A Pilot Study on Convolutional Neural Networks for Motion Estimation from Ultrasound Images

Ewan Evain, Khuram Faraz, Thomas Grenier, Damien Garcia, Mathieu De Craene and Olivier Bernard

***Abstract*—In recent years, deep learning has been successfully applied to the analysis and processing of ultrasound images. To date, most of this research has focused on segmentation and view recognition. This paper benchmarks different convolutional neural network algorithms for motion estimation in ultrasound imaging. We evaluated and compared several networks derived from FlowNet2, one of the most efficient architectures in computer vision. The networks were tested with and without transfer learning and the best configuration was compared against the particle-imaging-velocimetry method, a popular state-of-the-art block-matching algorithm. Rotations are known to be difficult to track from ultrasound images due to a significant speckle decorrelation. We thus focused on images of rotating disks, that could be tracked through speckle features only. Our database consisted of synthetic and *in-vitro* B-mode images after log-compression, and covered a large range of rotational speeds. One of the FlowNet2 sub-networks, FlowNet2SD, produced competitive results with a motion field error smaller than 1 pixel on real data after transfer learning based on simulated data. These errors remains small for a large velocity range without the need for hyper-parameter tuning, which indicates the high potential and adaptability of deep learning solutions to motion estimation in ultrasound imaging.**

***Index Terms*—Motion estimation, deep learning, ultrasound.**

I. Introduction

THE comparison of state-of-the-art image processing algorithms with approaches relying on Deep Learning (DL) is a fast growing topic in ultrasound image analysis. In recent years, DL has been shown to outperform traditional approaches for a wide spectrum of image analysis tasks, from the recognition and evaluation of standard acquisition views [1], [2], to the segmentation [3], [4] and reconstruction of echocardiographic images, in both 2D [5] and 3D [6]. When it comes to DL-based motion estimation from images in the computer vision community, various approaches have been investigated based on supervised or unsupervised learning. Supervised learning techniques usually rely on synthetic data with a reference motion field, while unsupervised techniques generally involve intensity-based losses. In the latter, the loss is computed from a pair of images that are warped according to the estimated displacement field [7]–[9]. In a recent study, Ilg *et al.* [10] reviewed and benchmarked a large set of DL-based motion estimators (see references there in). This study revealed that the FlowNet2 architecture [10] had excellent performance compared with other state-of-the-art algorithms while maintaining a low inference time. FlowNet2 combines several networks based on the U-Net architecture [11], a common choice for image segmentation, showing potential to solve image tracking problems. All these aspects led us to take FlowNet2 as a starting point for this study, and to investigate how the performance of this network generalizes when applied to ultrasound image sequences. Limiting the scope to DL-based motion estimators in ultrasound leaves a very small number of related studies. In [12], FlowNet2 was embedded with its pre-trained weights (as provided in [10]) to estimate coarse displacements in the context of elastography. Similarly, FlowNet2 was applied as such in [2] with a view classification and semantic partitioning of the mycoardium for quantifying longitudinal deformation. In [13], the authors used the first branch of the FlowNet2 architecture with transfer learning from a simulated dataset to retrieve displacements in ultrasound breast imaging for elastography. While this pioneer study reveals the value of adapting deep learning solutions for motion estimation in ultrasound, it is limited to the direct use of a branch of an existing network without any evaluation on the architecture for the targeted application. Moreover, this study focused on estimating strain from relatively small displacements, where the decorrelation of speckle is limited. Based on this literature review, the purpose of this paper is to answer the following three questions:

1) How do different CNN architectures compare on a given set of echo images in terms of motion accuracy?
2) How does it compare to non-DL algorithms that have been designed for ultrasound?
3) What is the gain brought by transfer learning, *i.e.* by re-training the weights of a network already pre-trained on natural (video) scene sequences.

For that purpose, we focused on synthetic and *in-vitro* datasets involving controlled motion fields (in our case rotations) on a simplified geometry - a disk. Assessing accuracy on these images before and after transfer learning on simulated data, for a number of FlowNet2-based networks, allowed for an accurate quantitative comparison of these networks. Moreover, we decided to focus on rigid rotations as these motions are important sources of speckle decorrelation in ultrasound, making it particularly difficult to estimate actual motion from apparent displacement (*i.e.* motion measured from the image itself). Finally, although our dataset involves simpler motion fields and less realistic images than in synthetic echocardiography of [14], our solution has the advantage of providing a reference displacement field over the entire image domain for training DL-based motion estimators. This ensures that the learning phase will not be biased by a segmentation pre-processing step designed to delineate the region of interest. The resulting

E. Evain, K. Faraz, T. Grenier, D. Garcia and O. Bernard are with the Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne, CNRS, Inserm, CREATIS UMR 5220, U1206, F-69621, LYON, France. E-mail: olivier.bernard@creatis.insa-lyon.fr.

E. Evain and M. De Craene are with Philips Research Paris (Medisys), Suresnes, France

network is thus self-sufficient and can accurately detect motion over the entire image domain.

II. Evaluated networks

FlowNet2 relies on the design of the FlowNet architecture published by the same authors [15]. In particular, FlowNet2 implements a chain of CNNs, mainly combining a set of two U-Net-like architectures [11]: FlowNetC and FlowNetS. Unlike FlowNetS, which involves a classical U-Net architecture, FlowNetC adds a correlation layer to mimic traditional image registration techniques more explicitly. These two networks are based on: *i)* a contraction part to retrieve features in a low-dimensional space; *ii)* an expansion part for projecting back this information at the resolution of the input image; *iii)* a set of skip connections to keep more information from the contractive part and avoid the vanishing gradient problem, conforming to the standard U-Net design [11]. In the expanding part, the upsampling scheme is stopped when the size of the feature maps corresponds to a quarter of the full resolution, opting for a bilinear interpolation to generate the final deformation field. This choice was motivated by the authors after observing the low benefit (in terms of accuracy) of reaching the full resolution for the last layers, and by limiting the number of parameters to be optimized. In total, FlowNet2 is composed of five CNNs divided into two parallel branches dedicated to low and large displacements, the outputs of which are merged through a third branch (named as merging branch in the following), as illustrated in Fig. 1. The branch that handles large displacement concatenates three networks, namely one FlowNetC and two FlowNetS CNNs. The branch in charge of small displacements includes a simple CNN, named FlowNetSD, with the same architecture as FlowNetS but with a few more convolution layers and smaller kernel size (3x3 instead of 7x7 and 5x5) and stride (1 instead of 2). These two branches have a pair of two consecutive color images as input, and provide as output the corresponding estimated motion field per pixel. Finally, the FlowNetS networks involved in the large displacement branch take as inputs: *i)* the displacement field estimated at the previous step; *ii)* the same pair of input images after warping one of them by the current displacement estimate; *iii)* the brightness error between the pair of modified images. This strategy allows to address large displacements in a multi-scale approach.

In this study, we investigated the performance of FlowNet2 in estimating displacements in ultrasound imaging, and that of each individual CNN involved in this architecture, namely FlowNetC, FlowNetS and FlowNetSD. Recently, Cai *et al.* used the FlowNet-SD architecture for the estimation of particle displacements in velocimetry imaging [16]. They showed that it was possible to improve the overall accuracy of this network by replacing the bilinear interpolation at the end of the expansion part with two additional upsampling layers, leading to a more classical U-Net-like architecture. This was justified by the fact that small displacements, especially sub-pixel displacements, would not be sufficiently addressed by bilinear interpolation. Inspired by this work, we also investigated the influence of replacing the bilinear interpolation with two upsampling layers for the FlowNetS and FlowNetSD architectures, leading to two modified networks referred to as FlowNetS* and FlowNetSD* in the following. We thus investigated the performance of 6 different networks for estimating motion in ultrasound imaging: FlowNet2, FlowNetC, FlowNetS, FlowNetS*, FlowNetSD and FlowNetSD*.

III. Simulated & In-Vitro dataset

All networks evaluated in this paper were trained in a supervised manner, which required setting up an ultrasound dataset with reference motion fields. To this end, we created a dataset consisting of synthetic and *in-vitro* data with known motion. We used B-mode images after scan conversion and log-compression since, in practice, they are the only data that can be retrieved from clinical ultrasound scanners and they require low storage compared with RF data. In particular, we worked on a spinning disk scenario, where the amount of displacement is well-controlled, both in simulations and *in-vitro* experiments. This strategy allowed us to design simulated and real image sequences with similar displacements and image intensities. It also targets a well-known challenge in the field of ultrasound, as it may be difficult to recover accurate rotations due to the greater speckle decorrelation it induces compared with translations. Working with a combination of simulated and *in-vitro* data allowed us to assess accuracy and robustness. Regarding accuracy, the simulated data showed the value of transfer learning for specializing the different networks to ultrasound, as described in Sec. IV. Regarding robustness, the *in-vitro* data allowed us to evaluate how several networks that were trained on synthetic data performed on real ultrasound images. Examples of *in-vitro* and simulated images, with the corresponding reference motion fields are provided in Fig. 2.

A. In-Vitro Data

We re-processed the *in-vitro* data described in [17]. These images were acquired with a Verasonics research scanner (V-1-128, Verasonics Inc., Redmond,WA) and a 2.5 MHz phased-array transducer (ATL P4–2, 64 elements) on a agar-based disk phantom with incremental angular velocities. 32 diverging waves with a triangle steering strategy were emitted to reconstruct one single image based on a dedicated delay and sum technique. The disk had four anechoic cysts positioned symmetrically with respect to its center, as shown in Fig. 2-b. Each sequence was composed of 293 frames with angular velocities from 1 to 5 rad/s. From the B-mode images delivered into a polar coordinate system, we reconstructed all the B-mode images into a Cartesian coordinate system on a uniform grid 441x321 with a pixel area of 0.45 mm^2. Each sequence was reconstructed with a frame rate of 312 Hz. In our experiments, we used pairs of images separated by 6 frames in order to work with a frame rate of 52 Hz. This temporal and spatial imaging resolutions are similar to the ones classically used in clinical echocardiography practice. Based on the range of the angular velocities, this meant estimating displacements between 0 (at the center of the disk) and 11 pixels (taking into account the spatial resolution of the grid) between two images.

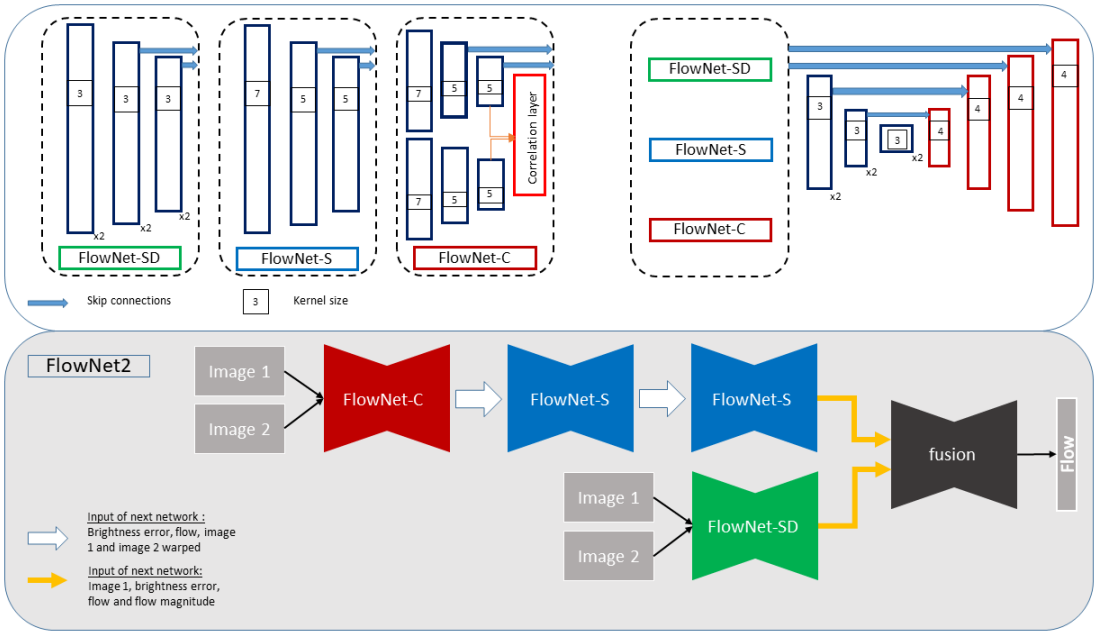
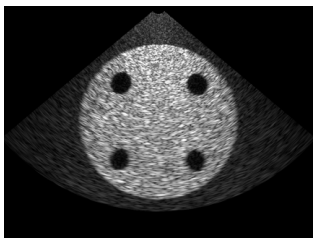
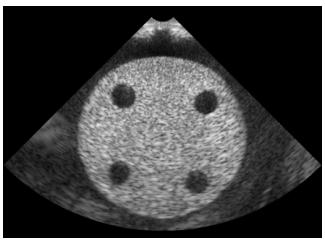


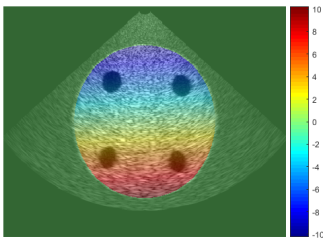
Fig. 1. Schematic view of the overall architecture of FlowNet2 (bottom) and the architecture of all the sub-networks that compose it (top). Convolution layers are shown in blue and deconvolution layers in red.



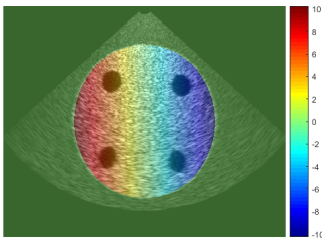
(a) Simulated data



(b) In-vitro data



(c) X component of motion field



(d) Y component of motion field

Fig. 2. Examples of simulated (a) and *in-vitro* (b) images extracted from our dataset, along with an example of reference motion field in pixels (c-d) corresponding to (a).

B. Simulated Data

Synthetic images of a spinning disk were generated using the ultrasound simulator proposed in [18]. The same acquisition protocol as the one used to acquire the *in-vitro* data was simulated. Foreground scatterers were randomly positioned inside a disk and moved at angular velocities ranging from 1 to 5 rad/s. The full synthetic dataset was composed of 14300 pairs of images with the corresponding reference displacement fields per pixel. For each angular velocity, 10 sequences of 287 images were simulated: 5 sequences with a homogeneous disk at the center of the image, and 5 including four anechoic cysts placed symmetrically with respect to the center of the disk, as illustrated in Fig. 2-a. For each sequence, background and foreground scatterers were randomly distributed on the first frame, leading to different speckle textures over the dataset. The spinning disk involved in the *in-vitro* experiment corresponds to an agar phantom immersed in water. There is therefore an intrinsic difference in terms of backscattering coefficient of the acoustic wave, and thus in the reconstructed B-mode images, between the background (water) and the object (disk). We experimentally fixed a ratio of 8 between the backscattered coefficients of the foreground and background scatterers involved in our simulations so to generate B-mode images with similar intensity histograms to the ones of the in-vitro dataset. As for the *in-vitro* data, all the simulated B-mode images were reconstructed into a Cartesian coordinate system on a uniform grid 428x321 with a pixel area of 0.47 mm^2. The frame rate was set at 312 Hz and pairs of images separated by 6 frames were used for motion estimation.

IV. Transfer Learning

This section provides details on the transfer learning strategy that we applied to specialize the networks described in Sec. II in ultrasound images. In particular, starting from the networks weights learned from several synthetic sequences provided by the FlowNet2 authors [10], we launched a new learning procedure independently for each network based exclusively on the 14300 synthetic ultrasound data described in Sec. III-B.

A. Loss functions

To perform transfer learning on the FlowNet2 architecture, we followed the recommendations of [10]. From the network

already trained from several synthetic sequences, we restricted the update procedure to the weights belonging to the merging branch during the learning phase (fusion network displayed in the bottom part of Fig. 1). For this purpose, the L_{pq} loss function with $p = 2$ and $q = 0.2$ was used, as suggested in [10]. To perform transfer learning on the remaining networks described in Sec. II, we opted for the EndPoint Error (EPE) as loss. EPE is a standard metric used in optical flow and is defined as the L_2 norm of the difference between the estimated flow vector and the ground truth [19] (in pixels):

$$EPE = \sqrt{(u - u_{gt})^2 + (v - v_{gt})^2}, \tag{1}$$

where (u, v) stands for the estimated flow and (u_{gt}, v_{gt}) is the ground truth displacement vector. For FlowNetC, FlowNetS and FlowNetSD architectures, the actual loss function corresponded to the weighted sum of the EPE calculated at different resolution levels in the expansion part of the networks as in [10]. The values of the weights involved in this loss function were the same than those chosen experimentally in [10]. For FlowNetS* and FlowNetSD*, since two additional layers were inserted at the end of the expansion part of the network, it was necessary to adapt the loss function by adding two EPE terms computed from each new resolution. The corresponding weights were the same as those proposed in [16], thus adding more importance to the last layers.

B. Hyper-parameters

An initial learning rate of $\lambda = 1e{-}4$ was set experimentally. Moreover, as recommended in the original paper of Ilg *et al.* [10], λ was then divided by two after reaching 40% (and every 20%) of the total number of epochs. This procedure was performed to ease convergence of the optimization process. Indeed, this scheme allows for fast learning at the beginning of the training process, making gradually smaller updates over the course of the optimization to refine the weights. The Adam optimizer [20] was used in this study and a batch size of 4 pairs of images was chosen mainly for reasons of memory capacity.

C. Dataset split

The full set of the synthetic data was divided into three folds: 60% for the training set, 20% for the validation set and 20% for the test set. Moreover, each fold contained data for the entire angular velocity range (*i.e.* from 1 to 5 rad/s). The validation set was used to select the most efficient weights of the deep learning architectures during the training process, while the test set was used to produce all the results given in this paper. It is important to note that all algorithms were tested only once on the test set and that no optimization was performed on it in order to avoid overfitting. For each velocity, a balanced selection of sequences with and without anechoic cysts was realized. Data from a whole sequence were not mixed across the 3 folds. In particular, 6 full sequences were chosen for the training phase, 2 full sequences for the validation phase and 2 full sequences for the testing phase. This procedure ensured that each fold contained the same diversity of information without introducing any bias.

D. Data Augmentation

To avoid overfitting, data augmentation was realized following the procedure described in [21]. The deployed augmentation integrated some typical alterations that could happen in ultrasound, *i.e.* variations in brightness, saturation and contrast. We involved variability in terms of translation, Gaussian noise, black borders and cut-outs, as they are reported to improve the generalization of networks [22]. Concerning the additive noise, we used a Gaussian noise with small variance in order to remain close to the original images while increasing the network robustness against intensity fluctuations. We also added random cropping to increase the variability of the dataset and reduce the dimensions of the images to 384x320 pixels to respect the input image size of the different networks described in Sec. II. Finally, to ensure that the networks did not overspecialize in a single direction of rotation, pairs of images were randomly flipped with a probability of 0.5. In this way, both clockwise and counterclockwise rotations were equally included during the training process.

V. Evaluation protocol

A. Metrics

Performance of all evaluated networks was assessed through: *i)* the EPE metric described in Sec. IV-A and computed inside the disk only; *ii)* the error on the angular velocity, which was computed by dividing at each point inside the disk the estimated velocity magnitude by the distance to the disk center. We reported the distribution of each of these metrics using the median and the median absolute deviation (MAD) defined as

$$MAD = median(|X - median(X)|)\,, \tag{2}$$

where X stands for the distribution of the metric values inside the disk. MAD was preferred to the standard deviation as it is more robust to outliers.

B. State-of-the-art method

We compared all CNN networks described in Sec. II with a state-of-the-art Particle Imaging Velocimetry (PIV) method, which was used during the challenge on synthetic aperture - vector flow imaging organized during the International Ultrasonic Symposium in 2018 [23]. This comparison was made both on the test set of the simulated data and on the *in-vitro* data. PIV is a block-matching algorithm that worked with ensembles of n consecutive images, under the assumption that the motion remained unchanged during that temporal window, to calculate the average of $n - 1$ cross correlation matrices (ensemble correlation, see [24]). Peak detection of the averaged normalized cross correlation provided the displacements with a pixel precision. Subpixel precision of the displacement estimates was then obtained through parabolic peak fitting of the cross correlation. Taking into account the physical properties of the ultrasound images involved in our experiments, we applied a multiscale strategy to estimate motion by using 3 different sizes of search areas, namely 24x24, 16x16 and 12x12 pixels. These values span a search range

TABLE I
MEDIAN EPE, ESTIMATED ANGULAR VELOCITY AND MAD DISPERSION VALUES COMPUTED INSIDE THE SPINNING DISK ON THE SYNTHETIC DATASET FROM THE NETWORKS DESCRIBED IN SEC. II. FOR EACH NETWORK, DIFFERENT LEARNING STRATEGIES WERE ASSESSED (✓: TRANSFER LEARNING, - PRE-TRAINED WEIGHTS FROM NATURAL SCENE IMAGES; ✗ RANDOM INITIALIZATION). THE BEST SCORES FOR EACH CATEGORY ARE HIGHLIGHTED IN BOLD WHILE THE OVERALL BEST NETWORK IS SHADED

Methods	TL	$1rad/s$		$2rad/s$		$3rad/s$		$4rad/s$		$5rad/s$	
		EPE pixel	Velocity rad/s	EPE pixel	Velocity rad/s	EPE pixel	Velocity rad/s	EPE pixel	Velocity rad/S	EPE pixel	Velocity rad/s
FlowNet2	✓	1.4 ±0.4	0.0 ± 0.0	1.6 ± 0.8	0.2 ± 0.2	2.8 ± 1.1	0.6 ± 0.5	4.5 ±1.5	0.4 ±0.3	6.1 ±1.9	0.2 ±0.2
	-	0.2 ±0.1	0.9 ±0.1	0.5 ±0.2	1.7 ±0.1	2.0 ±0.6	1.6 ±0.2	4.3 ±1.3	1.0 ±0.4	6.2 ±2.0	0.6 ±0.4
FlowNetC	✓	5.3 ±2.6	4.0 ± 2.4	6.5 ± 3.0	4.6 ± 2.7	7.8 ± 3.4	5.2 ± 3.1	8.7 ±3.5	5.3 ±3.1	9.6 ±3.5	5.3 ±3.1
	✗	1.5 ±0.5	0.1 ±0.1	2.9 ±0.8	0.1 ±0.1	4.4 ±1.1	0.1 ±0.1	5.9 ±1.5	0.1 ±0.1	7.3 ±1.9	0.1 ±0.1
	-	1.0 ±0.3	0.4 ±0.2	2.2 ±0.6	0.6 ±0.2	3.7 ±1.0	0.7 ±0.3	5.0 ±1.4	0.8 ±0.3	6.4 ±1.8	0.8 ±0.4
FlowNetS	✓	0.2 ±0.1	0.9 ±0.1	**0.3** ±0.1	**1.9** ±0.1	0.5 ±0.2	2.7 ±0.2	1.1 ±0.4	3.2 ±0.2	2.2 ±0.6	3.4 ±0.2
	✗	1.2 ±0.5	**1.0** ±0.5	1.5 ±0.7	1.3 ±0.5	2.6 ±1.1	1.5 ±0.5	3.8 ±1.5	1.7 ±0.6	4.9 ±1.8	1.9 ±0.6
	-	1.0 ±0.3	0.6 ±0.2	1.9 ±0.6	0.9 ±0.3	3.3 ±1.0	1.0 ±0.4	4.7 ±1.5	1.2 ±0.5	6.0 ±1.8	1.4 ±0.6
FlowNetS*	✓	2.0 ±1.0	1.2 ±0.8	3.2 ±1.3	1.1 ±0.8	4.6 ±1.6	1.2 ±0.8	5.9 ±1.9	1.2 ±0.8	7.3 ±2.2	1.2 ±0.8
	✗	1.9 ±0.7	1.1 ±0.7	3.1 ±1.0	1.1 ±0.7	4.5 ±1.3	1.1 ±0.7	5.9 ±1.6	1.2 ±0.7	7.4 ±1.9	1.4 ±0.8
FlowNetSD	✓	**0.1** ±0.0	1.1 ±0.0	0.4 ±0.1	2.2 ±0.0	**0.4** ±0.1	**3.2** ±0.1	**0.3** ±0.1	**4.2** ±0.1	**0.4** ±0.1	**4.9** ±0.1
	✗	1.6 ±0.4	0.3 ±0.2	3.3 ±0.7	0.6 ±0.3	4.5 ±1.0	0.6 ±0.3	5.8 ±1.4	0.7 ±0.4	7.3 ±1.7	0.8 ±0.4
	-	0.2 ±0.1	0.9 ±0.1	0.5 ±0.2	1.8 ±0.1	1.5 ±0.5	2.1 ±0.3	3.5 ±1.0	1.7 ±0.3	5.0 ±1.4	1.7 ±0.4
FlowNetSD*	✓	**0.1** ±0.0	1.1 ±0.0	0.5 ±0.1	2.3 ±0.0	0.6 ±0.2	3.4 ±0.0	0.6 ±0.2	4.4 ±0.0	0.4 ±0.1	5.2 ±0.1
	✗	1.4 ±0.4	0.1 ±0.1	2.8 ±0.7	0.1 ±0.1	4.3 ±1.1	0.1 ±0.1	5.8 ±1.4	0.1 ±0.1	7.2 ±1.8	0.1 ±0.1

from 3 to 6 times the speckle size and were experimentally tuned to obtain the best results. We also verified that adding a larger window did not improve the results and that the multiscale strategy returned the best results regardless of the frame rate. Experimentally, we observed that PIV produced erroneous results when the displacement was either too low (average displacements lower than one pixel) or too large (average displacements higher than ten pixels) between two consecutive frames. For this reason, we applied the PIV algorithm under two different conditions: *i)* by using pairs of images corresponding to a frame rate of 52 Hz and used to train the different networks (referred to as PIV); *ii)* by reducing the number of frames that separate two images of a pair in order to adapt the frame rate for each angular velocity and taking 16 images into account to obtain the best possible results (referred to as PIV-adapt). In practice, the higher the angular velocity, the higher the optimal frame rate chosen, up to 312 Hz for a velocity of 5 rad/s. This way of proceeding gives natural bounds of PIV accuracy, as giving to PIV data at a higher temporal resolution than the evaluated CNNs gives an *upper bound* of PIV's accuracy.

VI. ACCURACY BENCHMARKS

A. Numerical simulations

1) Network Selection: We first compared the performance of all CNNs described in Sec. II on the simulated database for three different configurations: *i)* using the weights of the networks pre-trained from [10] as published by the authors (rows marked as − under the TL column in Table I); *ii)* applying transfer learning starting from the pre-trained weights and tuning them on the simulated dataset described in Sec. III-B (rows marked as ✓ under the TL column in Table I); *iii)* learning the weights from scratch using random values as initialization (rows marked as ✗ under the TL column in Table I). Note that since 2 layers were added to FlowNetS* and FlowNetSD*, random initialization of these layers was applied before making the transfer learning for these two networks. Moreover, since these two networks were not studied in the original FlowNet2 paper, there is no pre-trained weights available for these architectures.

Table I summarizes all the results obtained during this experiment. Each network was evaluated for angular velocity increments of 1 rad/s, which corresponds on average to an increase of two pixels on pairwise displacements, making the tracking task particularly challenging for high rotation values ($\geq$ 4 rad/s). From the analysis of Table I, it first appears that the different training schemes produced EPE results that varied according to the type of architecture. For instance, the best results from FlowNetC were obtained from the pre-trained weights, the application of transfer learning degrading the results independently of the angular velocity. On the contrary, the transfer learning procedure applied on

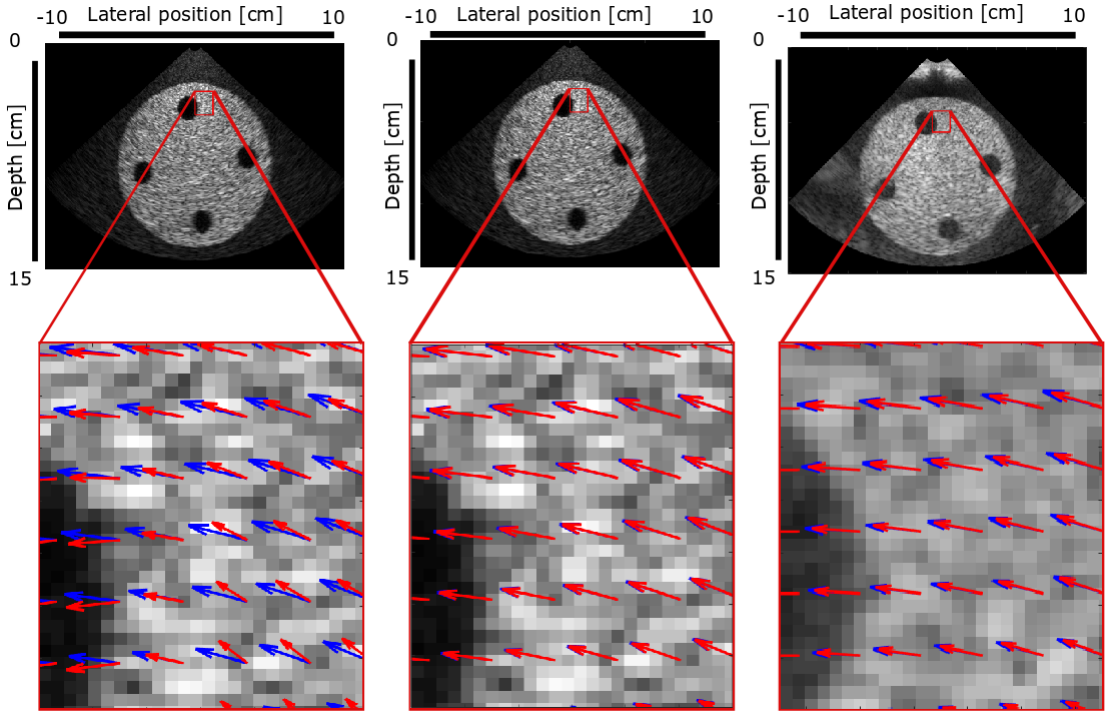


Fig. 3. Reference (blue) and estimated (red) motion fields using FlowNetSD on (left) a synthetic image without transfer learning; (middle) the same synthetic image but with transfer learning; (right) an *in-vitro* image with transfer learning. The displayed data were all extracted from a sequence with a spinning disk rotating at 1 rad/s.

FlowNetS and FlowNetSD improved significantly the quality of the estimation, the improvement being more visible with the increase in velocity. This illustrates the feasibility of adapting pre-trained networks, trained on images from another domain (*i.e.* computer vision), to the intrinsic characteristics of ultrasonic images, especially in terms of speckle decorrelation, for motion quantification.

Regarding the architecture complexity, it appears that both FlowNet2 and FlowNetSD produced the best EPE scores based on their pre-trained weights for angular velocity of 1 and 2 rad/s. However, for higher velocity values, the simpler architecture (*i.e.* FlowNetSD) obtained better results, which suggests that there is no need to increase network capacity to improve motion estimation for large displacements in ultrasonic imaging. Concerning FlowNetS* and FlowNetSD*, one can see that the insertion of two additional layers degraded the overall performance of FlowNetS and did not improve results for FlowNetSD. This reveals the uselessness of adding these two layers in the context of motion estimation in ultrasound and corroborates the choice made by FlowNet2 authors of a bilinear interpolation of the network outputs.

Based on this analysis, it appears that the FlowNetSD network with transfer learning achieved the best results in terms of median EPE for all angular velocity except at 2 rad/s, where the difference with the best network is minimal (0.1 px). It is also interesting to note the remarkably low MAD values obtained by this method, not exceeding 0.1 px. FlowNetSD also performed an accurate estimation of the angular velocity, with a maximum error of 0.2 rad/s. In addition, this method displayed similar performance across a wide range of velocities with MAD values ≤ 0.1 rad/s. An example of a motion field estimated by this technique is shown in Fig. 3. As FlowNetSD with transfer learning appeared to be the best CNN architecture to estimate motion from our simulated database of ultrasound images, we kept this network in our next experiments.

2) Comparison with PIV: Table II summarizes the comparison between FlowNetSD with transfer learning (FlowNetSD-TL) and the two PIV versions described in Sec. V-B. Regarding PIV, results for the velocity estimation were consistent with the actual true values up to 2 rad/s, along with EPE errors ≤ 0.2 px and MAD values ≤ 0.1 px. At 3 rad/s, PIV slightly underestimated the velocity with a value of 2.5 rad/s but with higher median EPE (from 0.2 px to 0.9 px) and MAD (from 0.1 px to 0.6 px) values. For higher angular velocities, PIV estimates deteriorated with EPE errors over 6.4 px and an underestimation of the velocity, revealing the limitations of this algorithm for angular velocities higher than 2 rad/s at a frame rate of 52Hz.

Nonetheless, when adapting the frame rate for each angular velocity (thus assuming the true displacement range was known for every input sequence and providing PIV with images at a higher temporal resolution than FlowNetSD), PIV-adapt results became consistent and accurate. Indeed, all estimated angular velocity values were accurate and EPE errors were found to be constant and around 0.2 px. It is interesting to note that FlowNetSD produced slightly worse results but close to the PIV-adapt method, with EPE errors $\leq$

TABLE II
MEDIAN EPE, ESTIMATED ANGULAR VELOCITY ACCURACY AND MAD DISPERSION VALUES COMPUTED INSIDE THE CENTERED SPINNING DISK ON THE SYNTHETIC DATASET (FIRST THREE ROWS) AND ON THE *in-vitro* DATASET (LAST THREE ROWS) FOR FIVE DIFFERENT ANGULAR VELOCITIES. FOR THIS EXPERIMENT, FLOWNETSD WITH TRANSFER LEARNING (FLOWNETSD-TL) WAS COMPARED WITH THE TWO VERSIONS OF THE NON-DEEP LEARNING STATE-OF-THE-ART PIV TECHNIQUE DESCRIBED IN SEC. V-B

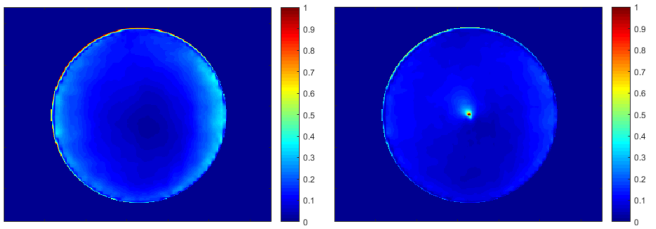
	Methods	$1 rad/s$		$2 rad/s$		$3 rad/s$		$4 rad/s$		$5 rad/s$	
		EPE pixel	Velocity rad/s	EPE pixel	Velocity rad/s	EPE pixel	Velocity rad/s	EPE pixel	Velocity rad/s	EPE pixel	Velocity rad/s
Simulated	FlowNetSD-TL	0.1 ±0.0	1.1 ± 0.0	0.4 ± 0.1	2.2 ± 0.0	0.4 ± 0.1	3.2 ± 0.1	0.3 ±0.1	4.2 ±0.1	0.4 ±0.1	4.9 ±0.1
	PIV	0.2 ±0.1	1.0 ± 0.1	0.2 ± 0.1	2.0 ± 0.1	0.9 ± 0.6	2.5 ± 0.6	3.7 ± 2.1	1.4 ± 1.1	6.4 ± 2.5	0.6 ± 1.0
	PIV-adapt	0.2 ± 0.1	1.0 ± 0.1	0.2 ± 0.1	2.0 ± 0.1	0.2 ± 0.1	2.9 ± 0.2	0.2 ± 0.1	3.9 ± 0.2	0.1 ± 0.1	4.9 ± 0.2
In-vitro	FlowNetSD-TL	0.7 ±0.2	1.3 ± 0.1	0.8 ± 0.3	2.3 ± 0.2	1.1 ± 0.4	3.4 ± 0.3	1.0 ±0.4	4.3 ±0.3	0.8 ±0.3	5.0 ±0.4
	PIV	0.4 ± 0.1	1.2 ± 0.1	0.8 ± 0.3	2.3 ± 0.4	2.3 ± 1.5	1.6 ± 1.2	4.8 ± 2.3	0.8 ± 1.1	6.5 ± 2.6	0.5 ± 1.0
	PIV-adapt	0.2 ± 0.1	1.2 ± 0.1	0.3 ± 0.1	2.4 ± 0.2	0.3 ± 0.1	3.6 ± 0.3	0.2 ± 0.1	4.5 ± 0.5	0.2 ± 0.1	5.6 ± 0.6

0.4 px and angular velocity errors $\leq$ 0.2 rad/s. However, the strong advantage of FlowNetSD is that all the results were obtained with data at the same temporal resolution, *i.e.* at 52 Hz, for all angular velocities. For qualitatively assessing the distribution of errors produced by FlowNetSD, we displayed in Fig. 4 the spatial distribution of EPE and angular velocity errors obtained for an angular velocity of 1 rad/s. From this figure, it can be seen that the error was uniformly distributed over the entire disk, except at the center and on the edges of the disk, where higher values punctually appeared.

Regarding the computation time, FlowNetSD had a training time of about 10 hours on an *NVidia* 980Ti GPU and the inference time on a pair of images was 130 ms. PIV algorithms used the CPU only and had different computation times per pair of images: 1.12 s for PIV (n=2, where n is defined in Sec. V-B) and 5.02 s for PIV-adapt (n=16) with a CPU at 2.3 GHz.

3) Robustness to non-centered images: We investigated the ability of our FlowNetSD-TL network to estimate angular velocity for a spinning disk that was *not* centered with respect to the acquisition field of view. This allowed us to verify whether using only a centered dataset during the training phase introduced a bias. This experiment was carried out for a shift of the disk of +/- 30 pixels in both lateral and axial directions for all 5 angular velocities. The obtained results are reported in Table III. Concerning FlowNetSD-TL, we observed that even if the EPE scores were not as good as those obtained from the centered spinning disks, they remained close with a median error between 0.3 and 1.1 pixels and MAD values $\leq$ 0.3 pixels, showing similar performance as on the *in-vitro* dataset. In terms of angular velocity estimates, results remained accurate between 1 and 4 rad/s (error $\leq$ 0.2 rad/s), and with a maximal error of 0.4 rad/s for the 5 rad/s velocity. For all estimated velocities, MAD values remained lower than 0.3 rad/s, revealing a good consistency of these measurements.

Regarding PIV, as in the centered images case, results for the velocity estimation were consistent with the actual true value up to 2 rad/s, along with EPE errors $\leq$ 0.5 px and MAD values $\leq$ 0.2 px. However, from 3 rad/s, PIV estimates deteriorated with EPE errors over 6.6 px and a systematic underestimation of the velocity that worsens with speed, which confirms the limitations of this algorithms for angular velocities higher than 2 rad/s at a frame rate of 52Hz. As long as PIV-adapt is concerned, results stay close to those obtained on the centered images, both in terms of EPE and estimated velocity. The median error over all speeds remained constant at 0.3 px with MAD values between 0.1 and 0.2 pixels, while the error on the estimated velocity remained lower than 0.1 rad/s. Finally, it is interesting to note that in the case of non-centered images, PIV-adapt produced even better results compared to those obtained with FlowNetSD-TL, revealing the need for adding non-centered scenario cases during data augmentation to make FlowNetSD-TL more robust to such situations.



(a) EPE error map (Ours) (b) Angular velocity error map

Fig. 4. Mean error maps of FlowNetSD with transfer learning computed (a) from the EPE (expressed in px) and (b) the angular velocity metrics (expressed in rad/s) over the synthetic dataset at 1 rad/s.

B. In-vitro experiments

The performance of FlowNetSD-TL was then assessed on the *in-vitro* dataset described in Sec. III-A. Results were reported in Table II. First of all, one can observe an increase of the EPE errors with respect to the values obtained on the simulated dataset for all angular velocities. This illustrates the challenge of processing real data compared to simulated ones. However, EPE errors remained around or under 1.1 px with MAD values $\leq$ 0.4 px for all angular velocities. This shows a good generalization of our network to real data, despite the fact of being trained solely on simulated data. FlowNetSD-TL also returned angular velocities close to the true values, with a

TABLE III
MEDIAN EPE, ESTIMATED ANGULAR VELOCITY ACCURACY AND MAD DISPERSION VALUES COMPUTED INSIDE THE NON-CENTERED SPINNING DISK ON THE SYNTHETIC DATASET FOR FIVE DIFFERENT ANGULAR VELOCITIES. FOR THIS EXPERIMENT, FLOWNETSD-TL WAS COMPARED WITH THE TWO VERSIONS OF THE NON-DEEP LEARNING STATE-OF-THE-ART PIV TECHNIQUE DESCRIBED IN SEC. V-B

Methods	$1rad/s$		$2rad/s$		$3rad/s$		$4rad/s$		$5rad/s$	
	EPE pixel	Velocity rad/s	EPE pixel	Velocity rad/s	EPE pixel	Velocity rad/s	EPE pixel	Velocity rad/s	EPE pixel	Velocity rad/s
FlowNetSD-TL	0.3 ±0.1	1.0 ± 0.1	0.6 ± 0.2	2.2 ± 0.1	0.8 ± 0.2	3.2 ± 0.1	0.8 ±0.2	4.0 ±0.2	1.1 ±0.3	4.6 ±0.2
PIV	0.3 ± 0.1	1.0 ± 0.1	0.5 ± 0.2	1.9 ± 0.3	1.9 ± 1.2	1.7 ± 0.9	4.7 ± 2.0	0.7 ± 0.9	6.6 ± 2.2	0.4 ± 0.9
PIV-adapt	0.3 ± 0.1	1.0 ± 0.1	0.3 ± 0.2	2.0 ± 0.2	0.3 ± 0.1	3.0 ± 0.3	0.3 ± 0.2	3.9 ± 0.4	0.3 ± 0.1	4.9 ± 0.4

maximum error of 0.4 rad/s for the 3 rad/s velocity. Moreover, MAD values were $\leq$ 0.4 rad/s for all angular velocities.

PIV also produced results that were worse on *in-vitro* data, with an increase of the EPE error of 0.2 and 0.6 px for the 1 and 2 rad/s velocities, respectively. In line with the simulated case, PIV results degenerated for angular velocities higher than 3 rad/s, with an EPE error of 6.5 px and an angular velocity error of 4.5 rad/s at 5 rad/s. It can thus be observed that FlowNetSD produced more accurate and stable results (in terms of EPE, MAD values and velocity errors) than the PIV method on *in-vitro* data across the different angular velocities at a frame rate of 52 Hz. Regarding PIV-adapt, results for the velocity estimation were consistent with the true values up to 4 rad/s, along with EPE errors $\leq$ 0.3 px and MAD values $\leq$ 0.1 px. At 5 rad/s, PIV-adapt slightly overestimated the velocity with a value of 5.6 rad/s. Interestingly, even if FlowNetSD works at a lower frame rate of 52 Hz (vs. 312 Hz for PIV-adapt), it produced very close velocity estimates to those of PIV-adapt (mean difference of 0.1 rad/s) for angular velocities $\leq$ 4 rad/s and a better velocity estimate at 5 rad/s. This demonstrates the strong potential FlowNetSD in robustly assessing motion from ultrasound images, even at a lower temporal resolution that is closer to typical values in echocardiography.

VII. DISCUSSION

In this paper, we introduced an evaluation framework for quantitative comparison of different deep learning architectures to quantify motion from ultrasound images. To the best of our knowledge, this is the first time a study evaluates on ultrasound images: *i)* how different CNN architectures compare in terms of motion accuracy; *ii)* what is the impact of transfer learning when specializing networks trained on generic video sequences to ultrasound images; *iii)* how networks trained on simulated data perform on real ultrasound images (*i.e.* generalization ability to real data); *iv)* how a DL-based tracking solution compares with standard state-of-the-art tracking methods tuned for processing ultrasound images.

In terms of comparing different network architectures on our ultrasound database, it appeared that the FlowNetC network performed poorly both before and after transfer learning on the simulated images. This could be explained by the inability of the correlation layer to cope with speckle decorrelation induced by the rotations in our images. When comparing networks with similar architectures (FlowNetS and FlowNetSD), we observed different performance in terms of accuracy. Interestingly, the main difference between the two networks mainly lies in the size of the convolution kernels. This suggests that this parameter impacts a lot the network accuracy and that it needs to be tuned carefully with respect to the acquisition system. In particular, if we assume a speckle size around 2 to 4 times the wavelength of the system, this amounts to 1.2 to 2.2 mm which corresponds to 3 to 5 pixels in our experiments. Regarding the best performing method (FlowNetSD), the size of the underlying convolution kernels was equal to 3 pixels, leading to a receptive field for the first two convolution layers of 3 and 5 pixels, which corresponds to the speckle size involved in the images. This tends to suggest that the extraction of features at the scale of the speckle size represents a good choice for the first layers of a CNN for capturing motion between ultrasound images. Finally, adding convolutional layers to reach the full image resolution at the output of the network did not improve the accuracy on our simulated dataset.

We compared FlowNetSD with the non-deep learning state-of-the-art PIV method [23], which showed excellent performance in a recent challenge [23]. At a sampling frequency of 52 Hz, PIV obtained accurate results for angular velocities $\leq$ 2 rad/s and failed to estimate velocities for values higher than 3 rad/s. To cope with large displacements, this method needs data at higher sampling frequencies, up to 312 Hz at 5 rad/s. In these conditions, PIV-adapt produced lower EPE errors and MAD values than FlowNetSD, especially for *in-vitro* images. Despite a motion estimate performed at 52 Hz, FlowNetSD yielded results with EPE errors $\leq$ 1 px for the full range of the tested velocities and with an accuracy on angular velocity estimates of less than 0.2 rad/s for simulated data and 0.4 rad/s for *in-vitro* data (with a better estimate of the angular velocity at 5 rad/s compared to PIV-adapt). Achieving a similar tracking accuracy over a large range of displacement amplitudes is remarkable, as standard registration algorithms' performance usually tends to deteriorate with larger motion.

One limitation of this study was the focus on a single motion pattern (rotations). Such focus allowed us to complete a study on a fully controlled motion, known to be one of the most important sources of speckle decorrelation. Results obtained from this pilot study revealed that deep learning solutions can be robust and accurate for the estimation of displacement fields in ultrasound, despite high speckle decorrelation. It is therefore important that future works extend this study to more

complex and realistic motion patterns. As out-of-plane motion is known to occur for many imaging scenarios and to bring an additional source of speckle decorrelation, it should also be addressed in future work. This would allow to address this challenge during the training phase, and to evaluate the impact of that specific motion artifact on the tracking accuracy. In particular, 3D numerical simulations could introduce such decorrelation during the synthetic generation of 2D images, in the prospect of incorporating physical distortions in the data augmentation strategy. This would certainly be beneficial to improving the robustness of the tracking algorithm with respect to ultrasound-specific motion artifacts. Finally, the geometry involved in the simulated training dataset (*i.e.* spinning disk centered on the image) was the same as the one in the *in-vitro* dataset, leading to ideal conditions that could improve the performance of the evaluated network. This aspect has been evaluated in Sec. VI-B. From this experiment, it can be seen that velocity estimation remained accurate but the EPE increased with the angular speed, revealing the importance of designing simulations with as much variability as in real cases.

VIII. Conclusion

In this paper, we benchmarked different CNN architectures on simulated and *in-vitro* data for tracking rotations between 1 and 5 rad/s from pairs of ultrasound images. Different networks all derived from the FlowNet2 architecture were compared with the PIV method, a state-of-the-art block matching algorithm tailored to ultrasound. Our quantitative evaluation on both simulated and *in-vitro* images revealed that the FlowNetSD network, after adapting its weights to ultrasound using transfer learning on simulated data, produced accurate motion estimation on *in-vitro* data for the full range of angular velocities and at a single frame rate of 52 Hz. Interestingly, FlowNetSD obtained angular velocity estimates comparable with the PIV-adapt method when the latter required adapting the acquisition frequency up to 312 Hz. This pilot study therefore reveals that deep learning solutions represent a potentially powerful alternative to standard tracking algorithms that can prove both robust and accurate for retrieving displacement fields from ultrasound images, including for large displacements and rotations, despite speckle decorrelation. The full dataset is made available for download at this link.

References

[1] C. Raynaud, H. Langet, M. S. Amzulescu, E. Saloux, H. Bertrand, P. Allain, and P. Piro, "Handcrafted features vs convnets in 2d echocardiographic images," in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. IEEE, 2017, pp. 1116–1119.

[2] A. Østvik, E. Smistad, S. A. Aase, B. O. Haugen, and L. Lovstakken, "Real-time standard view classification in transthoracic echocardiography using convolutional neural networks," *Ultrasound in Medicine and Biology*, vol. 45, no. 2, pp. 374–384, Feb 2019.

[3] S. Leclerc, E. Smistad, J. Pedrosa, A. Østvik, F. Cervenansky, F. Espinosa, T. Espeland, E. A. R. Berg, P.-M. Jodoin, T. Grenier *et al.*, "Deep learning for segmentation using an open large-scale dataset in 2d echocardiography," *IEEE transactions on medical imaging*, 2019.

[4] R. J. van Sloun, R. R. Wildeboer, C. K. Mannaerts, A. W. Postema, M. Gayet, H. P. Beerlage, G. Salomon, H. Wijkstra, and M. Mischi, "Deep learning for real-time, automatic, and scanner-adapted prostate (zone) segmentation of transrectal ultrasound, for example, magnetic resonance imaging–transrectal ultrasound fusion prostate biopsy," *European urology focus*, 2019.

[5] D. Perdios, M. Vonlanthen, A. Besson, F. Martinez, M. Arditi, and J. Thiran, "Deep convolutional neural network for ultrasound image enhancement," in *2018 IEEE International Ultrasonics Symposium (IUS)*, Oct 2018, pp. 1–4.

[6] R. Prevost, M. Salehi, S. Jagoda, N. Kumar, J. Sprung, A. Ladikos, R. Bauer, O. Zettinig, and W. Wein, "3d freehand ultrasound without external tracking using deep learning," *Medical image analysis*, vol. 48, pp. 187–202, 2018.

[7] B. D. de Vos, F. F. Berendsen, M. A. Viergever, M. Staring, and I. Išgum, "End-to-end unsupervised deformable image registration with a convolutional neural network," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2017, pp. 204–212.

[8] H. Li and Y. Fan, "Non-rigid image registration using self-supervised fully convolutional networks without training data," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, 2018, pp. 1075–1078.

[9] M. Morales, D. Izquierdo-Garcia, I. Aganj, J. Kalpathy-Cramer, B. Rosen, and C. Catana, "Implementation and validation of a three-dimensional cardiac motion estimation network," *Radiology: Artificial Intelligence*, vol. 1, p. e180080, 07 2019.

[10] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

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

[12] M. G. Kibria and H. Rivaz, "Gluenet: Ultrasound elastography using convolutional neural network," in *Simulation, Image Processing, and Ultrasound Systems for Assisted Diagnosis and Navigation*. Springer, 2018, pp. 21–28.

[13] B. Peng, Y. Xian, and J. Jiang, "A convolution neural network-based speckle tracking method for ultrasound elastography," in *2018 IEEE International Ultrasonics Symposium (IUS)*. IEEE, 2018, pp. 206–212.

[14] M. Alessandrini, B. Chakraborty, B. Heyde, O. Bernard, M. De Craene, M. Sermesant, and J. D'hooge, "Realistic vendor-specific synthetic ultrasound data for quality assurance of 2-d speckle tracking echocardiography: Simulation pipeline and open access database," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 65, no. 3, pp. 411–422, 2018.

[15] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.

[16] S. Cai, S. Zhou, C. Xu, and Q. Gao, "Dense motion estimation of particle images via a convolutional neural network," *Experiments in Fluids*, vol. 60, no. 4, p. 73, 2019.

[17] J. Porée, D. Posada, A. Hodzic, F. Tournoux, G. Cloutier, and D. Garcia, "High-frame-rate echocardiography using coherent compounding with doppler-based motion-compensation," *IEEE transactions on medical imaging*, vol. 35, no. 7, pp. 1647–1657, 2016.

[18] S. Shahriari and D. Garcia, "Meshfree simulations of ultrasound vector flow imaging using smoothed particle hydrodynamics," *Physics in Medicine & Biology*, vol. 63, no. 20, p. 205011, oct 2018.

[19] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *null*. IEEE, 2003, p. 958.

[22] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[23] V. Perrot and D. Garcia, "Back to basics in ultrasound velocimetry: tracking speckles by using a standard piv algorithm," in *2018 IEEE International Ultrasonics Symposium (IUS)*. IEEE, 2018, pp. 206–212.

[24] P. Joos, J. Porée, H. Liebgott, D. Vray, M. Baudet, J. Faurie, F. Tournoux, G. Cloutier, B. Nicolas, and D. Garcia, "High-frame-rate speckle-tracking echocardiography," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 65, no. 5, pp. 720–728, May 2018.