# TRACKABLE-SPECKLE DETECTION USING A DUAL-PATH CONVOLUTIONAL NEURAL NETWORK FOR NODES SELECTION IN SPECKLE TRACKING ECHOCARDIOGRAPHY

**M. Shiri[1,*], H. Behnam[1], H. Yeganegi[2], Z.A. Sani[3] and N. Nematollahi[1]**

[1]Department of Biomedical Engineering, School of Electrical Engineering, Iran University of Science and Technology, 1684613114, Narmak, Tehran, Iran.

[2]Graduate School of Systemic Neurosciences, Ludwig Maximilians University, Grosshaderner Street 2, D-82152, Planegg, Munich, Germany.

[3]Rajaie Cardiovascular Medical & Research Center, Tehran University of Medical Sciences, 1416634793 , Keshavarz, Tehran, Iran.

[*]Corresponding Author's Email: milad70shiri@gmail.com

**Article History**: Received March 30, 2022; Revised April 19, 2022; Accepted May 25, 2022

**ABSTRACT:** Speckle tracking echocardiography (STE) is widely used to quantify regional motion and deformation of heart tissues. Before tracking, a segmentation step is first carried out, and only a set of nodes in the segmented model are tracked. However, a random selection of the nodes even after tissue segmentation could lead to an inaccurate estimation. In this paper, a convolutional neural network (CNN)-based method is presented to detect trackable speckle spots that have important properties of the texture for speckle tracking. The proposed CNN was trained and validated on 29500 ultrasound manually labelled image patches extracted from the echocardiography of 65 people. Using the proposed network, in silico experiments for automatic node selection were conducted to investigate the applicability of the proposed method in speckle tracking. The results were statistically highly significant (P<.001) and demonstrated that the proposed method has the least tracking error among various existing methods.

***KEYWORDS***: *Speckle Detection, Convolutional Neural Network, Ultrasound Imaging, Echocardiography*

## 1.0 INTRODUCTION

Ultrasound imaging is a promising modality for medical diagnosis, mostly because it is non-invasive, cheap and easy to use [1-3]. Echocardiography, in particular, is an ultrasound-based tool for cardiac imaging which is vastly used for identifying heart defects [4]. In addition to visual assessment of the heart, many computer-aided and fully automatic techniques have been proposed for qualitative and quantitative cardiac function evaluations [5-7]. Tracking the myocardium and calculating its elasticity provides us with a common method of quantifying the mechanical activity of the heart [8]. Two-dimensional (2D) speckle tracking is commonly used for tissue tracking based on ultrasound B-mode images. [9,10].

Speckles are the bright points seen in an echocardiogram, which are the results of diffuse scattering [10]. Since speckles are caused by interference of backscattered signals from extremely small neighbouring elements in the tissue, the speckle pattern is strictly correlated with the microstructure of the underlying tissue [11]. As a result, the topological pattern of closely-connected speckles is usually similar over consecutive frames and can be used for tracking [12]. However, speckles are not always suitable for tracking and could lead to misestimating displacements in speckle tracking. Hence, speckle detection algorithms are not useful for spotting trackable speckle patterns.

Speckle tracking methods essentially use mentioned property of speckle pattern to estimate the displacement of desired points between two or more frames. Occasionally, speckle tracking is carried out for each point of the entire image or at least of the segmented model [9,12]. But more likely, a segmentation is first carried out, and only a selection of nodes is tracked. Either way, nodes should be selected wisely in a suitable way for tracking purposes; otherwise, it could lead to miscalculation of displacement, even for a node selected randomly or manually from the segmented region[13]. The tracking nodes are usually selected manually and occasionally automatically based on textural features. Manually selecting nodes depends on the observer's precision and is often difficult to proceed. Therefore, automatic node selection methods are preferred for their accuracy and speed compared to observer's judgment. As explained in [14] and [11], speckle characterization and detection methods are useful for selecting important and informative regions of the image and, therefore, node selection.

The earliest approaches for image-based speckle characterization were mostly based on the extraction of the first-order features. Mean, variance, skewness, kurtosis and energy are some of these texture statistics that can be calculated from the first-order histogram of the gray level image. Speckle discrimination properties of statistics were evaluated by Prager et al. [15] using a combination of simulations and the homodyned-k distribution. Marti et al. [16] presented a fully automatic speckle detection method by computing the statistical features from the ultrasound image and finding optimally discriminant low-order speckle statistics. They succeeded in classifying regions as speckle or non-speckle by defining application-specific discriminant functions. Azar et al. [17] proposed a new combination of statistical features and explored their properties for speckle detection. These features were used as inputs to unsupervised clustering algorithms for the speckle classification. However, first-order statistics contain information only about the intensity values and the gray level distribution of the image, not about the pixel neighbourhood relationships. Therefore, the statistic features for two different windows (one with a speckle pattern, the other with a non-speckle pattern) with similar first-order histograms are indistinguishable. As a solution, second-order features derived from the co-occurrence matrix can represent this spatial relationship between pixels. These features include angular second moment, correlation, contrast and entropy, also known as Haralick features [18]. In [19] and [20], Wagner et al. evaluated second-order statistics for detecting and classification of speckle texture in diagnostic ultrasound. Carmo et al. [21] assessed the performance of several speckle detection methods that employed co-occurrence matrices for B-mode images. Widynski et al. [14] proposed a method for speckle spot detection using a morphological tree representation. They used detected speckle spots as markers for speckle tracking.

As shown in the result section, existing methods are not sufficient for automatic detection of suitable nodes, which is mainly because of two reasons: first, speckle patterns are complicated; therefore, it is not easy to model them based on some first and second-order features, second, mostly the information in the consecutive frame for detecting trackable speckles has been neglected. Moreover, Brynolfsson et al. [22] reported that Haralick features are sensitive to the image data. The result is highly dependent on parameter settings such as noise,

resolution and number of gray levels.

Since deep learning has been recently used as a powerful alternative for ultrasound image enhancement, segmentation and speckle reduction [23-25], it seems interesting to employ convolutional neural networks in speckle detection applications. Although numerous advanced techniques have been proposed for ultrasound speckle reduction [23-26], many studies on speckle detection based on ultrasound B-mode images are limited [11,21].

This study proposes a convolutional neural network (CNN) with a specialized structure for detecting trackable speckle spots, which can be employed in ultrasound speckle tracking applications. Specifically, we are interested in regions where speckle pattern has more properties of texture and stability over time, making them useful for tracking. The proposed network is able to classify a region of interest into two categories of trackable-speckle and non-trackable patterns. It should be noted that a region is defined as a squared window, and a node is placed at the centre of the window.

In order to find the optimal network, several CNNs with different configurations were trained and tested on a labelled dataset, including about 29500 image patches of ultrasound trackable-speckle and non-trackable patterns. To evaluate the network's performance, a comprehensive comparison between the proposed method and the existing methods, i.e. combination of feature extraction and supervised learning methods, was conducted. In order to investigate the applicability of the proposed method in speckle tracking, in silico experiments were conducted and compared with the state-of-the-art methods. The rest of the paper is organized as follows: In section 2, the concepts, methods and problems in speckle detection are briefly explained. In section 3, the proposed method is examined in detail. Section 4 explains the procedure of data preparation and conducting experiments. The results of the network-tuning and the evaluations are presented in section 5, and finally, section 6 concludes the study.

## 2.0 BACKGROUND

In this section, the importance and background of speckle detection are briefly explained. Specifically, the conventional approach for speckle detection is overviewed.

## 2.1 Speckle Detection

Speckles are also known as natural acoustic markers because they result from accumulating backscattered echoes from small elements in the tissue. Although they cannot exactly represent the structure of the tissue and are considered noises in some cases, speckles are highly correlated with the underlying microstructure of the tissue [11]. As a result, speckles within a window are usually stable and can be tracked when the associated region of the tissue moves [13,27].

A well-established algorithm for speckle tracking is block matching [28]. Considering that Frame A and Frame B are two consecutive frames in a sequence of echocardiograms, in the block matching technique, a window in frame a is compared with the neighbouring windows in frame b. The window with the highest matching score in frame b is considered the best match for the window in frame a. This selection is how we can tell that a tissue region is moved to a new position. In order to achieve accurate tissue tracking, the window in the frame should have a stable, trackable speckle pattern; otherwise, it leads to poor estimation of the new location. In other words, speckles with a similar topological structure over the next few frames are the best choices for speckle tracking. In this study, trackable speckle detection is a term that is used for examining whether a window has such a speckle pattern or not. Here, the window examined in speckle detection is called the target window. A window extracted from an image is also known as a patch.

In this study, the following criteria were considered as the underlying rule to classify the trackable speckle visually: "A batch of closely-connected speckles in a frame appears in the next frame with or without a geometric transformation, following a similar topology". Based on [14], the underlying definition of a speckle pattern is purely topological. Therefore, no assumption should be made about the intensity values.

## 2.2 Conventional Approach

A common technique for speckle detection usually uses feature extractors and machine learning models [11,17]. First, a set of statistical features is calculated for the target window. Then a decision is made to recognize the window either as trackable-speckle or non-trackable

patterns based on these values. Supervised classifiers can perform this categorizing task. It is worth mentioning that the performance of the speckle detector is highly dependent on the features and the supervised learning algorithm that we choose. Here, to compare our proposed model with conventional methods, the optimal combination of features and models is explained. A set of features suitable for speckle detection based on the first-order and second-order (specifically Haralick features [18]) statistics includes Mean, Variance, Skewness, Kurtosis, Angular Second Moment (ASM), Contrast, Correlation and Homogeneity [16,29]. Their mathematical formulas are described in Table 1. The first four measures can be calculated based on pixel intensity, while the rest should be computed using the co-occurrence matrix. A co-occurrence matrix (P ) displays the frequency that two pixels with gray levels I1 and I2 appear in the window separated by a relative distance d in relative orientation θ [30]. Formally, given the image f with a set of G discrete intensity levels, the

matrix Pdθ(i, j) is defined such that its (i, j)th entry is equal to the number of times that

$$f(x1, y1) = i \; and \; f(x2, y2) = j \tag{1}$$

where

$$(x2, y2) = (x1, y1) + (d \cos\theta, d \sin\theta). \tag{2}$$

In the matrix $Pd\theta(i, j)$, i and j represent rows and columns, respectively.

A short description for each feature is as follows. The Mean represents the average level of intensity of the window being examined. The Variance measures how far the intensity values are spread out from the mean. Skewness and Kurtosis indicate the degree of histogram asymmetry around the mean and the histogram sharpness, respectively. ASM is also known as Energy which measures the smoothness of the image. The Contrast represents the local level of differences. The relation between pixels in two different directions is measured by Correlation. Finally, low-contrast images show high values for Homogeneity. Equation (2) indicates that changing $d$ and $\theta$ results in different co-occurrence matrices and, consequently, several versions of Haralick features. Four angles of 0°, 45°, 90° and 135° are usually picked as the orientation $\theta$. Since choices for d depends on the image parameters, e.g. texture, resolution and dimensions, three relative distances of $1px$, $3px$ and $7px$ were considered. In practice, for

each *d*, the resulting values for the four directions are averaged out. Thus, the total number of features extracted for each image patch in the dataset was sixteen.

Table 1. Mathematical description of selected first-order and second-order features

| Feature | Formula |
|---|---|
| Mean(m) | $\sum_{x,y}^{M,N} \frac{f(x,y)}{M \times N}$ |
| Variance(σ2) | $\sum_{x,y}^{M,N} \frac{(f(x,y)-m)^2}{M \times N}$ |
| Skewness | $\sum_{x,y}^{M,N} \frac{(f(x,y)-m)^3}{M \times N \times \sigma^3}$ |
| Kurtosis | $\sum_{x,y}^{M,N} \frac{(f(x,y)-m)^4}{M \times N \times \sigma^4}$ |
| ASM | $\sum_{i,j} P(x,y)^2$ |
| Contrast | $\sum_{i,j} \lvert i-j \rvert^2 + log\ P(x,y)^2$ |
| Correlation | $\sum_{i,j} \frac{(i-\mu_i) \times (j-\mu_j)}{\sigma_i \times \sigma_j} \times P(i,j)$ |
| Homogeneity | $\sum_{i,j} \frac{P(i,j)}{1+\lvert i-j \rvert^2}$ |

Note: $\mu_i$, $\mu_j$, $\sigma_i$ and $\sigma_j$ are the means and standard deviations of resulted vectors from summing the columns and rows of $P(i,j)$, respectively.

In the classification stage, three classifiers of K-Nearest Neighbor (KNN), Random Forest and AdaBoost were employed to label the image patch as trackable-speckle or non-trackable patterns based on the extracted features.

## 2.3 Problems

A conventional method has three major drawbacks that lead to poor trackable-speckle spot detection. The first one is the selection of texture features based on which building a precise model of the complex pattern of ultrasound speckles becomes difficult. The second one concerns that the initial parameters for feature calculation, e.g. d and θ, should be selected concerning the image's properties such as noise, resolution and texture. The conventional approach's last and most important problem is that it relies only on the target window and ignores the next frame's information for detecting a trackable pattern.

## 3.0 THE PROPOSED METHOD

This section introduces a convolutional neural network (CNN) with a novel architecture, called SpeckleDNet, to address the issues explained in section 2.3 for trackable speckle detection.

An important aspect of a CNN is that the feature extraction process is integrated into the network, not as a distinct but inseparable part from the learning process. This architecture lets the network to adaptively learn and extract the most discriminative features based on the relation between the inputs and outputs. Thus, SpeckleDNet inherently solves the problems of the conventional approach related to the selection of features. But the innovational aspect of SpeckleDNet is that it also takes advantage of the next frame because, as discussed in section 2.1, considering the next frame as a complementary source is beneficial for achieving higher performance in detecting trackable speckle spots.

The duty of SpeckleDNet is to correctly classify a target window into two categories of trackable-speckle or non-trackable patterns with the help of the complimentary window. As shown in Figure 1, the target window is a window in the frame meant to be classified, while the complementary window (patch) is the window in frame b that should be used as the extra information. Both windows have the same central position in the frames. Since the speckles within the target window might have a displacement between frames a and b, the size of the complementary window is considered twice the size of the target window to contain surrounding pixels. Since the network is trained with patches with different sizes, the size of the region of interest can be selected from 12×12 to 48×48 pixels. Therefore, the windows should be eventually resized to fit the dimensions of the corresponding input layer.

In order to process both the target and complementary windows, SpeckleDNet has a dual-path structure with two distinct input gates. As illustrated in Figure 2, the two pathways are called the main pathway and the complementary pathway. The main pathway itself is divided into two parts named primary-path and merge-path. The complementary pathway can be merged into the main pathway by concatenating the outputs of the complement pathway and the primary path together and feeding the concatenation result to the merge path. So, the output dimensions of the complement pathway and the primary path are the same. After running comprehensive experiments, the

optimal architecture for SpeckleDNet was fine-tuned as follows. The dimensions of the input layers are 30×30 and 40×40 pixels for the main and complementary pathways, respectively, which any user input image will be resized to. The main pathway starts with the primary part, which consists of two consecutive convolutional layers of Conv (3×3)×32 and Conv (3×3)×128, and one MaxPooling (3×3) layer. The complementary path includes a combination of convolution and pooling layers in the following order: Conv (3×3)×32, MaxPooling (2×2), Conv (3×3) ×64, Conv (3×3)×64 and MaxPooling (2×2). Through these two paths, the dimensions of the target and complementary windows are gradually reduced to the same size of 13×13 pixels. The outputs of the primary and complementary paths are concatenated to have a total of 192 feature maps which then are fed to the merge path. The merge path includes two consecutive convolutional layers of Conv (3×3)×32 and Conv (3×3)×128, one MaxPooling (3×3) layer, a fully connected layer with 512 neurons and finally one neuron as the output. For convolutional layers, the values in the parentheses show the dimensions of the filters, and the third value indicates the number of filters.

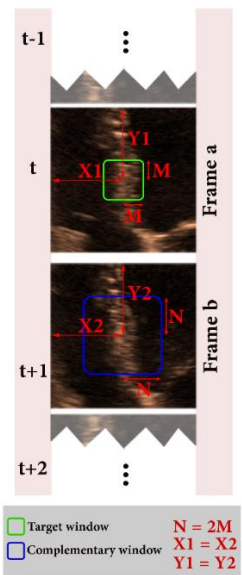


Figure 1: The target and complementary windows in two consecutive frames. It should be noted that the windows are shown relatively larger for presentation purposes.

A rectified linear unit is used as the activation function for the hidden layers. A sigmoid activation function is used for the output layer to have the occurrence probability of two classes. For each convolutional layer, a dropout rate of 0.3 is considered to minimize the overfitting effect. Based on the benchmark presented in [31], the Adam optimizer with a learning rate of 0.001 is an optimal choice for the network.

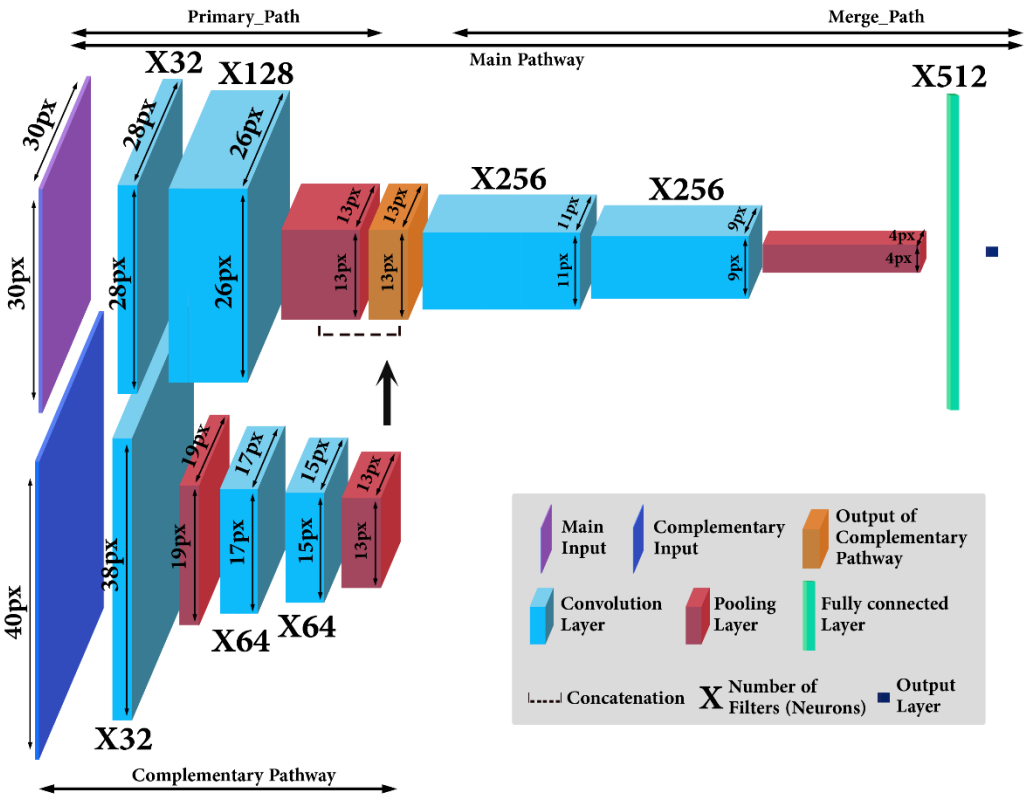


Figure 2: The target and complementary windows in two consecutive frames.

## 4.0 SETUP AND EXPERIMENTS

### 4.1 Data acquisition

Deep learning needs a huge amount of data to work [32]. In order to provide this study with a sufficient amount of data, echocardiographic sequences of 65 people were obtained with the help of expert cardiologists and in collaboration with Rajaie Cardiovascular Medical

& Research Center (RCMRC), Tehran, Iran. The subjects included males and females with both normal and ischemic cardiac conditions. Written informed consent for gathering this dataset was taken from all the subjects, and the study was approved by the ethics committee of RCMRC. The echocardiography was performed in all the three views of the apical 4-chamber, apical long-axis and parasternal short-axis for each subject. A clinical echocardiography ultrasound system (GE Vivid 7, GE Medical System, Milwaukee, Wisconsin, USA) with a 1.7 MHz phased array probe was used for this purpose. The sequences were captured with different frame rates between 42 and 68 with the size of 640×480 pixels. Based on the physical condition of each subject, different depths from 13 to 22 cm were considered for the echocardiography.

### 4.2 Dataset Preparation

A dataset including paired patches for main and complementary input gates was needed to feed and train the proposed network. Therefore, an average of 10 patches with different sizes ranging from 12×12 to 48×48 pixels were randomly extracted from each frame, except the last one, of all the sequences. Then, for each extracted target patch, its complimentary patch from the next frame was extracted with the same centre. As explained in Section 3, the size of the complementary patch in frame b was considered twice the size of its associated target patch in frame a. However, to feed the network, the patches were eventually resized to the dimensions of the corresponding input layer, i.e. 30×30 and 40×40 pixels for the main and complementary pathways. The most important part was manually labelling the paired patches as trackable-speckle or non-trackable. For this purpose, three experts in echocardiography performed annotating tasks based on the criterion classification explained in Section 2.1. Then, the consensus of the three experts was considered as the final labels.

Overall, the dataset was prepared with 15000, 3500 and 11000 (total 29500) samples for training, validation, and testing. The category distribution in each dataset was considered roughly balanced between the two classes. In order to have a reliable evaluation, patches for training, validation and testing were extracted from completely different subjects. Several samples from a total of 29500 extracted patches are shown in Figure 3. Subplots (a) and (b) display trackable-

speckle or non-trackable pattern samples that are considered as the target patches. Subplots (c) and (d) present the corresponding complementary patches extracted from the next frame for the samples in subplots (a) and (b), respectively.

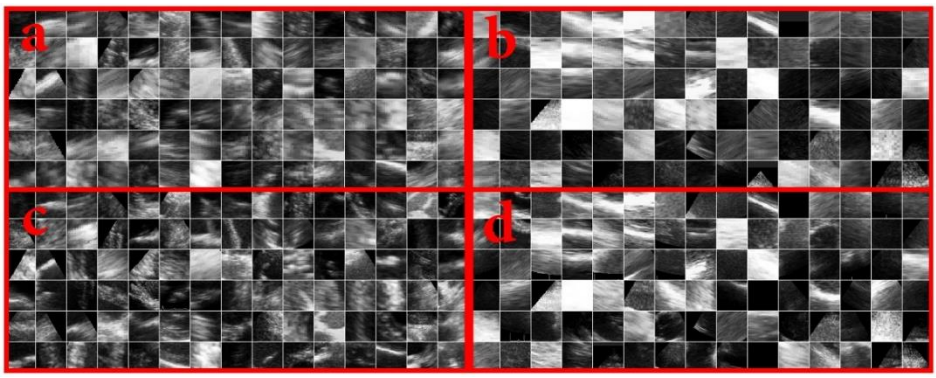


Figure 3: A few samples from a total of 29500 extracted patches. (a) and (b) display trackable-speckle and non-trackable samples. (c) and (d) present the corresponding complementary patches for the samples in subplots (a) and (b), respectively.

### 4.3 Implementation

SpeckleDNet was implemented in Python 3 using TensorFlow 2.0 and Keras framework. The source code is available for public usage at https://github.com/mitechworld/SpeckleDNet. A system with an Intel Core i7 CPU and 8 gigabytes of RAM was used to conduct the experiments. To increase the training speed, multicore processing on the CPU was also employed.

### 4.4 Metrics

In this study, categorizing a window as a trackable-speckle is considered a positive output. Thus, if a window truly contains a trackable-speckle pattern, it counts as a True Positive (TP) prediction. When the window has a non-trackable pattern but is categorized as a trackable-speckle, it should be considered a False Positive (FP) prediction. On the contrary, True Negative (TN)/False Negative (FN) represents the number of observations predicted correctly/wrongly as non-trackable. In speckle tracking, it is important to track only the windows with the trackable-speckle pattern even if the price is to ignore some of them. In other words, although the model should

correctly identify trackable-speckle windows as much as possible (decreasing the FN number), it is not preferred to tag a non-trackable window as a trackable-speckle (decreasing the FP number) because it eventually leads to a poor tracking result.

On the one hand, a metric is desired to demonstrate how much a model is good in assigning trackable-speckle labels only to those windows that are, in fact, with a trackable-speckle pattern. This can be measured by Precision which is formulated as (3). On the other hand, it is necessary to measure the general ability of a model in finding the trackable-speckle windows. Recall is a common metric for this purpose and is calculated based on (4). The optimal values for these metrics are achievable by changing the probability threshold of the classifier and plotting the precision-recall curve.

Overall, in order to evaluate a model with regard to both Precision and Recall, two popular measures are F1-score (5) and the area under the precision-recall curve (AUPRC). F1 measures the ability of a model for a specific probability threshold, whereas AUPRC summarizes the model's ability across thresholds.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1 = 2\frac{Precision.Recall}{Precision + Recall} \tag{5}$$

### 4.5 The procedure of tuning CNN Parameters

To find the optimal parameters for the network and achieve the proposed model, i.e. SpeckleDNet, several network structures through grid search were carefully trained and tested on the train and validation datasets, respectively. Grid search is a hyper-parameter tuning method in which different parameters are considered to find the optimal combination of values for the network. All the networks were compared based on their AUPRC score for tuning the parameters. As discussed in Section 3, the proposed network has a dual-path structure that includes the main and complementary pathways. Therefore, the parameters were first tuned for the main pathway alone, and then they

were adjusted for the complementary pathway considering the configured main pathway.

In the first series of experiments, a single-path structure was the subject of inquiry to find the optimal dimensions of the input layer, the number of layers and filters for the main path. Table 2 contains three networks with different numbers of layers where w is a coefficient for the number of filters. Accordingly, different numbers of filters were generated by assigning selected values of 10, 16 and 22 to the coefficient w. After finding the structure with the optional number of layers and filters, dimensions between 14×14 and 46×46 were tested as the input size for the elected structure.

Table 2: Three candidate structures for the main pathway

| Number of convolution layers | Structure |
|---|---|
| 3 | Conv(3× 3)× 2w,<br>Conv(3×3)×4w,<br>MaxPooling(2×2),<br>Conv(3×3)×16w,<br>FC×32w |
| 4 | Conv(3× 3)× 2w,<br>Conv(3×3)×4w,<br>MaxPooling(2×2),<br>Conv(3×3)×8w,<br>Conv(3×3)×16w,<br>MaxPooling(2×2),<br>FC×32w |
| 5 | Conv(3× 3)× 2w,<br>Conv(3×3)×4w,<br>MaxPooling(2×2),<br>Conv(3×3)×8w,<br>Conv(3×3)×8w,<br>MaxPooling(2×2),<br>Conv(3×3)×16w,<br>MaxPooling(2×2),<br>FC×32w |

Note: w is the coefficient for the number of filters.

The second series of experiments concerned the complementary path with regard to the chosen main pathway. As shown in Table 3, three structures with a different number of layers were examined to find the optimal combination of convolution and pooling layers. Considering that the output dimensions of the complementary pathway should be the same as the primary pathway's, the input size for a particular structure was fixed. Similar to the first series of

experiments, coefficient w was replaced by 10, 16 and 22 to generate different numbers of filters. The results of these experiments are presented in Section 5.1.

Table 3. Three candidate structures for the complementary pathway

| Number of convolution layers | Structure |
|---|---|
| 2 | Conv(3× 3)× 2w,<br>MaxPooling(2×2),<br>Conv(3×3)×4w |
| 4 | Conv(3× 3)× 2w,<br>MaxPooling(2×2),<br>Conv(3×3)×4w,<br>Conv(3×3)×4w,<br>Conv(3×3)×4w |
| 6 | Conv(3× 3)× 2w,<br>MaxPooling(2×2),<br>Conv(3×3)×4w,<br>Conv(3×3)×4w,<br>Conv(3×3)×4w,<br>Conv(3×3)×4w,<br>Conv(3×3)×4w |

Note: w is the coefficient for the number of filters.

### 4.6 Comparing SpeckleDNet with a conventional approach

A set of experiments were conducted to compare the classification results of SpeckleDNet and the conventional approaches (which consist of two parts of feature extraction and classification). As explained in Section 2.2, a combination of sixteen features and three classifiers were selected, which are used as the conventional method. In order to have a fair comparison, the conventional methods were evaluated on the same dataset as for SpeckleDNet. All the methods were compared based on Precision, Recall and F1 scores.

### 4.7 Evaluation of SpeckleDNet in speckle tracking

In this set of experiments, the goal was to investigate the impact of the proposed speckle detection method in speckle tracking and compare it with others. For this purpose, a synthetic database generated and explained by [33,34] was used. This dataset consists of 8 synthetic ultrasound sequences. The simulated sequences appear similar to real ultrasound recordings, yet, the myocardial motion is fully controlled

by the electromechanical (E/M) model in [35]. Therefore, it is a feasible alternative for the evaluation of different speckle tracking applications because we can calculate the accuracy of displacement estimation based on the ground truth data. In fact, in the simulated model, we have access to the nodes and their positions in each frame. In order to evaluate speckle tracking accuracy, a window around a node is first selected and then tracked over the consecutive frames. The error between the estimated position and the ground truth position shows the accuracy of speckle tracking. However, as explained earlier, not every node is suitable for tracking, and it could lead to a miscalculation of displacements. Therefore, a set of suitable nodes for tracking were chosen using the proposed speckle detection network, and only those nodes were tracked. The overall error was compared with the case in which the same number of nodes were chosen randomly for tracking. To achieve a more reliable conclusion, the process was repeated with a different number of nodes, and the errors were averaged at the end. The results are explained in section 5.

## 5.0 RESULTS AND DISCUSSION

### 5.1 Choosing the parameters of SpeckleDNet

Based on the procedure in Section 4.5, several network structures were carefully tested to find the best hyper-parameters for the proposed SpeckleDNet. The result of trying different numbers of layers and filters for the main path is shown in Figure 4, from which it is understandable that increasing the number of layers and filters did not necessarily lead to better performance. Based on the maximum AUPRC of 0.963, the 4-layer architecture with the filter coefficient w=16 was selected for the main path. For the selected structure, the AUPRC scores of 0.947, 0.958, 0.964, 0.955 and 0.949 were recorded for different dimensions of 14×14, 22×22, 30×30, 38×38 and 46×46 pixels, respectively. The result illustrates that enlarging the input size might cause an improvement but only to a certain point after which the performance decreases. Therefore, the dimensions of 30×30 pixels with a maximum score of 0.964 were considered the optimal window size for the main pathway input.

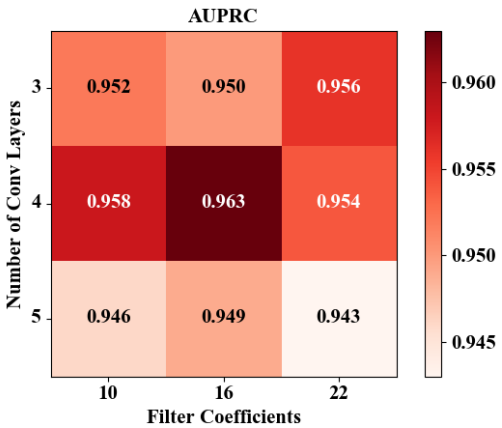


Figure 4: The result of trying different numbers of layers and filters for the main path.

Similarly, the best configuration for the complementary path was selected from nine candidate networks whose AUPRC scores are shown in Figure 5. The structure with four layers and the filter coefficient w=10 achieved the maximum score of 0.978. Based on the selected structure and the desired output size, there was no choice but to consider the input size as 40×40 pixels for the complementary pathway. With this configuration, the total number of trainable parameters for the proposed network reached 3,063,489.

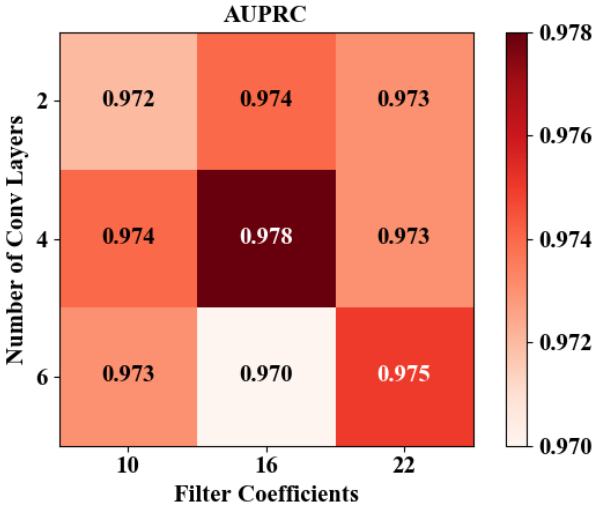


Figure 5: The result of trying a different number of layers and filters for the complementary path.

Comparing the results of the single-path structure with the dual-path structure reveals that using the complementary pathway led to an increase in the AUPRC score. It demonstrates that using the information in the next frame was beneficial for trackable speckle detection.

### 5.2 Performance of SpeckleDNet

The proposed method in this study for trackable speckle detection was compared with three common methods in terms of precision, recall, F1-score and accuracy. The results are presented in Table 4. It is obvious that the proposed CNN network with the F1-socre of 0.9484 outperforms the random forest method, which earned the best F1-score (0.8910) in comparison to KNN (0.8591) and AdaBoost (0.8769). The same conclusion can be made based on the accuracy score as well. SpeckleDNet achieved an accuracy of 94.60% in trackable speckle detection while the other three methods achieved the maximum accuracy of 88.56%.

Table 4. Comparison between the proposed method and three common methods for trackable speckle detection

| | First-order and second-order features + | | | |
|---|---|---|---|---|
| | KNN | Random forest | AdaBoost | **SpeckleDNet** |
| Precision | 0.8121 | 0.8718 | 0.8535 | **0.9316** |
| Recall | 0.9119 | 0.9110 | 0.9016 | **0.9658** |
| F1 score | 0.8591 | 0.8910 | 0.8769 | **0.9484** |
| Accuracy | 0.8465 | 0.8856 | 0.8701 | **0.9460** |

### 5.3 Performance of SpeckleDNet

The averaged error of displacement estimation was calculated as 7.16 mm ± 1.02 mm (mean error ± std) for the case in which all the nodes were selected by our proposed method. In comparison, for randomly selected nodes, it was calculated as 9.49 mm ±0.49 mm. The p-value, calculated as 0.00066 ("0.05), shows that the performance is significant.

## 6.0 CONCLUSION

This study introduces a customized convolutional neural network with a dual-path structure named SpeckleDNet for speckle detection. The

network was trained and evaluated on a dataset with 29500 ultrasound image patches of trackable-speckle or non-trackable patterns. The results showed that the proposed CNN-based method performs considerably better than the existing methods. The F1-score and accuracy of SpeckleDNet earned the maximum values of 0.9484 and 94.60% among all the methods. So, it is recommended that SpeckleDNet be used instead of the conventional methods to achieve a more accurate trackable speckle detection system.

## 7.0 REFERENCES

[1] C Moran, Carmel M., and Adrian JW Thomson. "Preclinical ultrasound imaging—A review of techniques and imaging applications." Frontiers in Physics 8 (2020). DOI 10.3389/fphy.2020.00124

[2] Huang, Q., Zeng, Z.: A review on real-time 3d ultrasound imaging technology. Biomed Res Int 2017, 6027029 (2017). DOI 10.1155/2017/6027029.

[3] Rix, A., Lederle, W., Theek, B., Lammers, T., Moonen, C., Schmitz, G., Kiessling, F.: Advanced ultrasound technologies for diagnosis and therapy. J Nucl Med 59(5), 740–746 (2018). DOI 10.2967/jnumed.117.200030.

[4] Perperidis, A.: Postprocessing approaches for the improvement of cardiac ultrasound b-mode images: A review. IEEE Trans Ultrason Ferroelectr Freq Control 63(3), 470–85 (2016). DOI 10.1109/TUFFC.2016.2526670.

[5] Aubert R, Venner C, Huttin O, Haine D, Filippetti L, Guillaumot A, Mandry D, Marie PY, Juilliere Y, Chabot F, Chaouat A. Three-dimensional echocardiography for the assessment of right ventriculo-arterial coupling. Journal of the American Society of Echocardiography, 31(8), 905-15 (2018). DOI 10.1016/j.echo.2018.04.013

[6] Cameli, M., Mondillo, S., Solari, M., Righini, F.M., Andrei, V., Contaldi, C., De Marco, E., Di Mauro, M., Esposito, R., Gallina, S., Montisci, R., Rossi, A., Galderisi, M., Nistri, S., Agricola, E., Mele, D.: Echocardiographic assessment of left ventricular systolic function: from ejection fraction to torsion. Heart Fail Rev 21(1), 77–94 (2016). DOI 10.1007/s10741-015-9521-8.

[7] Shiri, M., Gifani, P., Behnam, H., Sani, Z.A., Yeganegi, H., Shojaeifard, M.: A color-encoded map to facilitate the identification of abnormal segments of the left ventricle by novice examiners. In: 2019 27th Iranian Conference on Electrical Engineering (ICEE), pp. 1737–1741 (2019). DOI

10.1109/IranianCEE.2019.8786695

[8] Amzulescu, M.S., De Craene, M., Langet, H., Pasquet, A., Vancraeynest, D., Pouleur, A.C., Vanoverschelde, J.L., Gerber, B.L.: Myocardial strain imaging: review of general principles, validation, and sources of discrepancies. Eur Heart J Cardiovasc Imaging 20(6), 605–619 (2019). DOI 10.1093/ehjci/jez041.

[9] Joos, P., Poree, J., Liebgott, H., Vray, D., Baudet, M., Faurie, J., Tournoux, F., Cloutier, G., Nicolas, B., Garcia, D., Baudet, M., Tournoux, F., Joos, P., Poree, J., Cloutier, G.,Liebgott, H., Faurie, J., Vray, D., Nicolas, B., Garcia, D.: High-frame-rate speckle-tracking echocardiography. IEEE Trans Ultrason Ferroelectr Freq Control 65(5), 720–728 (2018). DOI 10.1109/TUFFC.2018.2809553.

[10] Bansal, M., Kasliwal, R.R.: How do i do it? speckle-tracking echocardiography. Indian Heart J 65(1), 117–23 (2013). DOI 10.1016/j.ihj.2012.12.004.

[11] Damerjian, V., Tankyevych, O., Souag, N., Petit, E.: Speckle characterization methods in ultrasound images – a review. Irbm 35(4), 202–213 (2014). DOI 10.1016/j.irbm.2014.05.003

[12] Leitman, M., Lysyansky, P., Sidenko, S., Shir, V., Peleg, E., Binenbaum, M., Kaluski, E., Krakover, R., Vered, Z.: Two-dimensional strain-a novel software for real-time quantitative echocardiographic assessment of myocardial function. J Am Soc Echocardiogr 17(10), 1021–9 (2004). DOI 10.1016/j.echo.2004.06.019.

[13] Muraru, D., Niero, A., Rodriguez-Zanella, H., Cherata, D., Badano, L.: Three-dimensional speckle-tracking echocardiography: benefits and limitations of integrating myocardial mechanics with three-dimensional imaging. Cardiovasc Diagn Ther 8(1), 101–117 (2018). DOI 10.21037/cdt.2017.06.01.

[14] Widynski, N., G'eraud, T., Garcia, D.: Speckle spot detection in ultrasound images: Application to speckle reduction and speckle tracking. In: 2014 IEEE International Ultrasonics Symposium, pp. 1734–1737 (2014). DOI 10.1109/ULTSYM.2014.0430

[15] Prager, R.W., Gee, A.H., Treece, G.M., Berman, L.H.: Analysis of speckle in ultrasound images using fractional order statistics and the homodyned k-distribution. Ultrasonics 40(1-8), 133–137 (2002). DOI 10.1016/s0041-624x(02)00104-x

[16] Marti, R., Marti, J., Freixenet, J., Zwiggelaar, R., Vilanova, J.C., Barcelo, J.: Optimally discriminant moments for speckle detection in real b-scan images. Ultrasonics 48(3), 169–81 (2008). DOI 10.1016/j.ultras.2007.11.010.

[17] Azar, A.A., Rivaz, H., Boctor, E.: Speckle detection in ultrasonic images using unsupervised clustering techniques. Conf Proc IEEE Eng Med Biol Soc 2011, 8098–101 (2011). DOI 10.1109/IEMBS.2011.6091997.

[18] Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics SMC-3(6), 610–621 (1973). DOI 10.1109/tsmc.1973.4309314

[19] Wagner, R.F., Smith, S.W., Sandrik, J.M., Lopez, H.: Statistics of speckle in ultrasound b-scans. IEEE Transactions on Sonics and Ultrasonics 30(3), 156–163 (1983). DOI 10.1109/t-su.1983.31404

[20] Wagner, R.F., Insana, M.F., Brown, D.G.: Unified approach to the detection and classification of speckle texture in diagnostic ultrasound. Opt Eng 25(6), 738–742 (1986). DOI 10.1117/12.7973899.

[21] Carmo, B.S., Prager, R.W., Gee, A.H., Berman, L.H.: Speckle detection for 3d ultrasound. Ultrasonics 40(1-8), 129–132 (2002). DOI 10.1016/s0041-624x(02)00101-4

[22] Brynolfsson, P., Nilsson, D., Torheim, T., Asklund, T., Karlsson, C.T., Trygg, J., Nyholm, T., Garpebring, A.: Haralick texture features from apparent diffusion coefficient (adc) mri images depend on imaging and pre-processing parameters. Sci Rep 7(1), 4041 (2017). DOI 10.1038/s41598-017-04151-4.

[23] Abdel-Nasser, M., Omer, O.A.: Ultrasound image enhancement using a deep learning architecture. In: A.E. Hassanien, K. Shaalan, T. Gaber, A.T. Azar, M.F. Tolba (eds.) Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016, pp. 639–649. Springer International Publishing (2016). DOI 10.1007/978-3-319-48308-561

[24] Hyun, D., Brickson, L.L., Looby, K.T., Dahl, J.J.: Beamforming and speckle reduction using neural networks. IEEE Trans Ultrason Ferroelectr Freq Control 66(5), 898–910 (2019). DOI 10.1109/TUFFC.2019.2903795.

[25] Dietrichson, F., Smistad, E., Ostvik, A., Lovstakken, L.: Ultrasound speckle reduction using generative adversial networks. In: 2018 IEEE International Ultrasonics Symposium (IUS), pp. 1–4 (2018). DOI 10.1109/ULTSYM.2018.8579764

[26] Joel, T., Sivakumar, R.: An extensive review on despeckling of medical ultrasound images using various transformation techniques. Applied Acoustics 138, 18–27 (2018). DOI 10.1016/j.apacoust.2018.03.023

[27] Badano, L.P., Kolias, T.J., Muraru, D., Abraham, T.P., Aurigemma, G., Edvardsen, T., D'Hooge, J., Donal, E., Fraser, A.G., Marwick, T. and

Mertens, L. Standardization of left atrial, right ventricular, and right atrial deformation imaging using two-dimensional speckle tracking echocardiography: a consensus document of the EACVI/ASE/Industry Task Force to standardize deformation imaging. European Heart Journal-Cardiovascular Imaging, 19(6), pp.591-600 (2018). DOI 10.1093/ehjci/jey042

[28] Yaakob, R., Aryanfar, A., Halin, A.A., Sulaiman, N.: A comparison of different block matching algorithms for motion estimation. Procedia Technology 11, 199–205 (2013). DOI 10.1016/j.protcy.2013.12.181

[29] Molinari, F., Caresio, C., Acharya, U.R., Mookiah, M.R., Minetto, M.A.: Advances in quantitative muscle ultrasonography using texture analysis of ultrasound images. Ultrasound Med Biol 41(9), 2520–32 (2015). DOI 10.1016/j.ultrasmedbio.2015.04.021.

[30] Baek, J., Poul, S.S., Swanson, T.A., Tuthill, T. and Parker, K.J. Scattering signatures of normal versus abnormal livers with support vector machine classification. Ultrasound in Medicine & Biology, 46(12), pp.3379-3392 (2020). DOI 10.3390/e22050567

[31] Schneider, F., Balles, L., Hennig, P.: Deepobs: A deep learning optimizer benchmark suite. arXiv preprint arXiv:1903.05499 (2019)

[32] Razzak, M.I., Naz, S., Zaib, A.: Deep Learning for Medical Image Processing: Overview, Challenges and the Future, pp. 323–350. Springer International Publishing, Cham (2018). DOI 10.1007/978-3-319-65981-7 12. URL https://doi.org/10. 1007/978-3-319-65981-7_12

[33] Alessandrini, M., De Craene, M., Bernard, O.: A pipeline for the generation of realistic 3d synthetic echocardiographic sequences: Methodology and open-access database. IEEE Trans Med Imaging 34(7), 1436–1451 (2015). DOI 10.1109/TMI.2015.2396632

[34] Alessandrini, M., Heyde, B., Queiros, S.: Detailed evaluation of five 3d speckle tracking algorithms using synthetic echocardiographic recordings. IEEE Trans Med Imaging 35(8), 1915–1926 (2016). DOI 10.1109/TMI.2016.2537848

[35] Marchesseau, S., Delingette, H., Sermesant, M., Sorine, M., Rhode, K., Duckett, S., Rinaldi, C., Razavi, R., Ayache, N.: Preliminary specificity study of the bestel–cl'ement–sorine electromechanical model of the heart using parameter calibration from medical images. Journal of the Mechanical Behavior of Biomedical Materials 20, 259 – 271 (2013). DOI https://doi.org/10.1016/j.jmbbm.2012.11.021.