

Convolution neural networks for real-time needle detection and localization in 2D ultrasound

Cosmas Mwikirize¹  · John L. Noshier² · İlker Hacıhaliloglu^{1,2}

Received: 20 January 2018 / Accepted: 28 February 2018 / Published online: 6 March 2018
© CARS 2018

Abstract

Purpose We propose a framework for automatic and accurate detection of steeply inserted needles in 2D ultrasound data using convolution neural networks. We demonstrate its application in needle trajectory estimation and tip localization.

Methods Our approach consists of a unified network, comprising a fully convolutional network (FCN) and a fast region-based convolutional neural network (R-CNN). The FCN proposes candidate regions, which are then fed to a fast R-CNN for finer needle detection. We leverage a transfer learning paradigm, where the network weights are initialized by training with non-medical images, and fine-tuned with ex vivo ultrasound scans collected during insertion of a 17G epidural needle into freshly excised porcine and bovine tissue at depth settings up to 9 cm and 40°–75° insertion angles. Needle detection results are used to accurately estimate needle trajectory from intensity invariant needle features and perform needle tip localization from an intensity search along the needle trajectory.

Results Our needle detection model was trained and validated on 2500 ex vivo ultrasound scans. The detection system has a frame rate of 25 fps on a GPU and achieves 99.6% precision, 99.78% recall rate and an F_1 score of 0.99. Validation for needle localization was performed on 400 scans collected using a different imaging platform, over a bovine/porcine lumbosacral spine phantom. Shaft localization error of $0.82^\circ \pm 0.4^\circ$, tip localization error of 0.23 ± 0.05 mm, and a total processing time of 0.58 s were achieved.

Conclusion The proposed method is fully automatic and provides robust needle localization results in challenging scanning conditions. The accurate and robust results coupled with real-time detection and sub-second total processing make the proposed method promising in applications for needle detection and localization during challenging minimally invasive ultrasound-guided procedures.

Keywords Needle localization · Ultrasound · Convolution neural networks · Minimally invasive procedures

Introduction

Minimally invasive procedures such as regional anesthesia and interventional oncology involve insertion of a needle toward target anatomy. In practice, image guidance is used to improve targeting accuracy. Of all imaging modalities, ultrasound (US) is ubiquitously used due to its real-time, low-cost and radiation-free capabilities. However, with the conventional two-dimensional (2D) US, aligning the needle with

the US imaging plane at steep angles and deep insertions is difficult. Moreover, reflective signal losses between the needle and probe, which increase with depth, reduce needle visibility. Additionally, high-intensity features arising from hyperechoic soft tissue or bone along the needle trajectory may interfere with needle visibility. Therefore, accurate needle localization remains an ongoing challenge at steep angles and deep insertions.

Hardware-based approaches for improving needle shaft and tip localization, for example external trackers [1] and specialized needles/probes, exist [2]. However, image processing-based methods which do not require additional hardware are easier to adapt in standard clinical workflow. To this end, methods utilizing the Hough transform (HT) [3], parallel projection [4], and machine learning [5–7] have been reported. In [6], a motion-based learning framework is used to

✉ Cosmas Mwikirize
cosmas.mwikirize@rutgers.edu

¹ Department of Biomedical Engineering, Rutgers University, Piscataway, NJ 08854, USA

² Department of Radiology, Rutgers Robert Wood Johnson Medical School, New Brunswick, NJ 08901, USA

track an imperceptible needle. Localization errors of $1.28^\circ \pm 1.09^\circ$ and 0.82 ± 1.21 mm are achieved for the shaft and tip respectively, as well as computation time per frame of 1.18 s. Except for [6], previous methods are largely intensity dependent and are not robust at steep insertion angles. A robust intensity invariant method based on local-phase projections and image regularization has been proposed [8,9]. However, it requires (1) a priori knowledge of the needle insertion side and (2) selection of a fixed region of interest (ROI), close to the transducer surface, for trajectory estimation.

Deep learning (DL) based on convolution neural networks (CNNs) has rapidly emerged as a powerful tool in medical image analysis [10]. The advantage of deep CNNs lies in their ability to automatically learn high-level, layered, hierarchical abstractions from image data by end-to-end training. Recently, deep CNNs were applied for detection of needles inserted in shallow depth (5–30 mm) and shallow insertion angles (10° – 35°) from 3D US data [11]. Shaft localization error of 0.5 mm and a total computation time of 2.19 s (per volume) were achieved, but tip localization was not reported. In this paper, we propose a faster CNN-based framework for needle detection in curvilinear 2D US, suitable for mid-to-steep insertions (40° – 75°) and deep targets (up to 9 cm). We show that our method is robust to low-intensity needle features and the presence of high-intensity artifacts such as bone. The main contributions of this work are: (1) a novel approach for detecting needles inserted in-plane, from 2D US data, using deep convolution neural networks, and (2) a fully automated method for needle trajectory and tip localization, using detection data and intensity invariant feature maps representing the needle. This addresses the limitations of the methods in [8,9] and leads to better tip and trajectory localization.

Methods

To achieve fast and accurate needle detection, we propose the following main steps: (1) generation of region proposals for the needle using a fully convolutional network (FCN). Several regions of various scales and aspect ratios are proposed, and each is assigned a score related to the overlap with the ground-truth needle information in the US image. A multi-loss task is then used to determine final classification and location of the region proposals. (2) The region proposals are utilized for needle detection in a fast R-CNN. Next, we describe the network architecture, and how it is trained to achieve a unified framework for fast needle detection and localization.

Needle detection framework

The proposed network architecture, depicted in Fig. 1, is based on the faster region-based convolutional neural

network (R-CNN) framework [12]. A faster R-CNN is translational invariant. Therefore, needles of various sizes can be inserted at different depths and insertion angles, and the detector will perform accurately, irrespective of the needle's geometrical transformation. Generally, a faster R-CNN consists of two major components: a fast R-CNN [13] and a region proposal network (RPN), modeled as a fully convolutional network (FCN). The two networks share convolution layers. This makes region proposal computation almost cost-free. Similar to [12], we use convolution, pooling, rectified linear unit (ReLU), fully connected, softmax and classification layers as building blocks. Hereafter, we describe design decisions that are specific to our architecture:

Fast R-CNN model Our objective is to have a small network that provides a cost-effective solution with high needle detection accuracy. Therefore, we construct a custom fast R-CNN following the paradigm of [14], consisting of only 5 learned layers: 3 convolution and 2 fully connected layers. The architecture of the fast R-CNN is shown in Table 1. To compare, the detection network in [12] utilizes 16 learned layers (VGG-16) [15] and 8 learned layers (ZF-NET) [16]. The more the learned layers, the more the learned parameters and hence increase in computational complexity. Unlike classification tasks such as in [12] where the input layer size corresponds to the input image size ($224 \times 224 \times 3$), detection tasks focus on a small region of the image where the object of interest might be located. The size chosen should be similar to the smallest detectable object. Therefore, we use a $32 \times 32 \times 3$ image input layer. In Fig. 2, we show the strongest feature maps derived from the convolution layers (*conv_1*, *conv_2* and *conv_3*) for the same input US image. Examining the feature maps and comparing them with the input image, it can be seen that *conv_1* learns the distinct linear features specific to the needle. On the other hand, *conv_2* and *conv_3* learn semantic features associated with the needle.

Regional proposal network (RPN) The RPN generates “areas of interest” for the fast R-CNN. This is achieved by ranking potential bounding boxes for the needle (called anchors). In our design, the input to the RPN is the feature map of the last convolution layer in the fast R-CNN (*conv_3*). We apply a sliding 3×3 convolution window over this feature map. At each window location, a maximum of 9 anchor boxes are predicted, generated from 3 scales with a scaling stride of $1.5 \times (d_m, 1.5 \times d_m, 2.25 \times d_m)$ and 3 aspect ratios (1:1, 1:2, 2:1), where d_m corresponds to the minimum dimension of the bounding boxes in the labeled training images. For a feature map of size $w \times h$, the maximum number of anchors is equal to $w \times h \times 9$. For our case, this is ~ 600 .

To minimize the number of “active” anchor boxes, we eliminate cross-boundary anchors. This leaves ~ 60 anchors per image. Each anchor box is then assigned a positive class label (needle) if the intersection-over-union (IoU) overlap

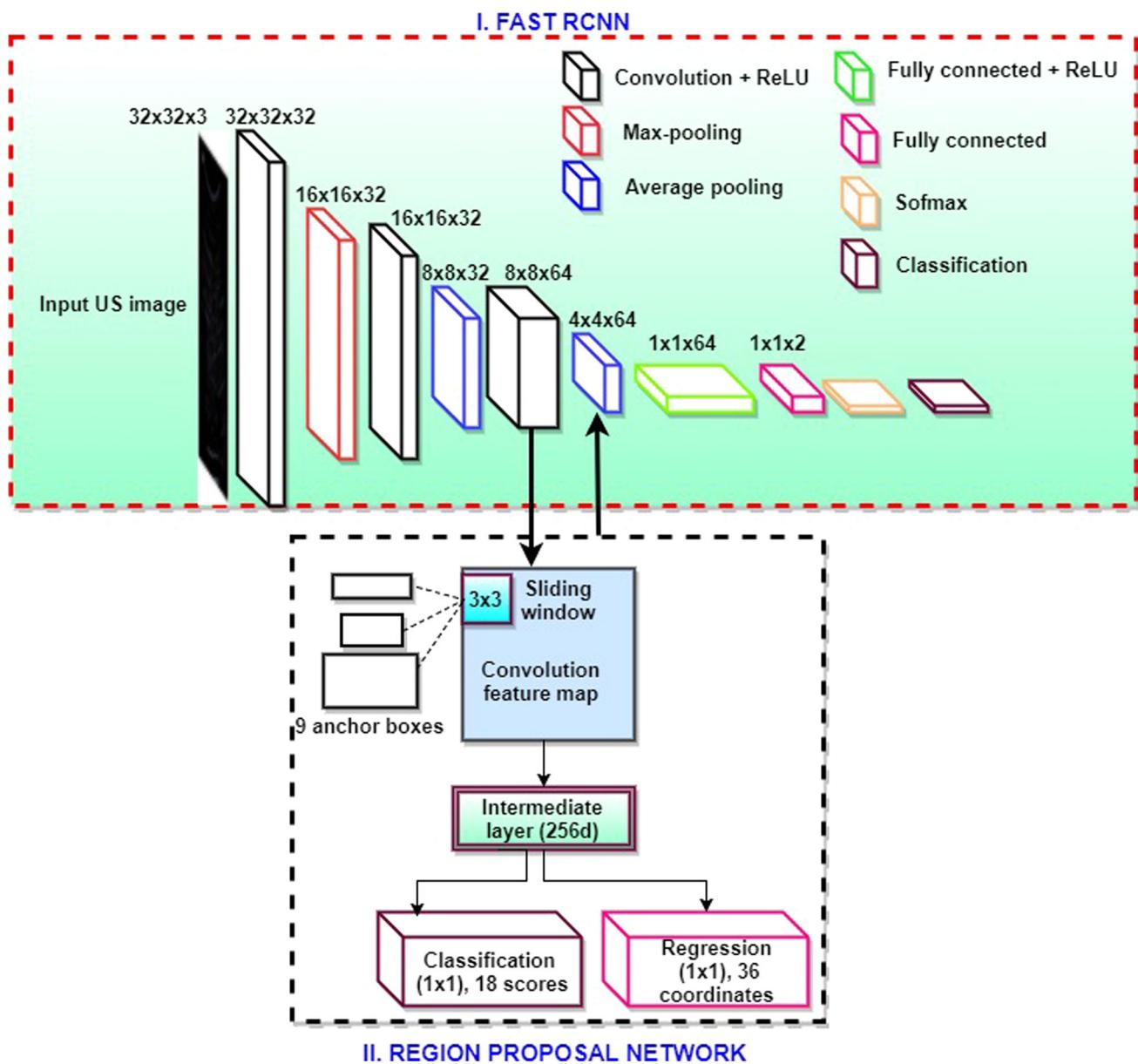


Fig. 1 Block diagram of the proposed framework, consisting of the fast R-CNN (I) and the region proposal network (II). The two networks share convolution layers, making the region proposal process almost cost-free

with the needle bounding box in the labeled US image is > 0.7 . Conversely, a negative class label (no needle) is assigned if $\text{IoU} < 0.3$. In so doing, we detect whether the region centered at each sliding window location contains needle data, and the sliding window location encodes coarse localization information with reference to the input US image. Each window location maps to a 256-dimensional intermediate layer, and lastly, to two sibling 1×1 fully connected layers, shared across all sliding window locations, for box-classification (needle or no needle) and box-regression (for finer localization information). The classification layer outputs a maximum of 18 scores, and the regression layer has

a maximum of 36 outputs encoding the coordinates of the 9 anchor boxes, for each sliding window location. The outputs of these fully connected layers are determined by minimizing the following RPN multi-task loss function [13]:

$$L(p_j, p_j^*, t_j, t_j^*) = \frac{1}{N_c} \sum_j L_c(p_j, p_j^*) + \frac{\lambda}{N_r} \sum_j p_j^* L_r(t_j, t_j^*). \quad (1)$$

Here, the first term describes the box-classifier and the second is the box regressor. j is the anchor index, p_j denotes

Table 1 Architecture of the fast R-CNN, the object detection part of the faster R-CNN

N	Layer name	Dimensions ($W \times H \times D$)	Kernel	Stride	Padding
0	<i>input</i>	$32 \times 32 \times 3$	—	—	—
1	<i>conv_1</i>	$32 \times 32 \times 32$	3	1	1
2	<i>relu_1</i>	$32 \times 32 \times 32$	—	—	—
3	<i>maxpool</i>	$16 \times 16 \times 32$	2	2	—
4	<i>conv_2</i>	$16 \times 16 \times 32$	3	1	1
5	<i>relu_2</i>	$16 \times 16 \times 32$	—	—	—
6	<i>avgpool_1</i>	$8 \times 8 \times 32$	2	2	—
7	<i>conv_3</i>	$8 \times 8 \times 64$	3	1	1
8	<i>relu_3</i>	$8 \times 8 \times 64$	—	—	—
9	<i>avgpool_2</i>	$4 \times 4 \times 64$	2	2	—
10	<i>fc_1</i>	$1 \times 1 \times 64$	—	—	—
11	<i>relu_4</i>	$1 \times 1 \times 64$	—	—	—
12	<i>fc_detection</i>	$1 \times 1 \times 2$	—	—	—
13	<i>softmax</i>	—	—	—	—
14	<i>classification</i>	—	—	—	—

The layers are illustrated in Fig. 1

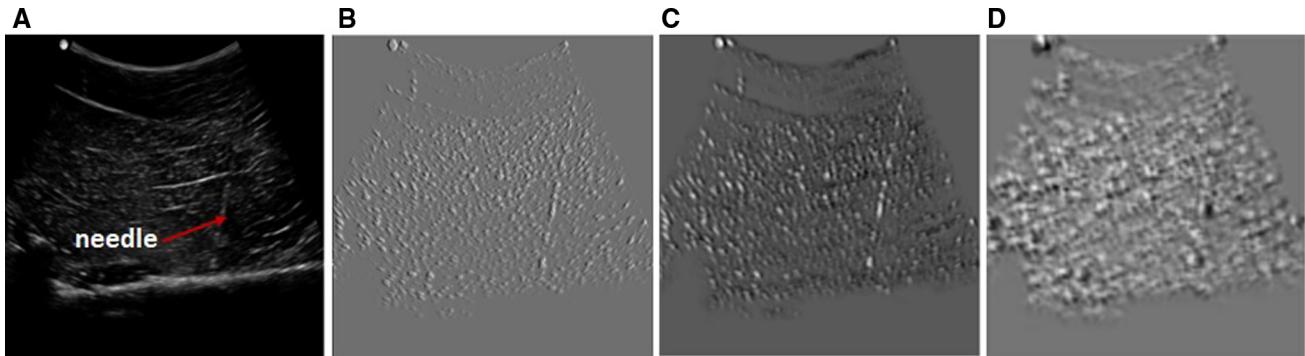


Fig. 2 Visualization of the strongest feature maps from *conv_1* (b), *conv_2* (c) and *conv_3* (d) for the same input image (a). *conv_1* learns the discriminant linear features of the needle, and finer needle features are refined by the subsequent layers

the associated predicted probability, and t_j is the predicted location. $p_j^* \in [0, 1]$ is the ground-truth label arising from the IoU scores described earlier, and t_j^* is the associated location. L_c is the log loss over two classes (needle or no needle), and L_r is the regression loss. L_r is a smooth L_1 loss [14]:

$$L_r(t, t^*) = \begin{cases} 0.5(t - t^*)^2, & \text{if } |t - t^*| < 1 \\ |t - t^*| - 0.5, & \text{otherwise} \end{cases} \quad (2)$$

In (1), N_c is set to 256, the mini-batch size used in our experiments, while N_r is set to 600, the approximate number of total anchor locations in our derived convolution feature map of the RPN. The regularization parameter, λ , is set to 10 as in [12]. Since the derived RPN proposals may overlap, we use non-maximum suppression based on p_j , using a threshold of 0.8. The top-N ranked proposal regions are then fed into the fast R-CNN to generate overall classification and tightened bounding boxes.

Training

To achieve a unified network, the RPN and fast R-CNN are trained using a 4-step alternating process described in [12]: Step (1) The RPN is trained end-to-end. Step (2) The fast R-CNN network is trained using region proposals derived from Step 1. Step (3) The RPN is retrained using shared weights of the fast R-CNN from Step 2. Step (4) Keeping the shared layers fixed, the fast R-CNN is retrained using the updated region proposals from Step 3. We utilize stochastic gradient descent with momentum (SGDM), an initial learning rate of 0.00001, a momentum of 0.9, a mini-batch size of 256, and 10 epochs for each stage. Fast R-CNN layer weights are initialized by pretraining with the CIFAR-10 dataset [17]. Our implementation uses MATLAB's Neural Networking Toolbox.

Dataset overview

Deep learning usually necessitates datasets comprising tens of thousands of images. However, when dealing with medical images, data sizes are usually small. A popular technique to circumvent this problem is transfer learning, where a pretrained network (typically on non-medical images) is fine-tuned using medical data, in our case, the domain specific US images containing needle information. The rich set of low-level features learned from non-medical images has previously been shown to be effective for medical image classification [18]. This is the reason why we use the CIFAR-10 dataset for weight initialization.

We collected 2D B-mode US images using a SonixGPS system (Analogic Corporation, Peabody, MA, USA) with a 2DC5-2/60 curvilinear probe. A 17-gauge (1.5 mm diameter, 90 mm length) Tuohy epidural needle (Arrow International, Reading, PA, USA) was inserted into different samples of freshly excised bovine and porcine tissue in plane, at various insertion angles (40° – 75°) and insertion depths (up to 9 cm). We also overlaid bovine/porcine tissue on a lumbosacral spine phantom, and collected images using a 2D hand-held wireless US system (Clarius C3, Clarius Mobile Health Corporation, Burnaby, British Columbia, Canada). These images were used only for validating needle trajectory and tip localization and were not included during training of the proposed network (“Needle trajectory localization” and “Tip localization” sections).

Using 2500 images from SonixGPS system, we performed tenfold cross-validation: the images were randomly partitioned into 10 subsamples, each of size 250. In turn, 9 of the subsamples (2250 images) were used as training data, while the other 250 images were used as a validation set. The cross-validation process was repeated, with each of the 10 subsamples used exactly once as validation data. The training images were labeled by an expert sonographer to indicate needle locations.

Image preprocessing US images from minimally invasive procedures such as biopsies and epidural spinal injections may contain high-intensity artifacts which increase the likelihood of false positives, thus reducing accuracy of needle detection. Preprocessing aims to reduce the influence of such artifacts. First, the B-mode image $I(x, y)$ is subjected to a Top-hat filter using a linear structuring element $L(x, y)$, and the filtered image $F(x, y) = I(x, y) - [I(x, y) \circ L(x, y)]$. Here, $I(x, y) \circ L(x, y)$ denotes an erosion operation computed by $((I(x, y) \ominus L(x, y)) \oplus L(x, y) \theta)$; \ominus and \oplus are morphological erosion and dilation operations, respectively. The contrast of the filtered image is then stretched to yield $P(x, y) = \max(F(x, y) \times F(x, y)) / \max(F(x, y))$, where $\max(F(x, y))$ is the maximum intensity in the filtered image.

Detection results

The network is evaluated on 2500 US images. Figure 3 shows sample qualitative results. In all cases, the needle shaft is detected, despite low shaft intensity. Needle detection is independent of insertion depth, but rather on availability of needle data in the US image to support the learning process. We have focused on demonstrating our approach on cases where the needle is slightly off-plane or there is substantial reflective loss of the backscattered US signal between the needle and the transducer, as is typical for steep insertions. In practice, these are challenging cases because the shaft will be invisible or discontinuous, and the tip will be separate from the shaft, and may also be imperceptible. For this reason, it is difficult for our approach to detect the whole needle despite the use of multiple scales and aspect ratios during the learning process. Indeed, due to needle discontinuity, various regions of the needle may be detected separately. This scenario is illustrated in Fig. 4. In such cases, we derive the region with the highest confidence score. As we will show in the next section, this is not an impediment to the needle localization process: this detection information is leveraged to automatically estimate the needle trajectory, and henceforth accurately localize the tip.

Training of the faster R-CNN network took an average of 42 min on a single NVIDIA GTX 1060 6GB GPU. The mean needle detection time was 0.04 s. This is a real-time rate and corresponds to 25 frames per s (fps). Overall, average precision and recall rates of 99.6 and 99.78%, respectively, and an F_1 score of 0.99 were achieved. The role of preprocessing in improving detection accuracy cannot be overstated. When the same dataset is analyzed using the proposed approach, but with the network trained on US images that are not preprocessed, the overall precision score drops to 69.83%, the recall to 73.88% and the F_1 score to 0.72.

We investigate the performance of the proposed faster R-CNN through ablation studies, where we remove or add features to the network and analyze the corresponding effect on its performance. First, we remove the RPN. This leaves a stand-alone fast R-CNN network, in the mold of [13]. The network weights are also initialized using the CIFAR-10 dataset and fine-tuned with US data. Next, we keep the RPN and fast R-CNN intact as described in “Needle detection framework” section, but train the network with only random weight initialization, without the CIFAR-10 dataset. We call this faster R-CNN1. Lastly, to show the effect of convolution layers, we train and test the network as described in “Needle detection framework” and “Training” sections, but with one convolution layer removed (faster R-CNN2), and then by adding two more convolution layers (faster R-CNN3). The last case is close to implementation of faster R-CNN with ZF-NET [16]. We also implement a VGG16 network following

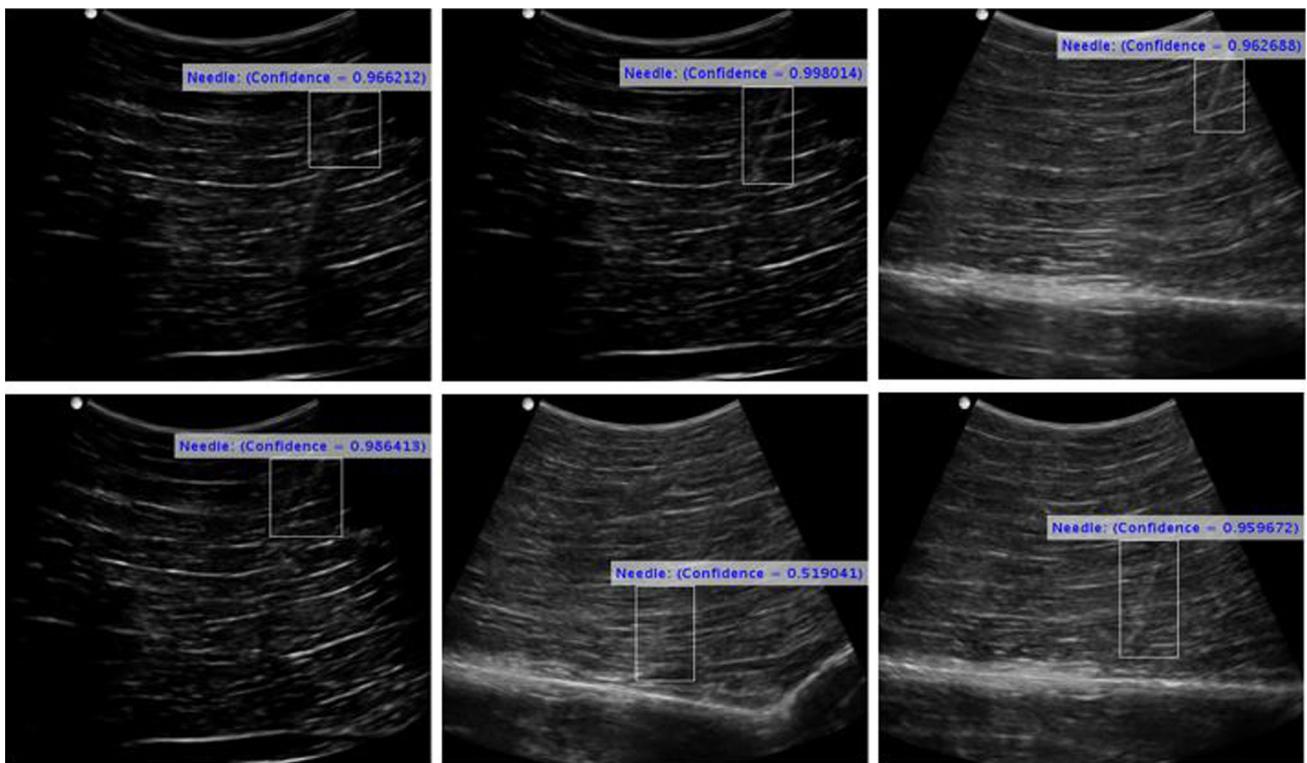


Fig. 3 Needle detection results. The needle shaft is accurately localized, despite low (top row) or imperceptible (bottom row) shaft intensity. The numbers on the bounding boxes are detection scores, a measure of the

confidence of detection. When multiple detections exist, we choose the highest detection score

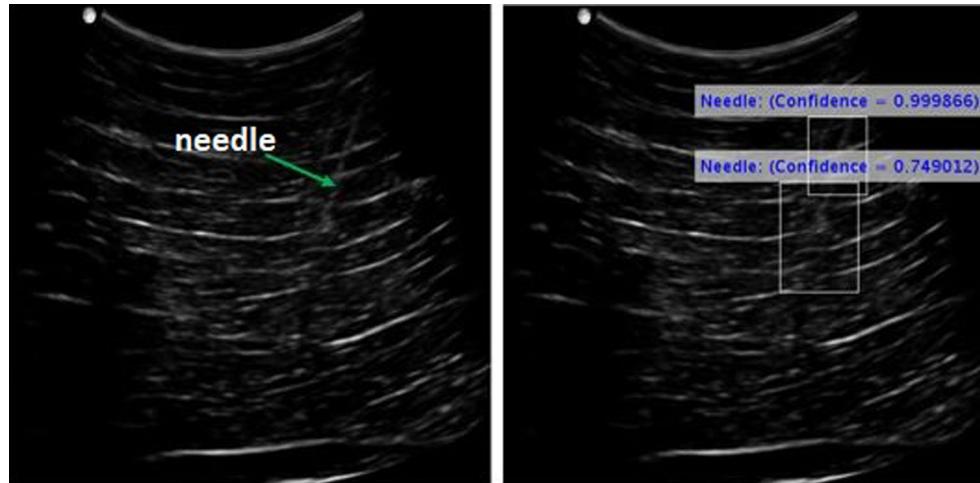


Fig. 4 Multiple needle detection for the same needle. Left: US image. Right: Needle detection results. In such a case, we extract the feature with the highest detection score

the approach of [15], pretrain it with the CIFAR-10 dataset and then fine-tune the weights with US data.

Table 2 shows a comparison of the performance of these networks. We performed tenfold cross-validation only with the proposed method, using a dataset of 2500 images as earlier mentioned, after proving its superior performance to the ablation variants. For the other networks, reported results are

from training on a dataset of 1500 randomly selected images and testing on 400 images that were not part of the training data. The proposed method gives the best combination of detection time and F_1 score. Addition of 2 convolution layers (faster R-CNN3) doubles the computational cost and reduces the F_1 score to 0.88. Reducing the number of convolution layers by one reduces the F_1 score to 0.83. Meanwhile,

Table 2 Comparing the proposed method with ablation variants

	Training time (min)	Test time (s)	Precision (%)	Recall (%)	F_1 score
Proposed method	42	0.04	99.6	99.78	0.99
Fast R-CNN [13]	9	0.84	89.2	97.2	0.93
Faster R-CNN1	29	0.04	79.7	89.3	0.84
Faster R-CNN2	24	0.04	84.7	80.8	0.83
Faster R-CNN3	66	0.08	86.3	90.0	0.88
VGG16 [15]	56	1.03	88.0	84.0	0.86

Faster RCNN1 is trained without pretraining with non-medical images. Faster RCNN2 has one less convolution layer than the proposed method, while faster RCNN3 has 2 more convolution layers

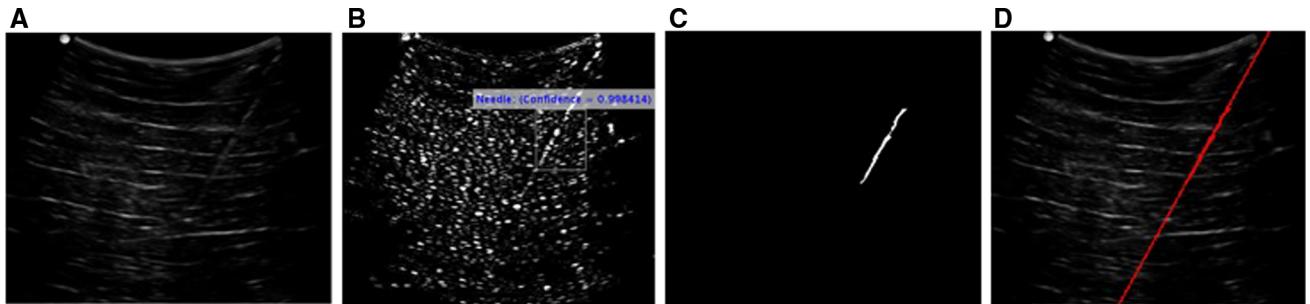


Fig. 5 Needle trajectory estimation. **a** US image, **b** preprocessed image with a marked ROI. The ROI is automatically determined as the needle bounding box from the detection step. **c** Image after extraction of

local-phase features and finding the longest connected component. **d** Estimated trajectory (red line) after applying the Hough transform

the VGG16 network is the slowest of all: the detection task is computationally heavy (~ 1 s on a GPU). This is anticipated since VGG16 has 41 layers (16 learned layers). Next, we present applications of the localization information deduced from the needle detection process.

Applications

Ultimately, the desired output is an enhanced needle image which can facilitate automatic needle shaft and tip localization. In Figs. 3 and 4, we observe that our method may not give a bounding box that contains the whole needle. While this is a limitation, the detection result is sufficient for automatic detection of the needle insertion side, estimation of the needle insertion trajectory, and facilitating automatic localization of the tip. This addresses the limitations of [8,9] where there is a need for a priori knowledge of the needle insertion side and selection of a fixed ROI close to the transducer surface. Next, we describe how we achieve automatic tip localization.

Needle trajectory localization

Recall that the detection step yields a bounding region for the needle, which we use as an automatically generated ROI. First, we construct a phase-based image descriptor, called

phase symmetry (PS (x, y)), using an orientation-tuned 2D Log-Gabor filter bank [9] applied to the ROI on the preprocessed US image (Fig. 5b). This filter's function is defined as:

$$\text{LG}(\omega, \theta) = \exp\left(\frac{-\log\left(\frac{\omega}{\kappa}\right)^2}{2\log(\sigma_\omega)^2}\right) \exp\left(\frac{-(\theta - \theta_m)^2}{2(\sigma_\theta)^2}\right). \quad (3)$$

Here, (ω, θ) are the frequency/orientation coordinates, κ is the center frequency, σ_ω is the bandwidth of the frequency spectrum, σ_θ is the angular bandwidth, and θ_m is the specific filter orientation. Although [9] provided a means to tune most of the filter parameters, extracting the PS image required a priori information: an estimate of the needle trajectory or insertion side of the needle (for θ_m) and a fixed ROI containing the shaft, close to the transducer surface. Now, we are determining the ROI, the insertion side and the insertion angle automatically. The ROI corresponds to the needle bounding box from the detection process. Figure 6a shows how we automatically determine the insertion side and an estimate of the insertion angle from this bounding box. We define the bounding box with parameters x_i , y_i , L and W . If (x_c, y_c) is the center of the image, for right-side insertions, $x_i < x_c$, and for left-side insertions, $x_i > x_c$. For the former case, an estimate of the needle trajectory, β is given by $\beta = \tan^{-1}\left(\frac{W}{L}\right)$ and for the latter, $\beta = 90 + \tan^{-1}\left(\frac{W}{L}\right)$. The filter bank is applied with 3 scales and 3 orientations,

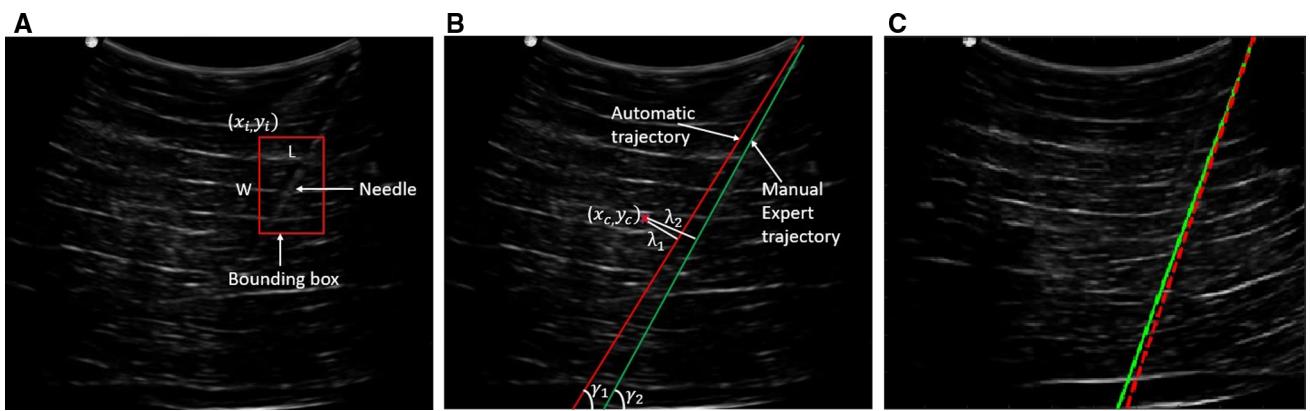


Fig. 6 **a** Estimation of needle insertion side and trajectory. Initial estimate of the trajectory is obtained from the diagonal of the needle bounding box. **b** Parameters used in calculating needle trajectory error. (x_c, y_c) is the center of the image. The automatically determined parameters are subtracted from the parameters obtained manually by an

expert. Here, the difference between the two trajectories is exaggerated for illustration purposes. **c** Trajectory from proposed method (green) and expert-labeled (red, dashed) overlaid on US image. The proposed method gives an accurate prediction of the trajectory

Table 3 Trajectory localization errors on data collected with the SonixGPS imaging system

Method	$\gamma_1 - \gamma_2$ ($^{\circ}$)	$\lambda_1 - \lambda_2$ (mm)
Proposed method	0.8 ± 0.4	0.4 ± 0.1
Method in [8]	1.5 ± 0.6	1.1 ± 0.2
Method in [9]	1.8 ± 0.8	1.4 ± 0.2

Table 4 Trajectory localization errors on data collected with the 2D Clarius C3 hand-held imaging system over a bovine/porcine lumbarosacral spine phantom

Method	$\gamma_1 - \gamma_2$ ($^{\circ}$)	$\lambda_1 - \lambda_2$ (mm)
Proposed method	0.82 ± 0.4	0.38 ± 0.1
Method in [8]	1.74 ± 0.6	1.31 ± 0.2
Method in [9]	1.86 ± 0.8	1.42 ± 0.2

$\theta_m = [\beta - 10, \beta, \beta + 10]$, and yields PS (x, y) , containing a prominent needle feature.

The PS (x, y) image may contain artifacts not belonging to the needle. We eliminate these by extracting the

longest connected component. The output of this operation, PS_L (x, y) is shown in Fig. 5c: a distinct, intensity invariant straight feature. The last step in trajectory estimation involves application of the Hough transform (HT). The automatically estimated trajectory is shown in Fig. 5d. We calculate trajectory error by comparing the automatically determined trajectory with the gold-standard trajectory estimated by an expert sonographer. The parameters used in computing this error are shown in Fig. 6b. If we denote the original US image as $I(x, y)_{m \times n}$, where m and n are the horizontal and vertical dimensions, then the center of the image is estimated as $(x_c, y_c) = (m/2, n/2)$. We calculate: (1) γ_1 , the angle subtended by the automatically detected trajectory on the horizontal axis, (2) γ_2 , the angle subtended by the trajectory labeled by an expert, on the horizontal axis, (3) λ_1 , the shortest distance between the automatically detected trajectory and the center of the image, and (4) λ_2 , the shortest distance between the expert-labeled trajectory and the center of the image. The trajectory error is then quantified using $\gamma_1 - \gamma_2$ and $\lambda_1 - \lambda_2$.

In Fig. 6c, we show a qualitative comparison between trajectory localization from the proposed method and the

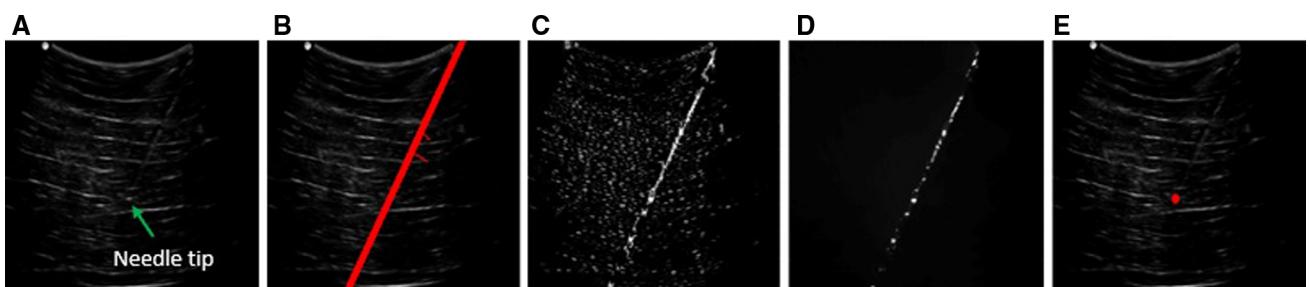


Fig. 7 The needle tip localization process. **a** US image. **b** Extended trajectory region (red) computed with the HT, and used to form the trajectory mask. **c** Output of MLESAC algorithm. **d** Enhanced needle image. **e** Automatically localized tip (red)

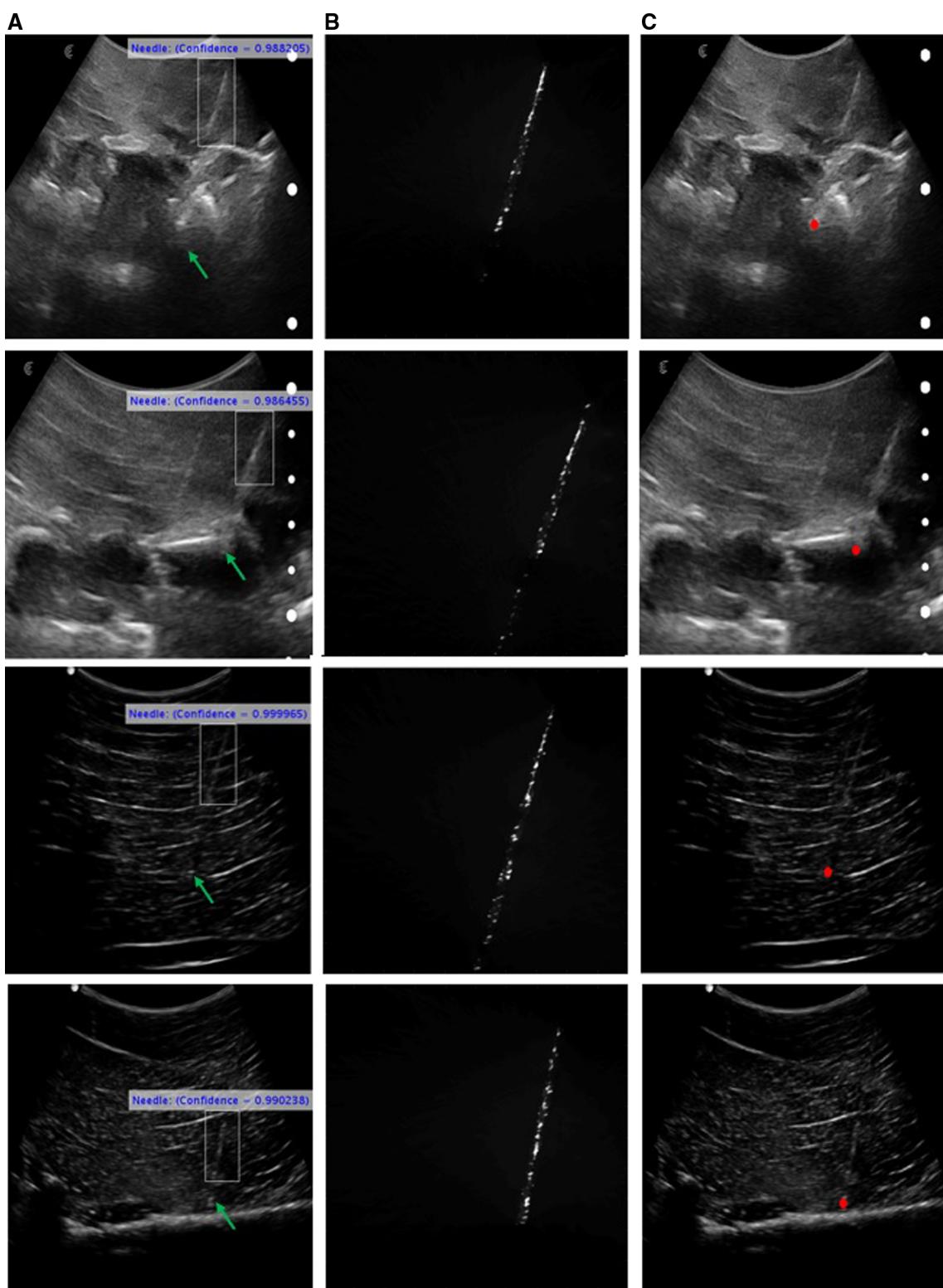


Fig. 8 Qualitative results for tip localization. **a** US image. The green arrow points to the expert-localized tip. **b** Enhanced tip image, **c** localized tip (red). Rows 1 and 2 show results for images collected over a bovine lumbosacral spine phantom (using Clarius US system), while

rows 3 and 4 show results for images collected over soft tissue (using SonixGPS imaging system). The proposed method achieves automatic and accurate tip localization despite low or inconspicuous needle information, and high-intensity artifacts along the needle trajectory

expert-localized trajectory. Quantitative results from analysis of 400 US images collected with the same imaging system as the training dataset, but not part of the training data, are given in Table 3. We also present results from analysis of 400 images collected with the Clarius US system, which were not part of the training data, in Table 4. Also shown are corresponding results from applying the state-of-the-art methods in [8,9] on the same datasets. The proposed approach gives superior performance to these methods, with statistically significant improvement in trajectory error ($p < 0.005$), obtained using one-tailed paired t test. However, there is no significant difference between the results obtained with the two imaging platforms, over different imaging media. This result emphasizes the clinical relevance of our approach.

Tip localization

From the estimated needle trajectory, we create a mask of the trajectory region; a region extending a few pixels on either side of the needle axis, where we are sure the needle axis must lie. Next, we use the approach previously reported in [8,9]. A mask of the trajectory region is convolved with the preprocessed US image. This is followed by line fitting using the MLESAC algorithm [19]. Finally, the needle tip is automatically localized by filtering the resulting image with a 2D Log-Gabor filter, and performing a statistical search along the trajectory. For a detailed description of this approach, please refer to [9]. The processing pipeline is illustrated in Fig. 7. Evaluation of tip localization is performed on 400 US images collected with the Clarius US system. Overall, the tip localization process (excluding needle detection) executes for 0.54 s. In Fig. 8, we show qualitative tip localization results for both images collected over the spine phantom and soft tissue. The former demonstrate clinical feasibility for procedures such as lumbar facet joint and medial branch blocks. Note that despite interference from bone and other artifacts, the needle tip is accurately localized. The tip localization error is determined from the Euclidean distance between the automatically localized tip and the manually localized tip by an expert sonographer and yields 0.23 ± 0.05 mm. On the other hand, the methods in [8,9] yield localization errors of 0.55 ± 0.12 and 0.84 ± 0.25 mm respectively. Comparing the proposed method and these previous methods, there is a statistically significant improvement in tip localization accuracy ($p < 0.005$) obtained using a one-tailed paired t test.

Discussion and conclusion

In this work, we have proposed a novel method for detection of needles in 2D US data, utilizing convolution neural networks. Our method achieves high precision (99.6%) recall rate (99.78%), and a detection time of 0.04 s. The proposed

method does not always achieve detection of the whole needle, especially if the needle shaft is broken, as is typical at steep insertion angles. However, the detected portion of the needle provides an automatically generated ROI, which is utilized for trajectory estimation and tip localization. Evaluation of the method on images collected with a different imaging system and not part of the training dataset reveals better trajectory estimation and tip localization accuracy than previously reported methods [8,9]. The overall detection and localization time of 0.58 s (~ 2 fps) is also better than previously reported [6–9,11] and can further be improved with better computing hardware. The proposed method is promising for clinical application and would work seamlessly in any imaging scenario, if the network is trained on a larger clinical dataset incorporating different imaging conditions. Although we have focused on hand-held needles, our method could find use in minimally invasive robotic interventions.

The proposed method achieves needle detection in cases where the needle is imperceptible. However, our current localization approach, based on the Hough transform, only works for non-bending needles. Further, reliance on an expert sonographer to determine the ground-truth tip localization is a limitation when the tip information is completely invisible in the US image. Future work will focus on localization of bent needles, use of a tracking system for ground-truth tip localization, and evaluating the proposed method on *in vivo* US data.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent This article does not contain patient data.

References

1. Moore J, Clarke C, Bainbridge D, Wedlake C, Wiles A, Pace D, Peters T (2009) Image guidance for spinal facet injections using tracked ultrasound. *Med Image Comput Comput Assist Interv* 12(Pt 1):516–523
2. Xia W, West S, Finlay M, Mari J, Ourselin S, David A, Desjardins A (2017) Looking beyond the imaging plane: 3D needle tracking with a linear array ultrasound probe. *Sci Rep* 7(1):3674
3. Ayvali E, Desai J (2014) Optical flow-based tracking of needles and needle-tip localization using circular hough transform in ultrasound images. *Ann Biomed Eng* 43(8):1828–1840
4. Barva M, Uhercik M, Mari JM, Kybic J, Duhamel JR, Liebgott H, Hlavac V, Cachard C (2008) Parallel integral projection transform for straight electrode localization in 3-D ultrasound images. *IEEE Trans Ultrason Ferroelectr Freq Control* 55(7):1559–1569
5. Mathiassen K, Dall’Alba D, Muradore R, Fiorini P, Elle O (2017) Robust real-time needle tracking in 2-D ultrasound images using

- statistical filtering. *IEEE Trans Control Syst Technol* 25(3):966–978
- 6. Beigi P, Rohling R, Salcudean S, Ng G (2017) CASPER: computer-aided segmentation of imperceptible motion-a learning-based tracking of an invisible needle in ultrasound. *IJCARS* 12(11):1857–1866
 - 7. Hatt CR, Ng G, Parthasarathy V (2015) Enhanced needle localization in ultrasound using beam steering and learning-based segmentation. *Comput Med Imaging Graph* 41:46–54
 - 8. Mwikirize C, Noshier JL, Hacihaliloglu I (2018) Signal attenuation maps for needle enhancement and localization in 2D ultrasound. *Int J CARS* 13(3):363–374
 - 9. Hacihaliloglu I, Beigi P, Ng G, Rohling RN, Salcudean S, Abolmaesumi P (2015) Projection-based phase features for localization of a needle tip in 2D curvilinear ultrasound. In: Medical image computing and computer-assisted intervention. Lecture notes in computer science, vol 9349. Springer, pp 347–54
 - 10. Litjens G, Kooi T, Bejnordi B, Setio A, Ciompi F, Ghafourian M, van der Laak J, van Ginneken B, Snchez C (2017) A survey on deep learning in medical image analysis. *Med Image Anal* 42:60–88
 - 11. Pourtaherian A, Zanjani FG, Zinger S, Mihajlovic N, Ng G, Korsten H, de With P (2017) Improving needle detection in 3D ultrasound using orthogonal-plane convolutional networks. In: Medical image computing and computer-assisted intervention. Lecture notes in computer science, vol 10434. Springer, pp 610–618
 - 12. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 39(6):1137–1149
 - 13. Girshick R (2015) Fast R-CNN. In: Proceedings of IEEE international conference on computer vision, pp 1440–148
 - 14. Krizhevsky A, Sutskever I, Hinton G (2017) Imagenet classification with deep convolutional neural networks. *Commun ACM* 60(6):84–90
 - 15. Simonyan K, Zisserman A (2015) Very deep convolutional network for large-scale image recognition. In: Proceedings of international conference on learning representations
 - 16. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional neural networks. In: Proceedings of 13th European conference on computer vision, vol 8183
 - 17. Krizhevsky A (2009) Learning multiple layers of features from tiny images. CIFAR10 dataset
 - 18. Huynh B, Li H, Giger M (2016) Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. *J Med Imaging* 3(3):034501
 - 19. Torr PHS, Zisserman A (2000) MLESAC: a new robust estimator with application to estimating image geometry. *J Comput Vis Image Underst* 78(1):138–156