boundaries of four chambers are clear for quantification, proper axis, sharp edges, not foreshortened.

According to the above scoring system and the expert's opinion, samples with a score of below 3 are considered uninterpretable with minimal clinical value. Distribution of data among the six quality-levels is demonstrated in Fig. 3.

C. Network Architecture

To optimize the network design, the dataset was randomly partitioned into training-validation (80%) and test sets (20%). Division of quality-levels among the training and test sets were examined via Pearson's χ^2 goodness-of-fit test to confirm that they represent the original data distribution (p -value > 0.05).

The network design was then optimized over the training-validation set using PSO in terms of the following variables: number of conv layers, number and size of kernels in each conv layer, and number of neurons in each fc layer. In this approach, the number of fc layers was fixed to two layers along with a fully-connected neuron that produced the final output of the network.

A diagram of the proposed hyper-parameter optimization methodology is depicted in Fig. 2. In this methodology, PSO was run twice, each time with a narrower search space for the hyper-parameters. Upon completion of the first PSO run, mean and standard deviations of each hyper-parameter across the best 20% solutions (particle positions) were calculated. Search space boundaries of hyper-parameter i were then updated based on the mean ($\bar{s}_i$) and standard deviation (σ_i) as:

$$[L_i, U_i] = [\bar{s}_i - \sigma_i, \bar{s}_i + \sigma_i]. \quad (6)$$

Although $[\bar{s}_i - 3\sigma_i, \bar{s}_i + 3\sigma_i]$ would provide a more robust search space, we chose a narrower range to speed up the convergence. The only categorical hyper-parameter, *i.e.* number of conv layers, was eliminated from the optimization process at the end of the first PSO, as all the top 20% solutions held three conv layers.

Each PSO-run consisted of six particles and a variable number of generations. The performance of each particle at every position was calculated via a three-fold cross-validation. In total, 430 DCNN models were trained throughout the hyper-parameter optimization process, all of which used the same set of folds for their performances to be comparable. Training each model took 2-3 hours. All models were trained with the same training parameters (see Section III-E) and all of them converged after 54 ± 6 epochs.

The resultant network architecture is illustrated in Fig. 4. It consists of three convolutional layers (conv1, conv2, conv3), each followed by ReLU activation functions. All convolutional kernels were convolved with a stride of one on padded inputs to preserve dimensions. The three pooling layers

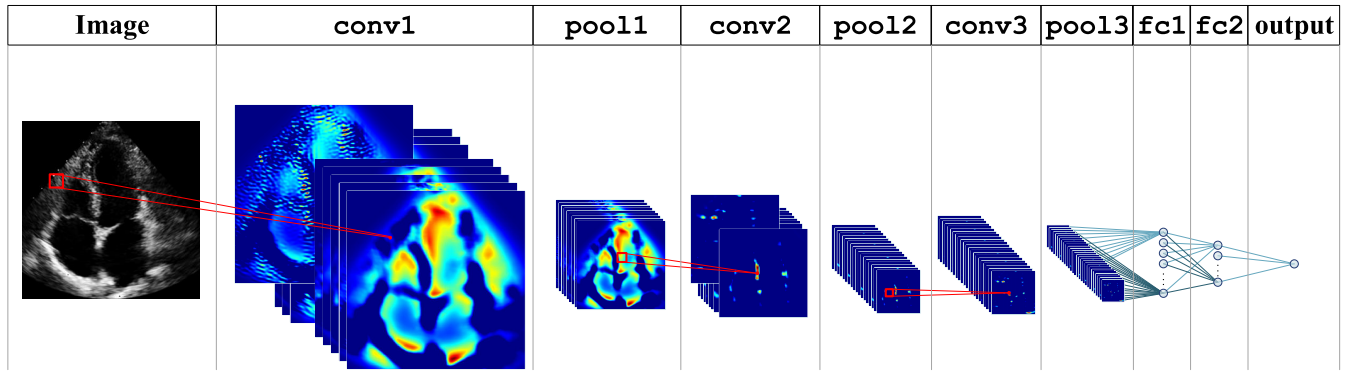


Fig. 4. Network architecture, consisting of three convolutional layers (conv1, conv2, conv3), three max-pooling layers (pool1, pool2, pool3), and two fully connected layers (fc1, fc2). Convolutional layers were applied on padded inputs with stride of one to create same-size outputs; while Max-pooling layers halved the size of their inputs in each dimension. The parameters and inputs of each layer are explained in Table I.

(pool1, pool2, pool3) used 3×3 filters with a stride of two.

In this network, the conv1 layer has 22 kernels of size 17×17 ; the conv2 layer has 53 kernels of size 11×11 ; and the conv3 layer has 64 kernels of size 11×11 . In the fully-connected stage, the network combined local features learned by the convolutional layers into a fewer number of signals using the two fully-connected layers, fc1, fc2, consisting of 1079 and 699 neurons, respectively. Finally, a single fully-connected neuron computed the final output of the network. Parameter settings of the final network architecture is summarized in Table I.

D. Regularization and Data Augmentation

To stabilize learning and prevent the model from over-fitting on the training data, several strategies were used. Regularization is a machine learning technique that adds a penalty term to the loss function to prevent the coefficients (weights) from getting too large. Here, we used a ℓ^2 regularization in the form of $\lambda ||w||_2^2$ with $\lambda = 0.02$.

Dropout layers (dropout) prevent co-adaptation of feature extractors and encourage neurons to follow the population behavior [36]. In each step of training, a dropout layer removes some units of its previous layer from the network. These units are chosen randomly based on the probability parameter of the dropout layer. This means that the network architecture changes in every training step. Thus, dropout layers integrate different network architectures into a single model [37]. In a sense, the dropout layer adds random noise to hidden layers to prevent over-fitting. In our design, a dropout layer was deployed after each fc layer with a dropout probability of 0.6.

To add translational invariance to the model and, at the same time, prevent over-fitting, samples were modified, on-the-fly, during training. In each mini-batch, each sample was translated horizontally by a random number of pixels generated from a zero-mean Gaussian distribution with a standard deviation of $1/20$ of the image width.

Rotational invariance was also encouraged by slightly rotating images, on-the-fly, with a random degree generated from

TABLE I
NETWORK ARCHITECTURE SUMMARY. NUMBERS IN THE $w \times h \times n$ FORMAT REPRESENT THE WIDTH, HEIGHT AND DEPTH OF THE PARAMETER, RESPECTIVELY

Layer	Parameter	Value
conv1	kernel size	17×17
	#kernels	22
	stride	1
	input size	267×267
	output size	$267 \times 267 \times 22$
pool1	kernel size	3×3
	stride	2
	input size	$267 \times 267 \times 22$
	output size	$133 \times 133 \times 22$
conv2	kernel size	11×11
	#kernels	53
	stride	1
	input size	$133 \times 133 \times 22$
pool2	kernel size	3×3
	stride	2
	input size	$133 \times 133 \times 53$
	output size	$66 \times 66 \times 53$
conv3	kernel size	11×11
	#kernels	64
	stride	1
	input size	$66 \times 66 \times 53$
pool3	kernel size	3×3
	stride	2
	input size	$66 \times 66 \times 64$
	output size	$22 \times 22 \times 64$
fc1	#neurons	1079
	input size	$22 \times 22 \times 64$
	output size	1079
fc2	#neurons	699
	input size	1079
	output size	699
output	#neurons	1
	input size	699
	output size	1

a zero-mean Gaussian distribution with $\sigma = 7$ degrees and capped to 2σ .

Since the expert quality assessments were based on non-augmented images, translation and rotation values could not increase beyond certain limits. We consulted an expert cardiologist on the study team to estimate the maximum values and to ensure that the above data augmentations did not affect the clinical value and image quality of the training data.

E. Training

Upon completion of the hyper-parameter optimization and finalizing the DCNN architecture, the resultant network was trained on the whole set of training-validation data. Training was repeated three times to account for the random initialization and the deployed stochastic training paradigm to emphasize robustness of the results. The performances of the trained models were evaluated based on the test set. At no stage was test data used or analyzed in the design of the networks or training of the models. All models were trained via the stochastic gradient descent.

During training, a small batch-size of 36 images was favored; to reward persistent reduction in the objective function, we deployed a relatively high momentum of 0.95. The initial learning rate was 0.0002 accompanied by a gradual drop of 0.5 every 1000 iterations. Here, an iteration is defined as a forward pass of the network for a mini-batch accompanied by a backpropagation to update the weights. The parameters of convolutional and fully-connected layers were initialized randomly from a zero-mean Gaussian distribution. This paradigm guaranteed a slow and steady convergence of the network, therefore training parameters such as batch-size, momentum, initial learning rate, and regularization weight were excluded from the hyper-parameter optimization. In each trial, training was continued until the network converged. Convergence was defined as a state in which no further progress was observed in the training loss.

Since data distribution among quality-levels was not uniform (Fig. 3), in an effort to prevent the model from forming a bias towards the more condensed middle scores (*i.e.* 2 and 3), we designed an online mini-batch selection strategy. This routine randomly chose $BatchSize/6$ samples from each quality-level, augmented them on-the-fly, and packed them together to form a mini-batch. As a result, the network always perceived the training data to be uniformly distributed among the quality-levels. Although an epoch was still defined as the number of iterations required for the network to meet all the training samples given a linear mini-batch selection strategy, the training was not affected by this definition.

The Caffe deep learning framework, developed by the Berkeley Vision and Learning Center [38] with a very efficient GPU implementation, was used for training and testing of the designed network. The experiments benefited from the Nvidia GeForce GTX 980 Ti GPU with 2816 CUDA cores and GPU clock of 1 GHz, interfaced via the CUDA runtime platform version 7.5. Using this configuration, the trained model calculated the AES for a frame of 267×267 pixels in less than 10 ms, which is sufficient for a real-time feedback system. We also tested the network with a slower GPU (Nvidia GeForce GTX 480) with 480 CUDA cores and GPU clock of 700 MHz, and the run-time for each frame did not increase beyond 10 ms, still satisfying the real-time requirements.

IV. RESULTS AND ANALYSIS

The designed DCNN (Fig. 4) was trained three times on the training data and was evaluated on the test set against expert cardiologist's manual scores. The performances of the trained

TABLE II

THE PERFORMANCE OF THE THREE TRAINED MODELS ON EACH QUALITY-LEVEL AND IN TOTAL. ALTHOUGH THEIR PERFORMANCES IN EACH QUALITY-LEVEL SLIGHTLY VARIES, THEIR OVERALL ACCURACIES MATCH

Model	Mean Absolute Error (MAE)						Total
	0	1	2	3	4	5	
model1	0.64	0.78	0.85	0.79	0.59	0.86	0.72
model2	0.66	0.84	0.86	0.77	0.58	0.87	0.72
model3	0.69	0.83	0.83	0.75	0.59	0.86	0.72
Average model	0.66	0.80	0.83	0.76	0.58	0.86	0.71

Overall and level-wise MAE are presented.

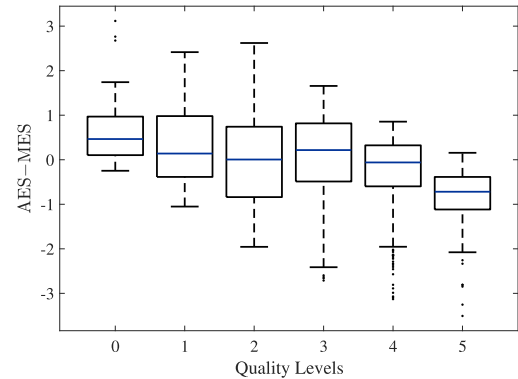


Fig. 5. Distribution of error in each quality-level.

TABLE III

CONFUSION MATRIX OF THE AVERAGE MODEL. THE DARK AND LIGHT GREEN NUMBERS INDICATE THE NUMBER AND RATIO OF EXACT ESTIMATES AND THE NUMBER AND RATIO OF ESTIMATES WITH $|AES - MES| = 1$, RESPECTIVELY

		AES						$p(AES - MES)$	
		0	1	2	3	4	5		
MES	0	37	29	4	3	0	0	0.51	0.90
	1	22	47	31	19	0	0	0.40	0.84
	2	14	53	57	45	14	1	0.31	0.84
	3	4	20	66	123	129	6	0.35	0.91
	4	0	7	33	109	284	73	0.56	0.92
	5	0	0	5	18	82	51	0.33	0.85

models were evaluated as the mean absolute error (MAE) between the predicted AES and the expert's manual echo scores (MES).

Table II presents the performance of the three trained models for each quality-level as well as the overall accuracy of each model. All trained models demonstrated a mean absolute error of 0.72 ± 0.59 when compared with the MES. Predicted values of the three trained models were averaged and reported as the average model in Table II with the mean absolute error of 0.71 ± 0.58 .

The distribution of error for each quality-level of the final model, calculated as $Error = AES - MES$, is depicted in the boxplot of Fig. 5. The confusion matrix of the rounded network AES assessed against the manual scores is also presented in Table III. A few test samples along with their corresponding expert's scores and their AES are depicted in Fig. 6.

A. Intra-Rater Reliability

The intra-rater reliability [39] of the expert was evaluated via re-scoring a subset of 200 random samples. The distribution of quality-levels in this randomly-selected subset was

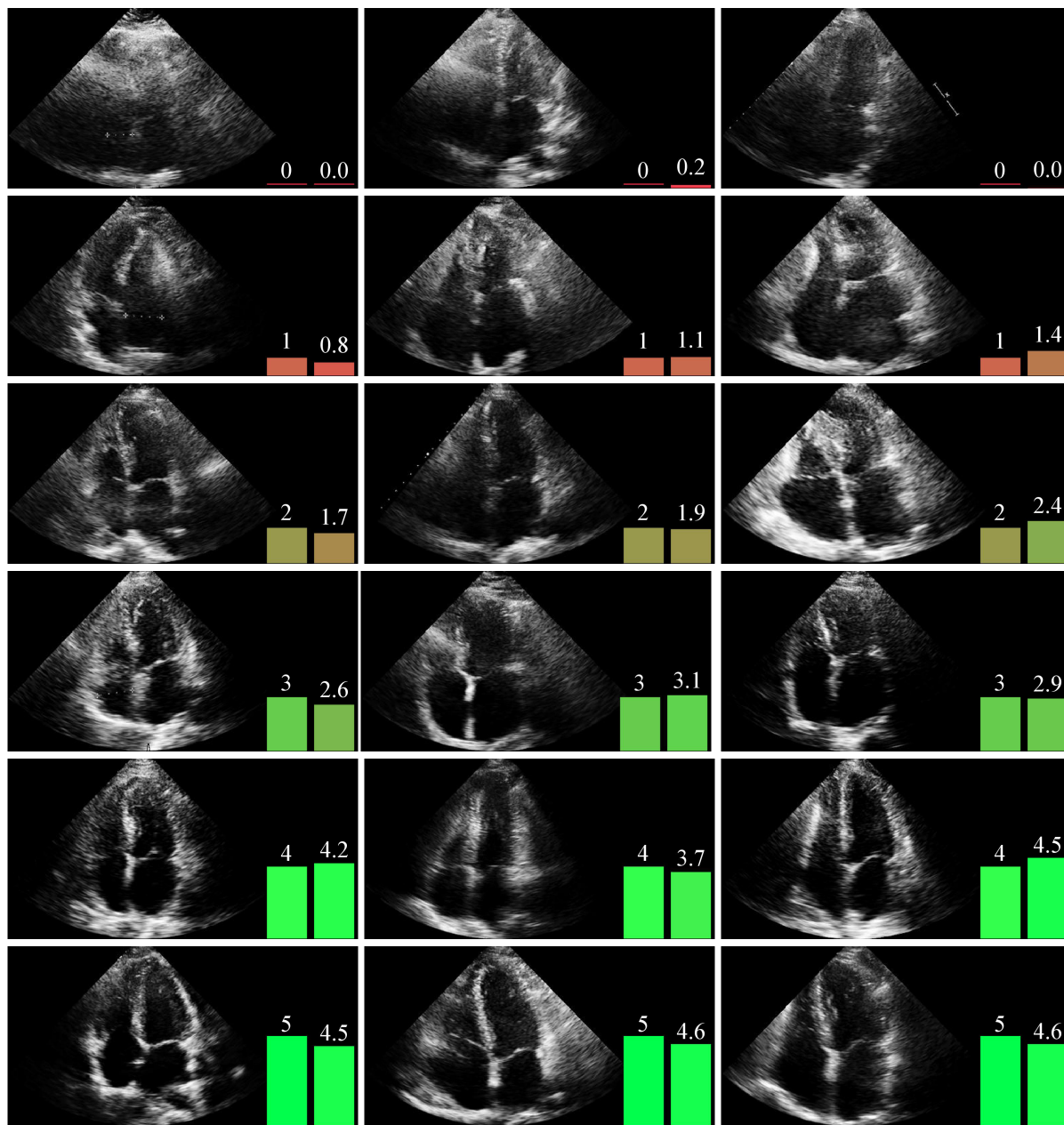


Fig. 6. Examples of automatic quality assessments of A4C echos (AES, right bar) along with the expert's opinion (MES, left bar).

compared to the original data using Pearson's χ^2 goodness-of-fit test to confirm that it represents the original data distribution (p -value > 0.05).

Re-scored samples demonstrated a high agreement with the original scores (Cohen's $\kappa = 0.80$, p -value < 0.05). The intra-rater reliability was also calculated at 0.65, which is comparable to the estimated MAE of the network.

B. Feature Visualization

Convolutional models make it possible to visualize learned hierarchical features to demonstrate what the network has encoded into the fully connected layers. With this property, one can visualize the learned kernel weights (filters) as well as the interpretable features in each layer. The latter can be

mapped to the anatomical structures, which are visible on the original input image.

A visualization method was proposed by Zeiler and Fergus using deconvolutional networks [40], in which they convolved the resultant feature-maps with the transposed versions of their corresponding kernels to reverse the function of `conv` layers. They also proposed max-unpooling via special switches which record the location of the local max in each pooling region during pooling. However, they only managed to visualize single prominent activations in each layer separately and did not visualize the whole feature-set encoded in a layer.

Here, we propose a novel technique which not only visualizes a single chosen feature from any layer, but can also visualize any combination of features. To visualize feature-maps, they need to be back-projected into the pixel-space

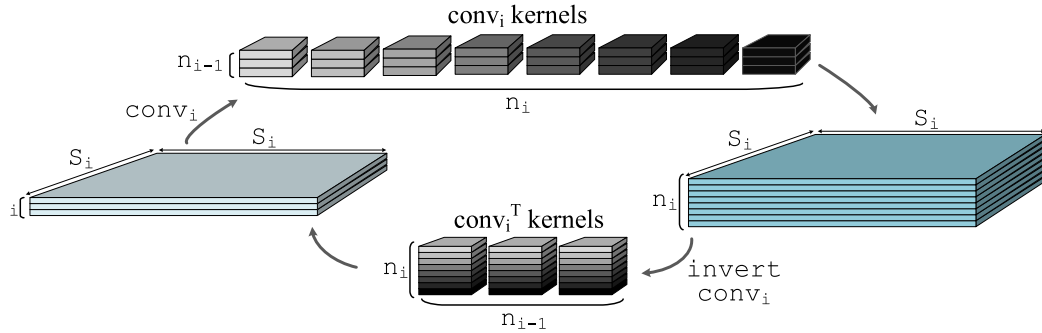


Fig. 7. Inverting the conv layer operation. In this sample diagram, the eight stacks of conv_i kernels are transposed and rearranged into three stacks of conv_i^T kernels. The rearrangement is demonstrated via gray-scale shades. Arrows represent 3D convolution.

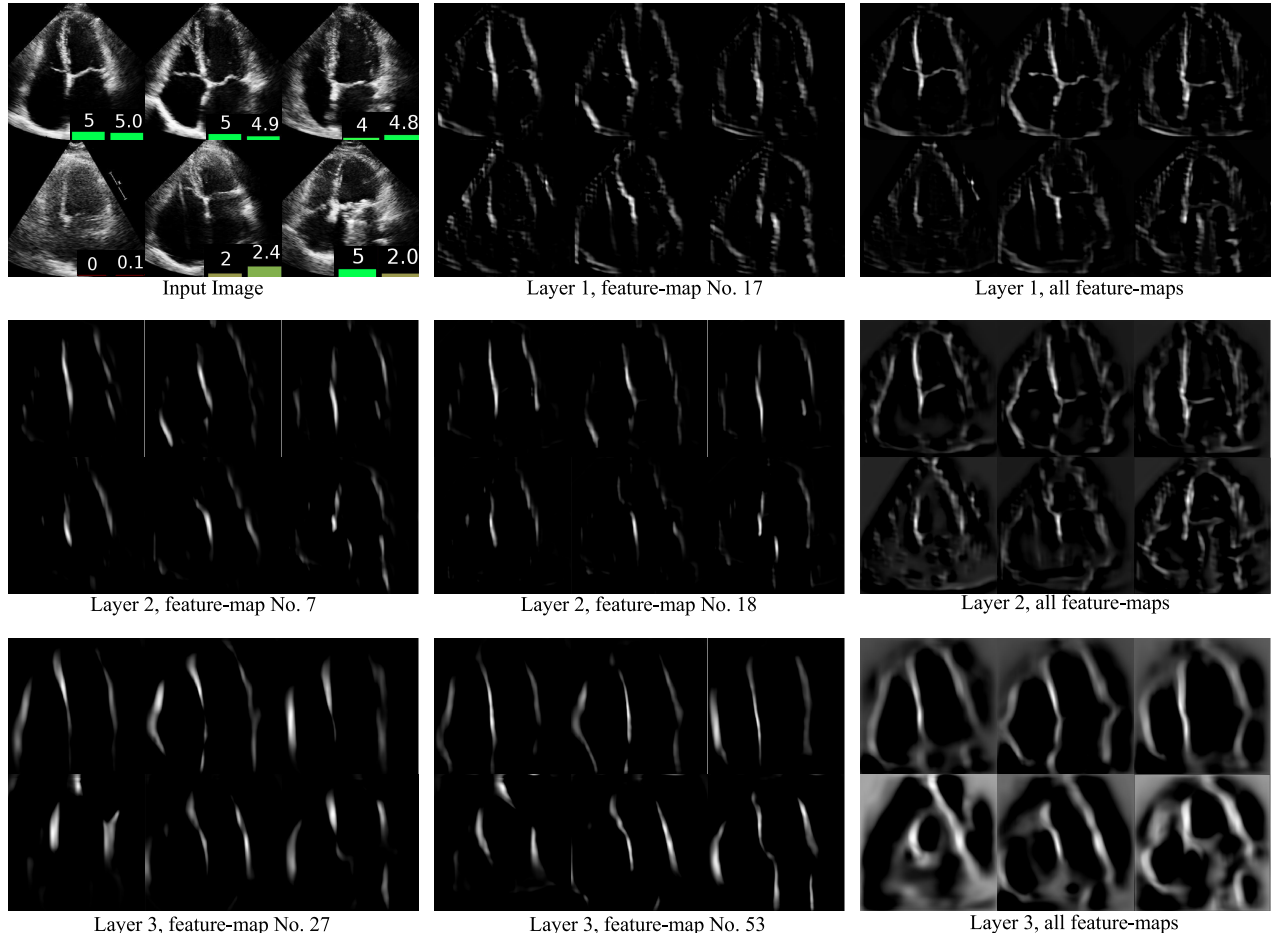


Fig. 8. Feature visualization of different layers on high quality, low quality, and outlier test samples. For each input image, the assigned MES (left bar) and the estimated AES (right bar) are demonstrated. As the network deepens, the level of abstraction increases. Feature-map No. 17 of layer 1 (top-middle) as well as the accumulated feature-maps of layer 1 (top-right) contain almost all the edges of the input image. On the other hand, feature-maps No. 27 and 53 of the third layer (bottom-left and bottom-middle) only represent the lateral walls and the septum. The accumulated feature-map of layer 3 (bottom-right) demonstrates an abstract representation of the heart's anatomy while ignoring the noise from the previous layers. It is worth mentioning that there is no direct one-to-one relationship between single feature-maps across different layers, as any feature-map of a deeper layer is result of a convolution on all the feature-maps of the previous layer.

via inverting the convolution and pooling operations. The inversions, however, will not result in the exact reconstruction of the layer's input as these are lossy operations.

To invert the pooling operation of the 3×3 max-pooling kernels with stride of two, we propose to use 5×5 bilinear kernels to enlarge the image back to its original dimensions.

To invert the convolution operation, transposed version of the conv kernels are convolved with outputs of the layer

(feature-maps) [40]. Inputs and outputs of the i th conv layer can be summarized into the 3D matrices $w_i \times h_i \times n_{i-1}$ and $w_i \times h_i \times n_i$, respectively, where w_i and h_i are width and height of the feature-map, and n_i represents the number of kernels for the i th conv layer (#kernels, see Table I). As explained in Section III-C, all the conv kernels in our design were applied with a stride of 1 on padded inputs resulting in outputs with the same width and height.

Kernel weights of the i th layer can also be summarized into a 4D matrix, $s_i \times s_i \times n_{i-1} \times n_i$, where s_i represents the equal dimensions of the kernel. In other words, this conv layer has n_i kernels, each with $s_i \times s_i \times n_{i-1}$ weights, which generates a single 2D feature-map via a 3D convolution with the input.

In our visualization method, in order to invert a kernel's convolutional operation, it is disassembled into n_{i-1} 2D square kernels of size s_i , each of which is transposed and convolved with the 2D output of the kernel to produce n_{i-1} reconstructed inputs. The same operation is repeated for the $k \leq n_i$ other kernels whose feature-maps are of interest. Finally, the resultant k inputs of size $w_i \times h_i \times n_{i-1}$ are summed up to reconstruct the 3D input weight matrix.

The above operation can be demonstrated as a 3D convolution of the conv layer's output with the transposed and rearranged 4D kernel matrix. The mentioned transposed convolution, demonstrated in Fig. 7, is referred to as deconvolution in deep learning and computer vision literature [40], [41], an ambiguous term which we avoid in this article.

To visualize the network's understanding of a sample echo image, the top-most activations in the pool1, pool2 and pool3 layers were back-projected into the pixel-space, and visualized separately. Moreover, the accumulated back-projections of all feature-maps in these three layers were also visualized. Example outputs of this experiment are demonstrated in Fig. 8.

V. DISCUSSION AND CONCLUSION

It has been suggested that providing real-time quality feedback during image acquisition encourages less experienced sonographers to acquire echo images of better quality [16]. In an attempt to provide such feedback, we propose a framework for automatic quality assessment of echo data. We take advantage of a fairly large dataset of 6,916 A4C images to design, optimize and train our deep neural network model. The result showed an mean absolute error of 0.71, which is in the same order as the intra-rater reliability of the expert.

As reported in Table II, the three trained models demonstrate a mean absolute error of 0.72 and exhibit almost the same performance on each quality level. This is an indication of the independence of the results from the random weight initializations in conv and fc layers, random mini-batch selection, random data augmentation, and random dropout of units during training.

According to the confusion matrix, which is based on rounded scores (Table III), among the 1,386 test samples were only 20 images with $|AES - MES| > 2$, most of which were further confirmed with the cardiologist to be outliers of the labeling process. An exception to this, along with its features, is shown in Fig. 8. The visualized features of layer 3 for this input image show that the network has failed to recognize the heart structure surrounding the left atrium, as the patient has a prosthetic mitral mechanical valve that has caused an artifact on the left atrium. While the cardiologist had ignored the artifact, knowing that it could not be avoided, and scored the image as high quality (quality-level 5), the trained network estimated its quality as low (quality-level 2) due to its obscured left atrium. This was also the case for another test sample,

where a prosthetic material was present on the interventricular septum. Based on these results, it is our understanding that, since cardiac prostheses are not common, the network did not learn to ignore prosthetic artifacts during training.

The visualized features (Fig. 8), confirmed that the network has actually learned sensible and interpretable characteristics of the echo with different levels of abstraction. As depicted in this figure, feature-maps of the third layer demonstrate an abstract representation of the heart's anatomy, which will later be used in the fc layers for the regression model. Moreover, it can be understood that visibility of the septum and lateral walls, which eventually form the four chambers, are among the prominent factors that contribute to a higher AES.

The proposed approach does not rely on hand-crafted features or view-specific templates [16], [17], but rather it takes advantage of a set of automatically learned hierarchical features. Consequently, it is not directly affected by the low-level variations of data, such as the tissue-dependent speckle patterns of ultrasound images. As a result, no preprocessing step, other than on-the-fly cropping, was applied on the dataset, in neither training nor testing phases. To demonstrate the negligible influence of speckle, repeating the tests after filtering the images with a nonlocal means-based method had no significant effect on the performance.

In our experiment, the trained model, deployed on the GPU, calculated the AES of an image with 267×267 pixels in less than 10 ms. This is faster than the native frame rate and faster than the 35 FPS rate obtained with a Hough transform approach applied on the parasternal echocardiograms of size 50×50 pixels [17].

Although Section III-B provides precise definitions for the six quality-levels, the expert cardiologist could not always decide on an exact quality score for a given echo due to the nature of ultrasound imaging. This is demonstrated via the kappa coefficient ($\kappa = 0.80$) as well as the intra-rater reliability coefficient (0.65). Moreover, in our effort to encourage simplicity, optimize clinicians's time, and generate more manually labeled samples, MES was limited to ordinal numbers. However, quality is a continuous property which justifies a continuous AES. Consequently, in our analysis, we did not round the estimated AES values to the nearest integers. Rounding up AES to the nearest integer resulted in a better MAE of 0.69.

Although the proposed model and DCNN architecture are promising, some challenges remain unanswered. Future steps include covering other standard echo views and extending the framework to respond to the cardiac cycle (echo cine) as a whole, rather than a single frame. In this research, the end-systolic frame was selected to accentuate the pointed apex that is present in an optimal A4C image. Given that this was the first sample set, we sought to use a frame that can be scored more precisely. Lastly, in order to reduce the variability among observers, the scoring system for the manual quality assessment should be upgraded to more distinctive definitions.

We believe it is feasible to train a generic DCNN that extracts distinctive features from any echo view. With this approach, to interpret the extracted features for a specific view, it would suffice to limit the training only to the fully-connected layers. Such a generalized design reduces the training time,

requires much smaller training set, and ensures optimal quality echo acquisition via real-time feedback to sonographers.

According to Fig. 3, the quality of a quarter of the used samples was below the minimal clinical value of 3 (Section III-B), for which the patient needs to be recalled for further testing and imaging. This procedure is costly for the health-care system. Therefore, our ultimate goal is to improve echo by reducing observer variability in data acquisition using a real-time feedback mechanism that helps the operator to re-adjust the probe and acquire an optimal echo.

By minimizing operator dependency on echo acquisition and analysis, this research would lead to widespread use of echo at any point-of-care, hence it would enable early and timely diagnosis and treatment of high-risk patients with improved accuracy, quality assurance, work-flow and throughput.

REFERENCES

- [1] Q. Ciampi and B. Villari, "Role of echocardiography in diagnosis and risk stratification in heart failure with left ventricular systolic dysfunction," *Cardiovascular Ultrasound*, vol. 5, no. 1, p. 34, 2007.
- [2] S. D. Solomon, *Essential Echocardiography*. New York, NY, USA: Humana Press, 2007.
- [3] R. M. Lang *et al.*, "Recommendations for cardiac chamber quantification by echocardiography in adults: An update from the american society of echocardiography and the european association of cardiovascular imaging," *J. Amer. Soc. Echocardiogr.*, vol. 28, no. 1, pp. 1–39, Aug. 2016.
- [4] S. F. Nagueh *et al.*, "Recommendations for the evaluation of left ventricular diastolic function by echocardiography: An update from the american society of echocardiography and the european association of cardiovascular imaging," *J. Amer. Soc. Echocardiogr.*, vol. 29, no. 4, pp. 277–314, Apr. 2016.
- [5] D. L. Mann, D. P. Zipes, P. Libby, R. O. Bonow, and E. Braunwald, "Echocardiography," in *Braunwald's Heart Disease: A Textbook of Cardiovascular Medicine*. Philadelphia, PA, USA: Saunders, 2015, ch. 14.
- [6] M. Grossgasteiger *et al.*, "Image quality influences the assessment of left ventricular function: An intraoperative comparison of five 2-dimensional echocardiographic methods with real-time 3-dimensional echocardiography as a reference," *J. Ultrasound Med.*, vol. 33, no. 2, pp. 297–306, 2014.
- [7] D. A. Tighe *et al.*, "Influence of image quality on the accuracy of real time three-dimensional echocardiography to measure left ventricular volumes in unselected patients: A comparison with gated-SPECT imaging," *Echocardiography*, vol. 24, no. 10, pp. 1073–1080, Nov. 2007.
- [8] E. O. Chukwu *et al.*, "Relative importance of errors in left ventricular quantitation by two-dimensional echocardiography: Insights from three-dimensional echocardiography and cardiac magnetic resonance imaging," *J. Amer. Soc. Echocardiogr.*, vol. 21, no. 9, pp. 990–997, Sep. 2008.
- [9] D. S. Blondheim *et al.*, "Reliability of visual assessment of global and segmental left ventricular function: A multicenter study by the israeli echocardiography research group," *J. Amer. Soc. Echocardiogr.*, vol. 23, no. 3, pp. 258–264, 2010.
- [10] S.-W. Huang *et al.*, "Detection and display of acoustic window for guiding and training cardiac ultrasound users," *Proc. SPIE*, vol. 9040, p. 904014, Mar. 2014.
- [11] L. Løvstakken, F. Orderud, and H. Torp, "Real-time indication of acoustic window for phased-array transducers in ultrasound imaging," in *Proc. IEEE Ultrason. Symp.*, Oct. 2007, pp. 1549–1552.
- [12] J. H. Park, S. K. Zhou, C. Simopoulos, J. Otsuki, and D. Comaniciu, "Automatic cardiac view classification of echocardiogram," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
- [13] G. N. Balaji, T. S. Subashini, and N. Chidambaram, "Automatic classification of cardiac views in echocardiogram using histogram and statistical features," in *Proc. ICICT*, Kochi, India, Dec. 2014. [Online]. Available: <https://goo.gl/izpXsp>
- [14] H. Wu, D. M. Bowers, T. T. Huynh, and R. Souvenir, "Echocardiogram view classification using low-level features," in *Proc. IEEE 10th Int. Symp. Biomed. Imag.*, Apr. 2013, pp. 752–755.
- [15] R. Kumar, F. Wang, D. Beymer, and T. Syeda-Mahmood, "Echocardiogram view classification using edge filtered scale-invariant motion features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 723–730.
- [16] S. R. Snare, H. Torp, F. Orderud, and B. O. Haugen, "Real-time scan assistant for echocardiography," *IEEE Trans. Ultrason., Ferroelect., Freq. Control*, vol. 59, no. 3, pp. 583–589, Mar. 2012.
- [17] S. K. Pavani *et al.*, "Quality metric for parasternal long AXis B-mode echocardiograms," in *Proc. 15th Int. Conf. MICCAI*, 2012, vol. 15, no. 2, pp. 478–485.
- [18] N. El-Zehiry *et al.*, *Learning the Manifold of Quality Ultrasound Acquisition* (Lecture Notes in Computer Science), vol. 8149. Berlin, Germany: Springer, 2013, pp. 122–130.
- [19] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.
- [20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, Sep. 2015.
- [21] J. Gu *et al.*, (Dec. 2015). "Recent advances in convolutional neural networks." [Online]. Available: <https://arxiv.org/abs/1512.07108>
- [22] S. Srinivas *et al.*, "A taxonomy of deep convolutional neural nets for computer vision," *Frontiers Robot. AI*, vol. 2, no. 36, Jan. 2016.
- [23] A. H. Abdi *et al.*, "Automatic quality assessment of apical four-chamber echocardiograms using deep convolutional neural networks," *Proc. SPIE*, vol. 10133, pp. 101330S-1–101330S-7, Feb. 2017.
- [24] Y. L. Cun *et al.*, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. San Francisco, CA, USA: Morgan Kaufmann, 1990, pp. 396–404.
- [25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, Nov. 1998.
- [26] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, Fort Lauderdale, FL, USA, Apr. 2011. [Online]. Available: <http://www.jmlr.org/proceedings/>
- [27] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
- [28] X. Jin *et al.*, "Deep learning with s-shaped rectified linear activation units," in *Proc. 13th AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2016, pp. 1737–1743.
- [29] D. Scherer, A. Müller, and S. Behnke, *Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition* (Lecture Notes in Computer Science), vol. 6354. Berlin, Germany: Springer, 2010, pp. 92–101.
- [30] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. 23rd IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 2559–2566.
- [31] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 281–305, 2012. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2188395>
- [32] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2546–2554.
- [33] B. A. Garro and R. A. Vázquez, "Designing artificial neural networks using particle swarm optimization algorithms," *Comput. Intell. Neurosci.*, vol. 2015, Jun. 2015, Art. no. 20. [Online]. Available: <https://goo.gl/7ECBrC>
- [34] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2001.
- [35] M. Clerc and J. Kennedy, "The particle swarm—Explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.
- [36] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. (Jul. 2012). "Improving neural networks by preventing co-adaptation of feature detectors." [Online]. Available: <https://arxiv.org/abs/1207.0580>
- [37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [38] Y. Jia *et al.* (Jun. 2014). "Caffe: Convolutional architecture for fast feature embedding." [Online]. Available: <https://arxiv.org/abs/1408.5093>
- [39] K. L. Gwet, "Intrarater reliability," in *Wiley Encyclopedia Clinical Trials*. Hoboken, NJ, USA: Wiley, 2008, pp. 1–14.
- [40] M. D. Zeiler and R. Fergus, *Visualizing and Understanding Convolutional Networks* (Lecture Notes in Computer Science), vol. 8689. Cham: Springer, 2014, pp. 818–833.
- [41] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2018–2025.