



H-ProSeg: Hybrid ultrasound prostate segmentation based on explainability-guided mathematical model



Tao Peng^{a,1}, Yiyun Wu^{b,1}, Jing Qin^c, Qingrong Jackie Wu^d, Jing Cai^{a,*}

^a Department of Health Technology and Informatics, The Hong Kong Polytechnic University, Hong Kong, China

^b Department of Medical Technology, Jiangsu Province Hospital, Nanjing, Jiangsu, China

^c Department of Nursing, The Hong Kong Polytechnic University, Hong Kong, China

^d Department of Radiation Oncology, Duke University Medical Center, Durham, NC, USA

ARTICLE INFO

Article history:

Received 31 July 2021

Revised 16 February 2022

Accepted 11 March 2022

Keywords:

Accurate prostate segmentation

Transrectal ultrasound

Principal curve

Differential evolution-based machine

learning

Explainability-guided mathematical model

ABSTRACT

Background and Objective: Accurate and robust prostate segmentation in transrectal ultrasound (TRUS) images is of great interest for image-guided prostate interventions and prostate cancer diagnosis. However, it remains a challenging task for various reasons, including a missing or ambiguous boundary between the prostate and surrounding tissues, the presence of shadow artifacts, intra-prostate intensity heterogeneity, and anatomical variations.

Methods: Here, we present a hybrid method for prostate segmentation (H-ProSeg) in TRUS images, using a small number of radiologist-defined seed points as the prior points. This method consists of three sub-networks. The first subnetwork uses an improved principal curve-based model to obtain data sequences consisting of seed points and their corresponding projection index. The second subnetwork uses an improved differential evolution-based artificial neural network for training to decrease the model error. The third subnetwork uses the parameters of the artificial neural network to explain the smooth mathematical description of the prostate contour. The performance of the H-ProSeg method was assessed in 55 brachytherapy patients using Dice similarity coefficient (DSC), Jaccard similarity coefficient (Ω), and accuracy (ACC) values.

Results: The H-ProSeg method achieved excellent segmentation accuracy, with DSC, Ω , and ACC values of 95.8%, 94.3%, and 95.4%, respectively. Meanwhile, the DSC, Ω , and ACC values of the proposed method were as high as 93.3%, 91.9%, and 93%, respectively, due to the influence of Gaussian noise (standard deviation of Gaussian function, $\sigma = 50$). Although the σ increased from 10 to 50, the DSC, Ω , and ACC values fluctuated by a maximum of approximately 2.5%, demonstrating the excellent robustness of our method.

Conclusions: Here, we present a hybrid method for accurate and robust prostate ultrasound image segmentation. The H-ProSeg method achieved superior performance compared with current state-of-the-art techniques. The knowledge of precise boundaries of the prostate is crucial for the conservation of risk structures. The proposed models have the potential to improve prostate cancer diagnosis and therapeutic outcomes.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

Prostate cancer is one of the most common noncutaneous cancers and the second leading cause of cancer-related deaths in men [1]. Transrectal ultrasound (TRUS) is an important imaging technique for prostate cancer diagnosis and treatment [2]. In various clinical applications of TRUS, the prostate volume is usually delineated manually by trained clinicians. However, manual de-

lineation is time-consuming, tedious, and subjective, resulting in heavy workloads for radiologists and large variability in prostate volume delineation. Despite many advances in the medical imaging field [3,4], and specifically, in prostate segmentation techniques [5,6], accurate prostate segmentation in ultrasound images remains a challenging task for the following reasons: (1) missing boundaries due to shadow artifacts produced by surrounding structures (i.e., the bladder and seminal vesicles), (2) blurry boundaries caused by low contrast between the prostate and neighboring organs, (3) inhomogeneous intensity distribution of the prostate structure, and (4) variability in prostate shape. Fig. 1 shows several challenging prostate segmentation cases using TRUS images.

* Corresponding author.

E-mail address: jing.cai@polyu.edu.hk (J. Cai).

¹ These authors contributed equally to this work.

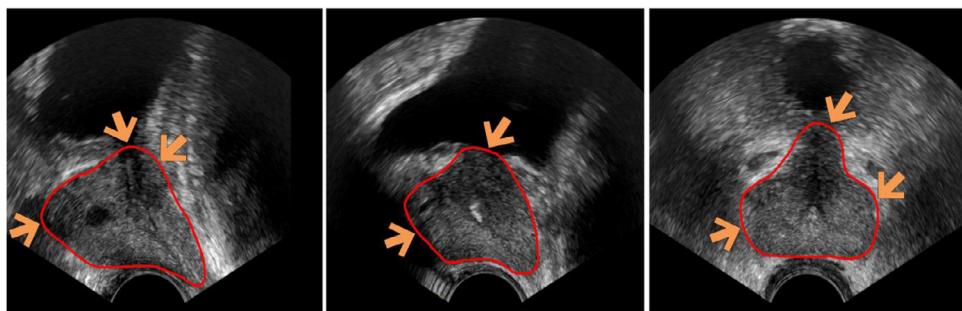


Fig. 1. Example of transrectal ultrasound images of the prostate. The red contour shows the prostate contour. There are various prostate shapes, and the prostate tissues present inhomogeneous intensity distributions. Meanwhile, there are missing or blurry boundaries between the prostate and the neighboring organs due to shadow artifacts. The orange arrows indicate missing or fuzzy boundaries. All TRUS images are from the Nanjing University of Chinese Medicine (shown in Section 3.5).

Various segmentation algorithms have been proposed in recent years, including (1) threshold segmentation algorithms [7], (2) region segmentation algorithms [8], and (3) contour detection algorithms [9]. Lei et al. [10] proposed a multidirectional deeply supervised V-Net for prostate segmentation in TRUS images. Although the Dice similarity coefficient (DSC) of the model is as high as 92%, the accuracy of the segmentation may be affected by systematic errors (e.g., a physician's contouring style), and this technique shows poor robustness. Jia et al. [11] developed a coarse-to-fine framework for prostate segmentation by combining a registration-based coarse segmentation method with an ensemble deep convolutional neural network (DCNN)-based fine segmentation method. This method was evaluated using the PROMISE12 dataset [12] and achieved a mean DSC of only 88%. Wang et al. [13] designed a three-dimensional deep neural network equipped with attention modules for prostate segmentation in TRUS images using complementary information encoded in different layers of the convolutional neural network (CNN). However, the refinement step of this method was not able to couple adjacent nodes and thus could not favor assignments with the same label to spatially proximal voxels. Compared with the methods described by Lei and Jia, Wang's method contours anatomical tissues more time efficiently and achieves more realistic shapes of anatomical tissues [14].

Region representation [15] and curve approximation models [16] have been used as the main components of contour extraction models to describe the contours of anatomical structures. Tan et al. [17] proposed an automatic prostate segmentation model that combines deep network and variational methods to analyze magnetic resonance imaging (MRI) images. Bardis et al. [18] developed a deep learning model for prostate segmentation and evaluated its performance in training sets of different sizes. However, the model robustness was not validated in more complicated situations, such as in the presence of noise, clutter, or partial occlusions. Tian et al. [19] designed a highly accurate interactive method based on a graph convolutional network for prostate segmentation in MRI images. However, the method is severely affected by observer variability due to human interactions in fine-tuning the coarse segmentation results from the deep learning module.

As a well-known principal curve-based method, principal component analysis is mainly used for dimension reduction and feature extraction [20]. The first principal component line minimizes the squared Euclidean distance from data points (P_n) set to lines. Furthermore, the first principal component is self-consistent, i.e., each point p_i of curve f is the average of all points P_n that project to curve f [21]. Based on the self-consistent property of principal components, Hastie and Stuetzle proposed the notation of a principal curve, which was defined as a “self-consistent” smooth curve that passes through the “middle” of an n -dimensional probability distribution or data cloud [22]. Kegl et al. [20] then proposed a

new notation of the principal curve with a limited length, known as the K-segment principal curve (KPC) method. The basic idea of the KPC method is to start with a first component line and, in each iteration of the method, increase the number of segments by one through the addition of a new vertex to the polygonal curve produced during the previous iteration. After adding a new vertex, the positions of all vertices are updated so that the resulting curve minimizes a penalized distance function. Based on the KPC method, we previously proposed a closed K-segment principal curve (CKPC) method [23]. This was the first time that a principal curve-based method was used in the medical image segmentation field. In contrast to the KPC method, the CKPC method uses a closed square rather than a line as the starting curve, while adding shape constraints so that it can better fit the closed data. Furthermore, the CKPC method includes stop conditions and constraint conditions (i.e., the selection of vertices and segments) to improve the accuracy of the results.

As the results of the principal curve-based method are a polygonal curve [24], Wang et al. [25] integrated a backpropagation neural network (BPNN) with the principal component analysis method to make the principal curve smoother. Based on the method reported by Wang et al. [25], we previously combined the CKPC method with the BPNN method for lung segmentation in computed tomography images [23]. Furthermore, when we used randomly initialized parameters (i.e., weights and thresholds) of the neural network (NN), the results of the NN were always easily constrained within the local minimum during training [26]. To address this problem, a differential evolution (DE)-based method was used to choose the optimal parameters of the NN due to its ability to identify the global optimum [27].

Accurate prostate segmentation in ultrasound images is a challenging task due to the low signal/noise ratio, low contrast, and blurry or unseen boundaries. The DSC values of most automatic methods used for TRUS images are only approximately 0.9 [28]. Hence, we propose a hybrid algorithm (H-ProSeg) for prostate segmentation in TRUS images using a small number of radiologist-defined seed points as the prior. The main contributions of our method are as follows.

- A segmentation framework, named H-ProSeg, was developed, consisting of an improved closed principal curve method and an improved machine learning method integrated into a segmentation framework. This framework takes advantage of the ability of the principal curve method to automatically pass through the “middle” of the dataset [29] and use machine learning to decrease the model error.
- Based on our previous CKPC method [14,30,23], a novel redefined closed K-segment principal curve (RCKPC) method was developed by including a vertex cleaning method and a redefined vertex optimization step. Furthermore, the RCKPC method

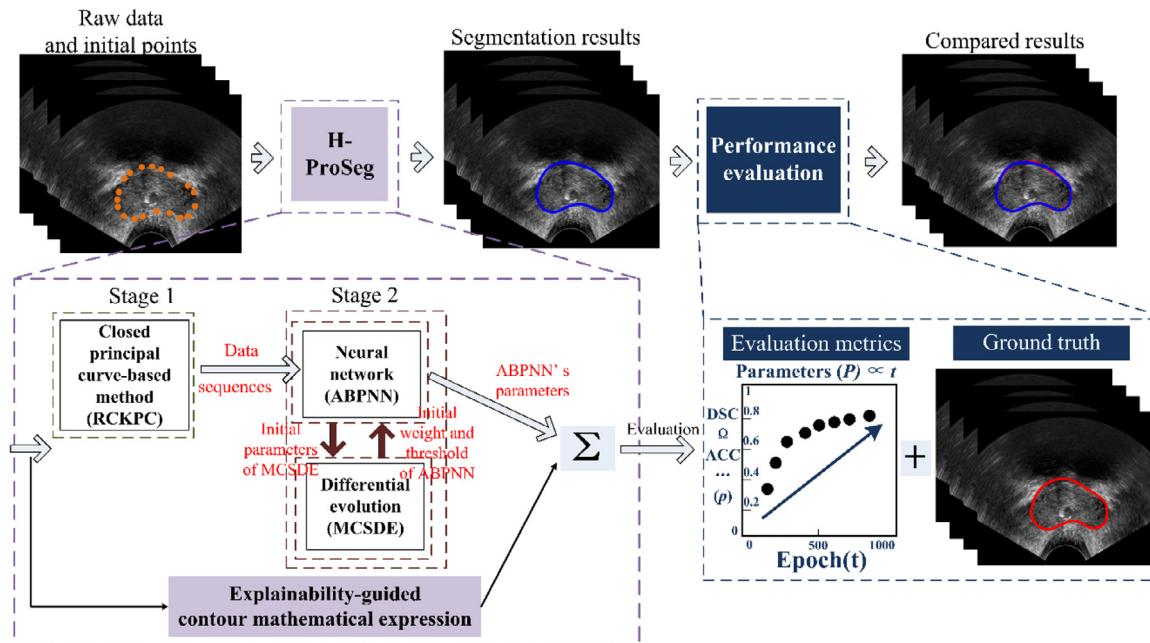


Fig. 2. Flowchart of the proposed H-ProSeg method. RCKPC, redefined closed K-segment principal curve; MCSDE, memory-based, cuckoo search-based differential evolution; ABPNN, adaptive learning-rate backpropagation neural network.

Table 1
Input and output of each step of the proposed method.

Method	Input	Output
RCKPC	Initial seed data points $\{p_1, p_2, \dots, p_n\}$	Data sequences: data points, p_i , and corresponding projection index, $t(p)$
MCSDE	Initial parameters (shown in Section 2.3.2)	Optimal weight and threshold of the ABPNN
ABPNN	Data sequences	Smooth contour consisting of optimized vertices

integrates the ability of the CKPC method to better deal with a closed dataset by using several cycle conditions.

- A new memory-based, cuckoo search-based differential evolution (MCSDE) method was developed to obtain the initial optimal parameters (i.e., weights and thresholds) of the adaptive learning-rate backpropagation neural network (ABPNN) [31]. The MCSDE method is based on previously described methods [32,33] but is innovatively enhanced with multi-mutation operators to ensure population diversity and, subsequently, improve the capability of the global search.
- An explainability-guided mathematical map function (shown in Section 2.4) denoted by the parameters of the ABPNN was first developed to generate smooth lung contours, represented by the outputs of the ABPNN (i.e., optimized vertices), that match the ground truth (GT) contours.

2. Materials and methods

The H-ProSeg method for prostate segmentation in TRUS images is shown in Fig. 2. The method involves a three-stage framework. In the first stage, an improved RCKPC method is used to achieve data sequences using less than 8% of the points of a contour manually delineated by radiologists as the prior points (Section 2.2). The MCSDE method is then used to preliminarily search for the initial optimal connection weights and thresholds of the ABPNN (Section 2.3). In the final stage, the ABPNN is used for training to minimize the model error and achieve accurate segmentation results. The parameters of the ABPNN were used to denote the explainability-guided mathematical model of the prostate contour (Section 2.4) Table 1 shows the input and output of each step of the proposed method.

2.1. Basic instruction

Our method is a hybrid framework containing principal curve-based, differential evolution-based, and machine learning-based methods and a mathematical model. Appendix A shows several important notations, namely, the most frequent notations used for each model in this study. Furthermore, for readers unfamiliar with the principal curve-based method, brief descriptions of the KPS and CKPS methods are given in Appendix B.2, and descriptions of the DE method and the previous mathematical model are given in Appendix B.3 and Appendix B.4, respectively.

2.2. RCKPC method

To solve the issue that the K-segment principal curve (KPC) method [29] cannot correctly obtain the projection index of a closed dataset, we previously developed an improved CKPC method by extending several constraints and stop conditions [14,30,23]. To the best of our knowledge, the improved CKPC method is the first such method developed for the medical image segmentation field. Compared with spline methods of data fitting (i.e., the least-squares method [34] and the cubic spline interpolation method [35]), the principal curve-based method obtains more continuous and accurate results [23]. However, the vertex optimization step of the CKPC method has low efficiency when dealing with complex data, and the practical usefulness of the CKPC method may be affected by abnormal data in the wrong position. Based on previously described methods [36] and [37], we developed a new RCKPC method to address the aforementioned issues by adding two improvements, i.e., (1) a redefined vertex optimization step

Algorithm 1 Improved RCKPC method.

```

Input: the initial data points,  $P_n = \{p_1, p_2, \dots, p_n\}$ 
Output: data sequences
1: Normalize the initial points  $p_i$  to the range  $\{(-1, -1) \sim (1, 1)\}$ 
2: The initial first principal component curve is closed and starts with a small square with the following vertex coordinates:  $\{(0.1, 0.1), (-0.1, 0.1), (-0.1, -0.1), (0.1, -0.1), (0.1, 0.1)\}$ 
3: For  $i = 1$  to the initial points set  $p_i$  do
4: Insert one vertex  $v_1$  and obtain the distance  $L_1$  from vertex  $v_1$  to the curve
5: if ((The curve's shape keeps closed) && ( $90^\circ <$  Angle between segments on the curve  $< 180^\circ$ ))
6: Obtain the current distance  $L_2$  when inserting another vertex  $v_2$ 
7: if ( $L_2 - L_1 < 0.002$ )
8: Projection step (described by Kegl et al. [20])
9: Improved vertex optimization step (shown in Section 2.2.1)
10: Vertex filtering step (shown in Section 2.2.2)
11: Update the position of each vertex (follows the principle described by Kegl et al. [20])
12: Adjust the shape of the curve
13: if (number of segments meets:  $(is > \beta n^{1/3} \Delta_n(f_{is,n})^{-1/2} r)$ )
14: break
15: else Insert a new vertex
16: else Insert a new vertex
17: else
18: Update the position of each vertex (follows the principle described by Kegl et al. [20])
19: Adjust the shape of the curve
20: if the positions of other vertices are not changed when inserting a new vertex, while the number of newly added vertices is more than five
21: break
22: End for
23: Project the data point  $p_i$  to the closed polygon curve
24: Obtain a closed polygon curve consisting of several segments
25: Obtain the data sequences

```

and (2) vertex filtering. The improved RCKPC method is denoted as [Algorithm 1](#).

2.2.1. Redefined vertex optimization step

We redefined the penalized distance function in the vertex optimization step to improve the efficiency of the algorithm. Compared with the previous vertex optimization step (shown in Eqs. (19)–(21) in Appendix B.2), we used a new penalized function $D(v_i)$ to replace $P(v_i)$, shown as follows:

$$G'_n(v_i) = \Delta_n(v_i) + \lambda_p D(v_i) \quad (1)$$

$$D(v_i) = \frac{1}{iv} \left(\sum_{i=1}^{iv} \Delta(p, v_i) \right) \quad (2)$$

where $\Delta(p, v_i)$ denotes the distance between a data point p and the nearest vertex v . Furthermore, we used the same average squared distance function $\Delta_n(v_i)$ used in the KPC and CKPC methods.

Compared with the function $P(v_i)$, there are three advantages of using function $D(v_i)$. First, it has fast optimization convergence. Second, it decreases the deviation of the principal curve $f(t)$. Third, using addition and the average to calculate $D(v_i)$ is more efficient than calculating $P(v_i)$ in both the KPC and CKPC methods, which use triangle functions.

2.2.2. Vertex filtering

The vertex filtering step is mainly used to clean abnormal vertices that are in the wrong position and avoid producing a distorted principal curve. The flag of vertex cleaning was defined using Eq. (3).

$$flag(v_i) = \begin{cases} 1, & \text{if } \{(l_{m-1} \text{ or } l_m) > r\}, m = 2, \dots, is - 1 \\ 0, & \text{if } \{(l_{m-1} \text{ and } l_m) \leq r\}, m = 2, \dots, is - 1 \end{cases} \quad (3)$$

Algorithm 2 Improved MCSDE-ABPNN method.

```

Input: model parameters
Output: best individual  $pop$  (used as the initial optimal parameters (i.e., weights and thresholds) of ABPNN)
1: Initialize individual  $pop$ , maximum iteration  $Gmax$ , and current iteration  $G$ , where the initialization step of MCSDE is the same as the initialization step in the DE method (shown in Appendix B.3).
Furthermore, the newly initialized objective function  $g(x) = \sum_{i=1}^{Np} pop_i^2$  is used according to a previously described method [38]. In addition, the population size  $N_p$  of MCSDE is computed as
 $N_p = (n + 1) \times q + (q + 1) \times u$ , where  $n$  is the total number of input neurons of the ABPNN,  $q$  is the total number of hidden neurons of the ABPNN, and  $u$  is the total number of output neurons of the ABPNN.
2: While ( $i < N_p$ )
3: Calculate the probability  $pro_G$  when using the mutation operator according to Eq. (5)
4: For each individual  $i = 1$  to  $N_p$  do
5: if ( $rand[0, 1] < pro_G$ )
6: Generate the mutant vector  $vec_i^{G+1}$  using Eq. (4-①)
7: else
8: Generate mutant vector  $vec_i^{G+1}$  using Eq. (4-②)
9: End if
10:  $pop_i^{G+1}$  = new solutions updated using Eqs. (10) and (11)
11:  $npop_i^{G+1}$  is a new combined solution produced by Eq. (22)
12: if ( $rand[0, 1] < CR$ ) using Eq. (23)
13:  $u_i^{G+1} = vec_i^{G+1}$ 
14: else
15:  $u_i^{G+1} = pop_i^G$ 
16: End if
17: if ( $g(u_i^{G+1}) < g(pop_i^G)$ ) using Eq. (24)
18:  $pop_i^{G+1} = u_i^{G+1}$ 
19: else
20:  $pop_i^{G+1} = pop_i^G$ 
21: End if
22: if ( $g(npop_i^{G+1}) < g(npop_i^G)$ )
23:  $pop_i^{G+1} = npop_i^{G+1}$  and  $vec_i^{G+1} = npop_i^{G+1}$ 
24: End if
25: Update  $F$  and  $CR$  according to Eqs. (6)–(9)
26: End for
27:  $i = i + 1$ ; Remember the global best individual  $pop_{best}$ 
28: End while
29: Use the global best individual  $pop_{best}$  values as the optimal weight and threshold of the ABPNN
30: Start ABPNN training
31: Store the model parameters of the optimal ABPNN
32: End

```

The vertex was kept if $flag(v_i) = 1$ and deleted if $flag(v_i) = 0$. Meanwhile, r is the data radius that determines the data scaling (shown in Appendix B.2.1), and l_m is the length of the m th line segment.

In addition, to avoid generating a distorted principal curve, the newly generated vertex, v_i , located outside the radius of the data points must be deleted. Furthermore, to ensure that the newly generated vertex can be projected by as many data points as possible, if the number of data points projected onto this vertex and the neighboring segment consisting of v_i and the nearest vertex, v_{i+1} , decreased to zero, vertex v_i was deleted.

2.3. MCSDE-ABPNN method

To address the issue that NN is always easily constrained by the local minimum during training, we used MCSDE to identify the optimal initial parameters (i.e., weights and thresholds) of the ABPNN.

2.3.1. Overall design

First, the numbers of input, hidden, and output neurons of the ABPNN were used to initialize the population size N_p of MCSDE, as shown in the initialization step of [Algorithm 2](#). The initial weight and threshold of the ABPNN were then used to update the mutant

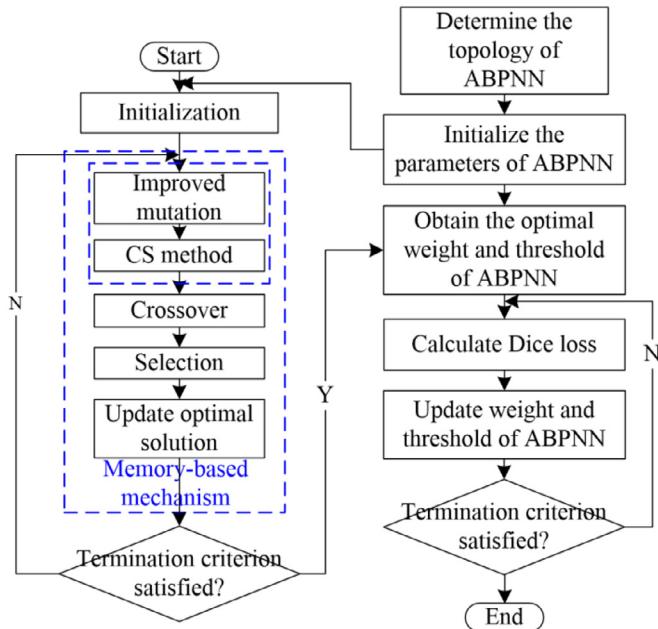


Fig. 3. Flowchart of the MCSDE-ABPNN method. CS, cuckoo search; MCSDE, memory-based, cuckoo search-based differential evolution; ABPNN, adaptive learning-rate backpropagation neural network.

vectors vec_i^{G+1} (shown in Eq. (4)). Finally, the outputs of MCSDE were used as the optimal parameters (i.e., weight and threshold) of the ABPNN Fig. 3. shows the flowchart of the MCSDE-ABPNN method. In addition, the details of the MCSDE method are presented as Algorithm 2.

3.2.3. MCSDE method

Generally, a good balance between a global and local search is important for improving the performance of an algorithm [32]. The cuckoo search (CS) method is good at searching for a promising region, but it is not good at fine-tuning at the end of the search step [32]. The DE method or a related method is suitable for local searches, but it does not memorize good results or use global information about the search space [39]. To improve the ability of the DE method to memorize and utilize the best results obtained in the previous cycle, we used a memory-based differential evolution (MDE) method. To improve the ability of global searching, a new MCSDE method combining the CS method with the MDE method was developed.

MDE method. Based on a previously described method [33], we used a memory-based mechanism to store the best mean mutation factor (uF) and the mean crossover rate (uCR) from the previous cycle, and then used these values as the initial values for the next cycle. The main steps are shown below:

(i) In the mutation step, two mutation operators were used to generate mutant vectors vec_i^{G+1} , as follows:

$$\text{vec}_i^{G+1} = \begin{cases} [w, a] \times \text{pop}_{i_1}^G + F \times (\text{pop}_{i_2}^G - \text{pop}_{i_3}^G), & \text{if } \text{rand}[0, 1] \\ < \text{pro}_G(1) \\ [w, a] \times \text{pop}_{i_1}^G + F \times (\text{pop}_{i_2}^G - \text{pop}_{i_3}^G) + F \\ \times (\text{pop}_{i_4}^G - \text{pop}_{i_5}^G), & \text{otherwise}(2) \end{cases} \quad (4)$$

where $[w, a]$ shows the matrix of the initial weight and the threshold of the ABPNN, i_k ($k = 1, 2, 3, 4, 5$) represents five random integer numbers in the range of $[1, N_p]$, F is the mutation factor in the range of $[0, 1]$, and $\text{rand}[0, 1]$ is a random number in the range of $[0, 1]$. pro_G indicates the probability of using the mutation operator

and was calculated as follows:

$$\text{pro}_G = \text{pro}_{\min} + \frac{G \times (\text{pro}_{\max} - \text{pro}_{\min})}{G_{\max}} \quad (5)$$

where pro_{\max} and pro_{\min} are the maximal and minimal probability of using the mutation operator, respectively, with both values in the range of $[0, 1]$.

(ii) The crossover step increased the variability of the parameter vector by mixing the mutated vectors vec_i^{G+1} and the target vectors pop_i^G , which is the same crossover step used in the DE method (shown in Eq. (23) in Appendix B.3).

(iii) In the selection step, the trial individual (offspring) produced by the crossover operator was compared with the target individual (parent). The details of the selection step are shown in Eq. (24) in Appendix B.4.

(iv) In the updating step, F and CR were updated as follows:

$$F = (1 - \text{val}) \times F + \text{rand}[0, 1] \times \text{mean}_L(S_F) \quad (6)$$

$$CR = (1 - \text{val}) \times CR + \text{rand}[0, 1] \times \text{mean}_L(S_{CR}) \quad (7)$$

where S_F and S_{CR} denote the successful mutation probabilities and successful crossover probabilities, respectively. The Lehmer mean $\text{mean}_L(\bullet)$ [33] was used to update the F and CR values using Eqs. (8) and (9).

$$\text{mean}_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F} \quad (8)$$

$$\text{mean}_L(S_{CR}) = \frac{\sum_{CR \in S_{CR}} CR^2}{\sum_{CR \in S_{CR}} CR} \quad (9)$$

CS method. The CS method [40] is a nature-inspired algorithm influenced by the obligate brood parasitism of some cuckoo species that lay their eggs in the nests of other host birds. In the CS method, each egg can be regarded as a solution. In the initial process, each solution was generated randomly. When generating the i th solution in the $(G+1)$ th generation, denoted by pop_i^{G+1} , a levy flight was performed as follows:

$$\text{pop}_i^{G+1} = \text{pop}_i^G + \alpha \oplus \text{Levy}(ap) \quad (10)$$

where α is the step size related to the size of the problem of interest, and ap is the adjusted parameter. We set $\alpha = 1$ based on data from previous studies [41]. The product \oplus represents entry-wise multiplications. Levy flight essentially provides a random walk while random steps are drawn from a Levy distribution for large steps.

$$\text{Levy}(ap) = G^{-ap}, 1 \leq ap \leq 3 \quad (11)$$

Two different methods were integrated to produce a new solution with the MCSDE method. The first component was the Levy flight in the CS method Eqs. (10) and ((11)), and the second component was the improved mutation operation in the MDE method (Eq. (4)). A new solution $n\text{pop}_i^{G+1}$ was generated using Eq. (12).

$$n\text{pop}_i^{G+1} = \text{rand}[0, 1] \times \text{pop}_i^{G+1} + (1 - \text{rand}[0, 1]) \times \text{vec}_i^{G+1} \quad (12)$$

3.2.3.3. ABPNN method

We used an ABPNN [31] with three layers (i.e., input, hidden, and output layers) as a fully-connected neural network that is able to express any continuous function [42]. The data sequences obtained using the RCKPC method were used in the ABPNN, in which data sequences consisted of ordered projection indices and the coordinates of the data seed points. The projection index was used as the input of the ABPNN, and the coordinates of the data seed points were used to minimize Dice loss during training.

The output layer included two units, corresponding to $c(x)$ and $c(y)$, which were treated as the continuous functions $c(x(t))$ and

$c(y(t))$, respectively, on the projection index, t . We then used the gradient descent method [43] to update the weights in the back-propagation stage.

In contrast to our previous study [23], here we used the sigmoid activation function, $h_1 = 1/(1 + e^{-x})$, from the input layer to the hidden layer. The sigmoid activation function was used because it ensures that the final output is normalized to $[0, 1]$, which helps prevent the model from being affected by different input sizes. The main improvements in this method include: (1) the use of the ABPNN to smooth the contour and (2) the use of the Gaussian activation function, $h_2 = (e)^{\gamma(-x^2)}$, from the hidden layer to the output layer, rather than the sigmoid activation function for more rapid convergence of the model.

2.4. Explainability-guided mathematical model of the prostate contour

The main innovation in this component of the study was that we used a mathematical map model (realized by the ABPNN) to generate a smooth prostate contour, represented by the outputs of ABPNN (i.e., optimized vertices), to match the GT contour. After training the ABPNN, a new explainability-guided mathematical map model was developed to generate a smooth prostate contour, $f(t)$, as shown below:

$$f(t) = (x(t), y(t)) = \left(\frac{1}{1 + e^{-\sqrt{-\ln c(x(t))}}}, \frac{1}{1 + e^{-\sqrt{-\ln c(y(t))}}} \right) \quad (13)$$

where $x(t)$ and $y(t)$ represent the x-axis and y-axis coordinates of the obtained vertices, respectively. Furthermore, the value of the output units $c(\bullet)$ was calculated using Eq. (14).

$$\begin{aligned} & (c(x(t)), c(y(t))) \\ &= \left(e^{-\left(\sum_{i=1}^q \left(\frac{1}{1+e^{-(twl_i-a1_i)}} \right) w2_{i,1}-a2_{i,1} \right)^2}, e^{-\left(\sum_{i=1}^q \left(\frac{1}{1+e^{-(twl_i-a1_i)}} \right) w2_{i,2}-a2_{i,2} \right)^2} \right) \end{aligned} \quad (14)$$

Eq. (14) is a modification of Eq. (25), and the parameters used in Eq. (14) were the same as those used in Eq. (25) (Appendix B.4).

2.5. Materials

We systematically evaluated the developed method on TRUS prostate images from 55 brachytherapy patients (a total of 393 slices) collected from the Nanjing University of Chinese Medicine, Nanjing, China. The TRUS images were captured using a Vinno 70 Lab digital ultrasound diagnostic instrument (Vinno, Suzhou, China) and an integrated TRUS probe with a frequency in the range of 4–8 MHz. The ages of the patients ranged from 18 to 75 years, with a mean age of 52. To minimize the requirement for manual labeling by physicians, only up to 10 slices from each patient were used for evaluation. The dataset was in Digital Imaging and Communications in Medicine format, and the original resolution of each TRUS image was 1200 pixels × 900 pixels.

In the following experiments, the following common metrics in medical image segmentation [44] were considered when evaluating the methods:

■ DSC:

$$DSC = \frac{2TP}{2TP + FP + FN} \quad (15)$$

■ Jaccard similarity coefficient (Ω):

$$\Omega = \frac{TP}{FP + TP + FN} \quad (16)$$

Table 2

Evaluation results for different levels of Gaussian noise.

	DSC (%)	Ω (%)	ACC (%)
Clean set	95.8	94.3	95.4
$\sigma = 10$	94.4	93.1	94.1
$\sigma = 20$	94.2	92.9	94
$\sigma = 30$	93.9	92.5	93.8
$\sigma = 50$	93.3	91.9	93

DSC, Dice similarity coefficient; Ω , Jaccard similarity coefficient; ACC, accuracy; σ , standard deviation.

■ Accuracy (ACC):

$$ACC = \frac{TP + TN}{TP + FN + FP + TN} \quad (17)$$

where TP , FP , FN , and TN represent the number of true positive, false positive, false negative, and true negative results, respectively.

We randomly selected 30 patients for training (215 slices), 10 patients for validation (70 slices), and the remaining 15 patients for testing (108 slices). We used a small number of radiologist-defined seed points as the prior points. To decrease the delineation workload of the radiologists, all prostate TRUS images were resized to a resolution of 600 pixels × 400 pixels. All GTs were marked and verified by three board-certified radiologists. Each expert radiologist independently verified their own marks and the anonymous marks of the other radiologists. Consensus GTs were achieved using the majority of three experts' annotations Fig. 4. shows the workflow used for extracting the GT. All experiments were conducted on a computer with an Intel Core i7-8750H central processing unit, a GeForce GTX 1070 graphics processing unit, and 8 GB of memory.

3. Results

3.1. Evaluation on the testing set

In this section, we discuss the model performance on the testing set using the determined optimal model, as described in Appendix A (10 neurons and 1000 epochs). The testing set contained 15 patients Fig. 5. shows the evaluation of each patient. Due to the possibility that the images from each patient may contain more than one slice, the average values were used to evaluate each patient. As seen in Fig. 5, the H-ProSeg method showed good performance, regardless of the evaluation parameter used. All DSC values were greater than 0.95, and 80% were greater than 0.955. The ACC values for 60% of the images were greater than 0.955, and the Ω values were greater than 0.94 for 86.6% of the radiographs. The average DSC, Ω , and ACC values were 95.8%, 94.3%, and 95.4%, respectively. The results of qualitative testing are presented step by step in Fig. 6.

3.2. Model robustness to Gaussian noise

We used different levels of Gaussian noise to corrupt the testing set, and then used the corrupted testing set to determine the effect of Gaussian noise on the quantitative and qualitative test results. After corruption with Gaussian noise, the pixel values of each testing image were affected, and the prostate contours in the test images became blurry or even missing. This made it more challenging to obtain an accurate segmentation result. To test the robustness of our method, we set the standard deviation (σ) of the Gaussian function to 10, 20, 30, and 50. The average segmentation results are shown in Table 2, and the corresponding result in one randomly selected slice is shown in Fig. 7.

Different levels of Gaussian noise may lead to different effects on the segmentation results. In many studies, the optimal σ of

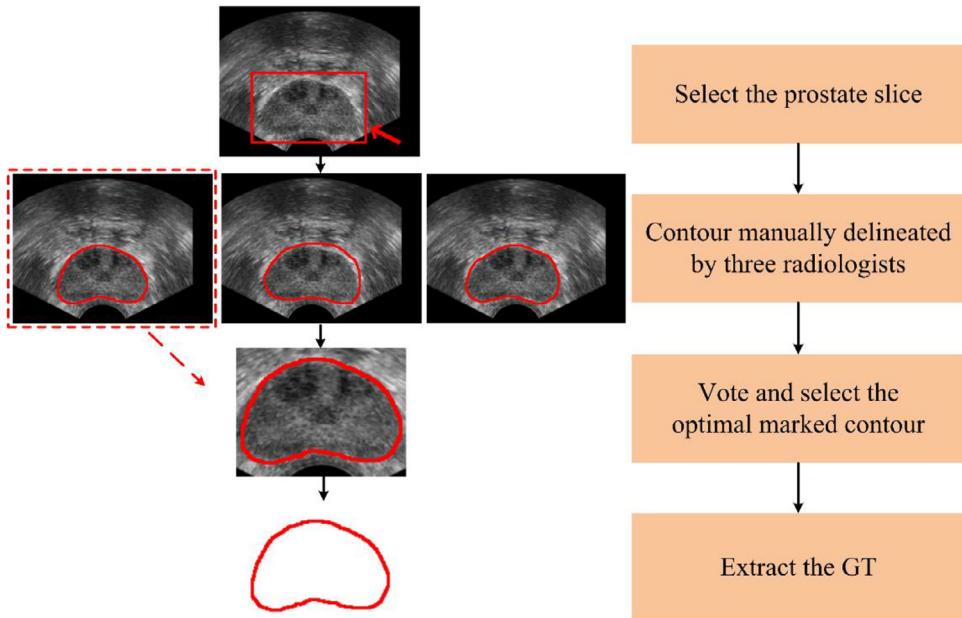


Fig. 4. The workflow for extracting the ground truth (GT).

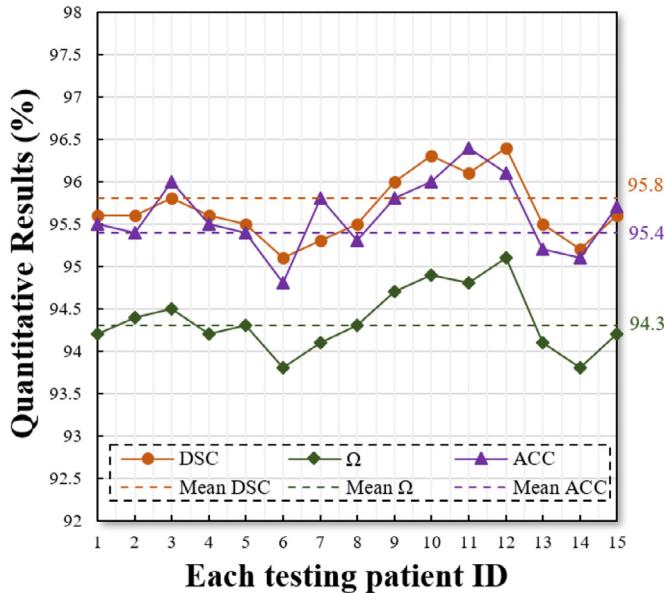


Fig. 5. Evaluation of the entire testing set using different evaluation metrics. The solid line shows the average value for each patient and the dotted line shows the average value for all patients. DSC, Dice similarity coefficient; Ω , Jaccard similarity coefficient; ACC, accuracy.

the Gaussian function has been chosen by repeated trials. For instance, Zhang et al. [45] proposed a deep learning method named BigAug for prostate segmentation. They tested their method on the Medical Segmentation Decathlon-Prostate dataset [46] and found that the σ in the image quality step and spatial configuration step ranged from [10,30] to [10,13], respectively. Arce-Santana et al. [47] proposed a probabilistic active contour region-based method for multiclass medical image segmentation and set the σ in the range of [2,16]. We added additional Gaussian noise (e.g., by using a larger standard deviation) to generate a more distorted image. Meanwhile, we used another metric for evaluation, the overlapped region (overlap) [48], calculated as:

$$\text{overlap} = \frac{|\text{redR} \cap \text{blueR}|}{\text{redR}} \quad (18)$$

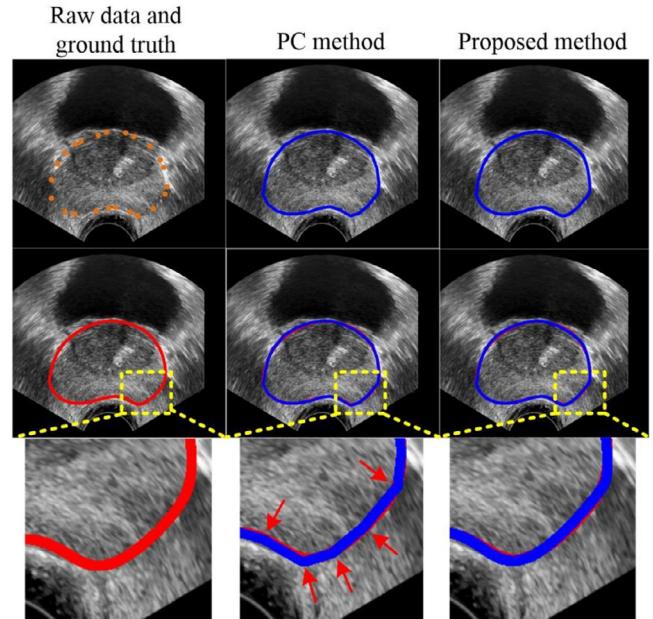


Fig. 6. Intermediate qualitative results. The red arrow indicates several vertices in the principal curve-based contour. PC, principal curve.

where the overlap rate (overlap) indicates the proportion of overlap (overlapR) between red (redR) and blue (blueR) regions.

As shown in Table 2 and Fig. 7, as the σ increased from 10 to 50, the overlap rate decreased from 0.899 to 0.799. This indicated that the raw images were severely affected by white noise (gray label = 255). Although the overlap rate decreased by approximately 20% (overlap decreased to 0.799), the values of DSC, Ω , and ACC fluctuated by a maximum of approximately 2.5%, demonstrating the high robustness of our method. Furthermore, the average values for DSC, Ω , and ACC decreased when incorporating different levels of Gaussian noise, but all average values remained greater than 90%, which further confirmed that our method effectively handled noisy data.

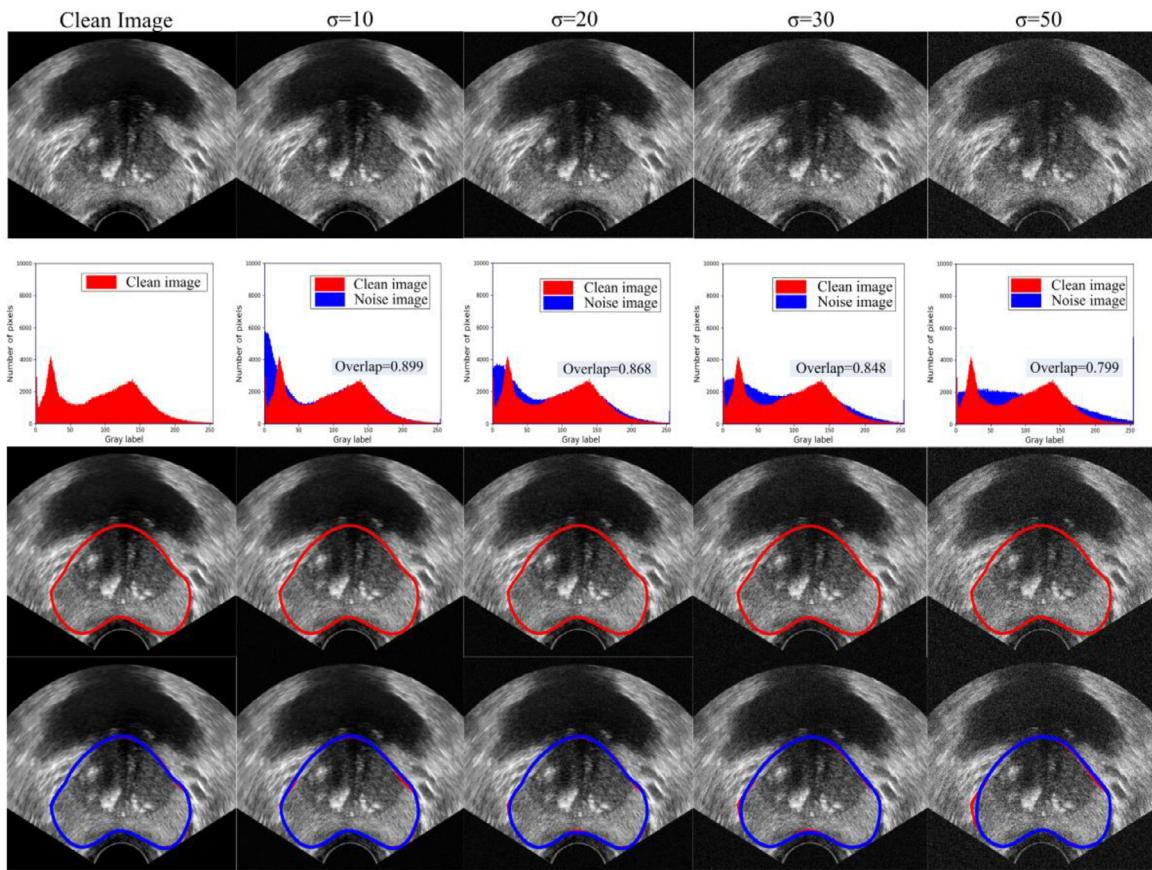


Fig. 7. Robustness of the proposed H-ProSeg method to different levels of Gaussian noise. The top row shows the original clean image (leftmost) and the images corrupted by different levels of Gaussian noise (standard deviation $[\sigma] = 10, 20, 30$, and 50). The second row shows the histogram of the initial images (first row), with red indicating the histogram of the clean image and blue indicating the histogram of the noisy image. The third row shows the ground truth on the initial images (first row), and the fourth row shows the corresponding segmentation results.

3.3. Competing algorithms

To further investigate the performance of the H-ProSeg method, we compared it qualitatively and quantitatively with three other hybrid algorithms. All the tested algorithms, including H-ProSeg, were principal curve-based and machine learning-based methods. Furthermore, all of the previously mentioned evaluation metrics were used. The architectures of the tested algorithms are summarized below,

- Hybrid Algorithm 1 combined the CKPC-BPNN method [23] with the MDE method (shown in Section 3.3.2).
- Hybrid Algorithm 2 was the CKPC-MCSDE-BPNN method. Compared with hybrid Algorithm 1, hybrid Algorithm 2 was improved by incorporating the technique proposed by Cobos et al. [40].
- Hybrid Algorithm 3 was the CKPC-MCSDE-ABPNN method. It was based on hybrid Algorithm 2, but improved by combining this algorithm with the ABPNN.

All of the tested algorithms were semiautomatic, taking advantage of the ability of the principal curve method to pass through the “middle” of the dataset automatically and then use parameters of the machine learning-based method to denote an appropriate map function to generate a smooth prostate contour that matches the GT contour. As the BPNN-based method had three layers (i.e., input, hidden, and output layers) as a fully connected neural network that expressed any continuous function [42], we also used a three-layered architecture for all of the competing algorithms. Furthermore, the sigmoid activation function from the input layer

to the hidden layer and the Gaussian activation function from the hidden layer to the output layer were used in the competing algorithms. To compare performance under the same conditions, all of the competing algorithms used the same training, validation, and testing sets. We chose 10 neurons and 1000 epochs to evaluate all of the models. Based on our previous data [23], we set the learning rate of both hybrid Algorithm 1 and hybrid Algorithm 2 at 0.4. As both hybrid Algorithm 3 and the H-ProSeg algorithm select the learning rate adaptively, it was only necessary to set the initial learning rate for these algorithms, with a constant value of 0.5.

To evaluate the performance of our algorithm, we tested all of the hybrid algorithms on the entire testing set. The quantitative results for all of the hybrid algorithms are shown in Table 3, along with the results of H-ProSeg presented in the previous section. We also chose six representative prostate TRUS results for a visual comparison, as shown in Fig. 8.

Overall, the H-ProSeg algorithm was capable of accurately extracting the prostate contours.

3.4. Comparison of point-guided methods using a different number of prior points

To further evaluate the performance of our method, we re-selected seed points closer to the shadowed regions as the prior points. Table 4 shows the quantitative comparison of point-guided methods using DSC as a metric, after selecting less than 1%, 5%, 8%, 10%, 15%, 30%, or 50% of points of contours manually delineated by radiologists as the prior points. As in the previous experiments (Section 3.1–3.3), we used less than 8% of contour points as

Table 3
Quantitative comparison among different hybrid algorithms.

Methods	DSC (%)	Ω (%)	ACC (%)	Hausdorff distance (mm)
Hybrid Algorithm 1	91.7 + 5.6	90.3 + 7.1	91.3 + 6	3.9 ± 3.9
Hybrid Algorithm 2	92.9 + 3.8	91.5 + 4.2	92.6 + 3.6	3.2 ± 1.8
Hybrid Algorithm 3	93.8 + 4.1	92.3 + 4.1	93.5 + 3.7	3.4 ± 2.3
Proposed Algorithm	95.8 ± 2.6	94.3 ± 3.1	95.4 ± 2.5	1.5 ± 1.1

DSC, Dice similarity coefficient; Ω , Jaccard similarity coefficient; ACC, accuracy.

Table 4
DSC values of seed point-guided methods using different percentages points as a prior.

Method	<1% of points	<5% of points	<8% of points	<10% of points	<15% of points	<30% of points	<50% of points
CSIM	80 ± 8.1	83.6 ± 6.1	85.3 ± 4.2	87.1 ± 3	91.7 + 2.1	94.2 ± 0.9	98.5% ± 0.2
U-Net	87.3 ± 6.3	90.1 ± 5.1	90.8 ± 5.6	91 ± 4.2	91.6 ± 4.1	96.2 ± 3.7	97.1 ± 1.3
H-ProSeg algorithm	92.9 ± 4.2	94.6 ± 3.5	95.5 ± 3.1	96 ± 2.5	96.9 ± 1.7	97.5 ± 0.9	98.2 ± 0.4

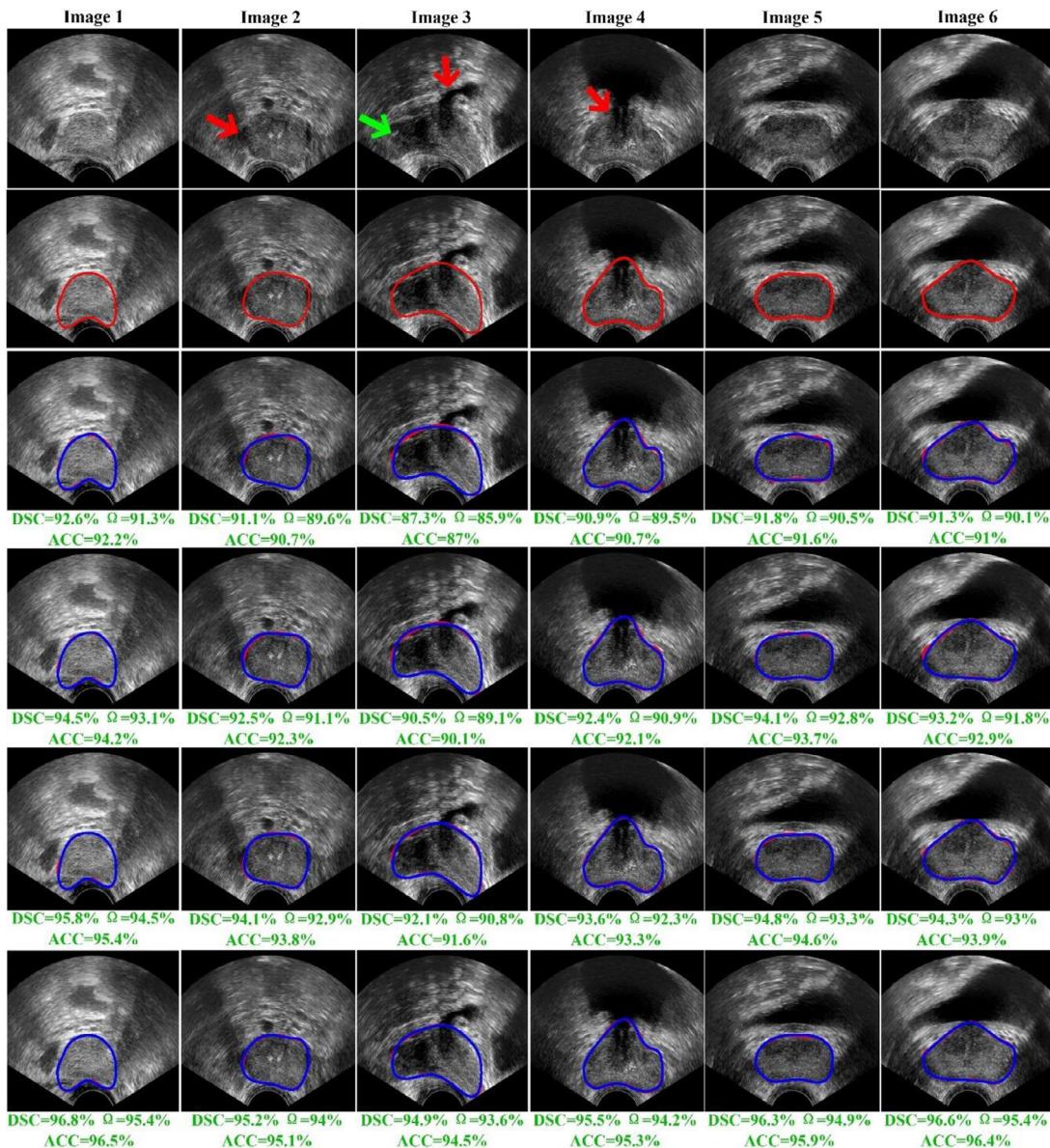


Fig. 8. Visualization of prostate segmentation results. The red and green arrows indicate shadow artifacts and prostatic calcification, respectively. The red and green arrows indicate shadow artifacts and prostatic calcification, respectively. The first row shows raw prostate TRUS images. The second row shows the ground truth (GT) delineated by the radiologists. The last four rows show the experimental results of the four models, hybrid Algorithms 1, 2, and 3 and H-ProSeg, respectively. The red lines indicate the GT, and the blue lines indicate the segmentation result. Each quantitative segmentation result is also presented. DSC, Dice similarity coefficient; Ω , Jaccard similarity coefficient; ACC, accuracy.

Table 5
Quantitative comparison with other state-of-the-art algorithms.

Paper	Method	Model	DSC (%)	Ω (%)	ACC (%)	Hausdorff distance (mm)	Testing time (s)
[14]	Hull-CKPC	Hybrid	94.6 \pm 4.3	93.2 \pm 5.7	94.5 \pm 4.1	3.5 \pm 1.3	4
[50]	U-Net	Deep learning	90.5 \pm 10.6	89.9 \pm 13.1	90.4 \pm 11.2	4.3 \pm 2.8	4
[51]	Mask-RCNN	Deep learning	91.9 \pm 9.6	90.6 \pm 11.9	91.3 \pm 9.1	4.6 \pm 1.9	5
[52]	U-Net++	Deep learning	93.2 \pm 8.1	92.6 \pm 10.3	92.7 \pm 7.7	3.9 \pm 1.5	5
Current	H-ProSeg	Hybrid	95.8 \pm 2.6	94.3 \pm 3.1	95.4 \pm 2.5	1.5 \pm 1.1	4

DSC, Dice similarity coefficient; Ω , Jaccard similarity coefficient; ACC, accuracy; closed K-segment principal curve U-Net, unity networking; RCNN, region-based convolutional neural network.

the prior points. However, in contrast, the prior points used in this experiment were closer to the shadowed regions. The DSC value showed a slight decrease ($95.5 \pm 3.1\%$) compared with the previous results ($95.8 \pm 2.6\%$), but fluctuation was not apparent.

Furthermore, we compared our method with the cubic spline interpolation model (CSIM) [49] and the unity networking (U-Net) method [50], with all methods using the same prior points. As seen in Table 4, when less than 1% of contour points were used as the prior points, the DSC value of the U-Net method was $87.3 \pm 6.3\%$. However, after increasing the number of seed points, the performance of the U-Net method improved. In addition, the CSIM heavily depends on the selection of seed points. When the number of seed points increased (from < 1% of contour points to < 50% of contour points), the average DSC value of the CSIM increased by as much as 23%.

Overall, our method had the best performance.

3.5. Performance comparison with state-of-the-art algorithms

To confirm the performance of our method, we compared it with state-of-the-art algorithms, including two categories of algorithms: hybrid and deep learning-based algorithms:

- Hybrid algorithms [14]: Peng et al. [14] presented the Hull-CKPC method for lung segmentation in chest X-ray images and validated its performance on several public datasets.
- Deep learning-based algorithms [50–52]: Ronneberger et al. [50] developed the U-Net method for biomedical image segmentation, and He et al. [51] developed the mask region-based convolutional neural network (Mask-RCNN) model for object segmentation. Zhou et al. [52] developed the unity networking plus (U-Net++) for multi-organ segmentation.

Deep learning-based algorithms are fully automatic, whereas hybrid algorithms are semiautomatic, as they use a small number of radiologist-defined seed points as the prior points. Hybrid models give comparable or better segmentation results than deep learning models, which indicates that a hybrid model consisting of a principal curve model with a machine learning model has a strong ability to fit the dataset accurately.

As deep learning-based algorithms are affected by the amount of training data [53], we used the rotation method for data augmentation to improve the performance of the deep learning-based algorithms. Based on the raw training data (215 slices), we increased the training data set to 2000 slices. All of the training scans were randomly rotated in the range of $[-15^\circ, 15^\circ]$ until they reached the expected number of training data (2000 slices). Furthermore, all of the deep learning-based algorithms use the combo loss function [54] for training, and the learning rate was reduced on a plateau with a patience of 10 and early stopping with a patience of 50 for a maximum of 1000 epochs.

The results summarized in Table 5 show that the H-ProSeg method had the best performance for prostate segmentation of all the algorithms tested.

4. Discussion

Accurate and robust prostate segmentation is challenging due to the high variability of the prostate anatomical structures. Shape and contour-based methods have recently been proposed as the most appropriate segmentation methods for the prostate [10,14]. Based on the importance of interpretable artificial intelligence technologies in the medical image processing field [55,56], we developed and evaluated a hybrid H-ProSeg method using principal curve-based and differential evolution neural network-based methods for prostate segmentation in ultrasound images. In this section, we discuss the entire study from different aspects.

The H-ProSeg method: The advantage of using the H-ProSeg method for prostate segmentation in ultrasound images is that it produces accurate and robust results, as shown in Section 3. The H-ProSeg method uses the ability of the principal curve-based method to automatically approach the center of the dataset [29,20] and the interpretable neural network-based method for training to decrease the model error [23]. However, the segmentation results of principal curve-based methods are not sufficiently smooth [57]. To address this issue, we developed a smooth mathematical model of the prostate contour, explained by the parameters of the ABPNN, using an improved differential evolution-based model to assist in identifying the optimal ABPNN.

The H-ProSeg method (with and without augmentation/different subregions): To evaluate the effect of data size on different subregions (i.e., apex, mid-gland, and base subregions), we re-analyzed data from 25 patients, including 100 apex slices and base slices. We added 100 slices for training, 50 slices for validation, and 50 slices for testing to the previous data set from the mid-gland region (393 mid-gland slices, 215 slices for training, 70 slices for validation, and 108 slices for testing). The number of apex slices and base slices were equally divided among the training, validation, and testing sets. Furthermore, we increased the training data (315 slices) to 2000 slices using a rotation step with a rotated angle randomly selected from the range of $[-15^\circ, 15^\circ]$. After augmentation, the DSC, Ω , and ACC values were 97.2%, 95.6%, and 96.6%, respectively. By comparison, when the data augmentation step was not used (DSC = 95.8%, Ω = 94.3%, and ACC = 95.4%), the DSC, Ω , and ACC values increased by approximately 1.46%, 1.27%, and 1.25%, respectively. Furthermore, we added representative qualitative comparisons of different subregions, as shown in Fig. 9. Overall, the performance of our method improved after data augmentation.

Table 6 shows the results of our method (without augmentation) after adding the apex and base slices. The results for the average total DSC and ACC values presented in Table 5 (DSC = 95.8 ± 2.6 and ACC = 95.4 ± 2.5), were in agreement with those proposed in Table 6 (DSC = 95.8 ± 2.6 and ACC = 95.4 ± 2.5). The main difference between these experiments was that the experiment used to generate the data in Table 5 only used mid-gland slices for training, whereas the experiment used to generate the data in Table 6 added apex and base slices for training.

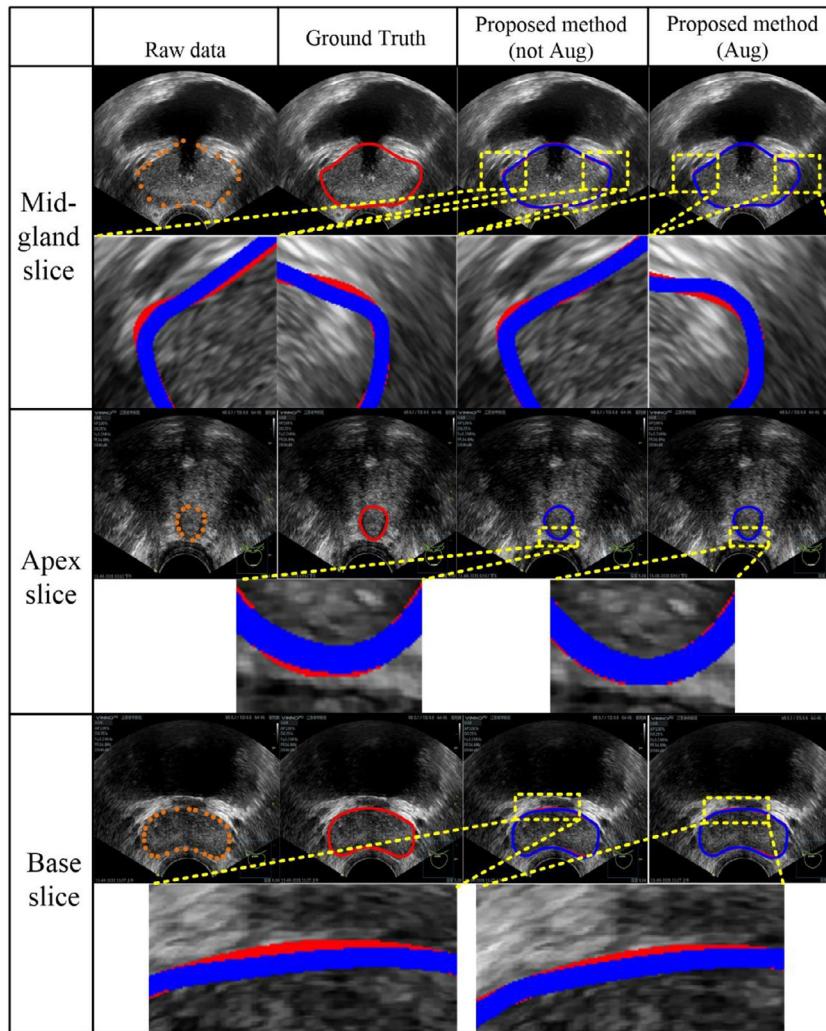


Fig. 9. Qualitative comparison of three different representative slices (mid-gland, apex, and base slices). The first two rows show the mid-gland slice and the corresponding partial magnification region, respectively. The second two rows show the apex slice with its corresponding partial magnification region. The last two rows show the base slice with its corresponding partial magnification region. The orange points indicate the seed points, the red curve indicates the ground truth (GT), and the blue curve indicates the experimental result. The most voted contour (best GT) was used as the GT.

While the average total DSC and ACC values in Tables 5 and 6 are the same, the standard deviations of both metrics (DSC and ACC) were higher for the data presented in Table 6. A possible reason for this finding is that the experiments represented in Table 6 used a diverse data set (i.e., apex, mid-gland, and base slices), which increased the difficulty of prediction and affected the stability of the model.

H-ProSeg method (cases with damaged prostate boundary): Accurate prostate segmentation in TRUS images is a challenging task due to shadowing caused by calcification, which results in regions of the boundary not being visible. In Fig. 8 (Image 3), we show a case affected by calcification. However, the prostate contour was not damaged in this case, because the prostate is only damaged when the region of calcification is sufficiently large. However, such cases are rare. We further assessed our model by adding a case in which the contour was missing because of shadowing due to calcification, as shown in Fig. 10.

The H-ProSeg method and competing algorithms: All of the algorithms incorporating the H-ProSeg method consisted of principal curve-based and neural network-based methods. As seen in Fig. 8, when the same method was used to test different slices, the results showed a large degree of variation, mainly because the prostate is often severely affected by calcification and shadow ar-

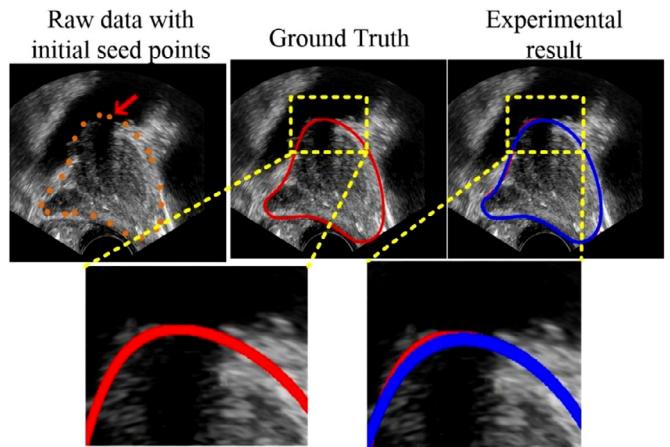


Fig. 10. Missing prostate contour as a result of shadowing due to calcification. The red arrow indicates the missing prostate boundary.

tifacts [58]. However, all of the methods incorporated the ability of the principal curve to fit the dataset well [22,59], resulting in good segmentation performance (all metrics were greater than

Table 6

Evaluation on different regions of the prostate using different ground truth. GT1, GT2, and GT3 represent the ground truths (GTs) that were manually delineated by three different radiologists. The best GT represents the most voted contour as the GT for each case. The DSC (total) was calculated for different regions (i.e., apex, mid-gland, and base regions), using the best GT. ACC, accuracy.

	Prostate region	GTs	DSC (%) (Each GT at different regions)	DSC (%) (Total)	ACC (%) (Each GT at different regions)	ACC (%) (Total)
H-ProSeg method (no augmentation)	Apex region	GT1	95.3 ± 4.7	95.8% ± 3.5	95.2 ± 3.9	95.4% ± 3.7
		GT2	96 ± 4.1		95.5 ± 4.1	
		GT3	95.8 ± 3.9		95.4 ± 3.7	
		Best GT	95.7 ± 3.5		95.3 ± 3.4	
		GT1	96.3 ± 3.5		96.1 ± 4.3	
	Mid-gland region	GT2	96.4 ± 3.6		96.2 ± 3.9	
		GT3	96.1 ± 3.3		95.6 ± 3.9	
		Best GT	96.2 ± 3		95.9 ± 3.5	
		GT1	95.3 ± 5.1		95.1 ± 5.7	
		GT2	95.5 ± 5.8		95.4 ± 6.2	
H-ProSeg method (with augmentation)	Apex region	GT3	95.6 ± 4.9		95.2 ± 4.9	
		Best GT	95.4 ± 4.1		95.1 ± 4.1	
		GT1	96.6 ± 4.7	97.2% ± 2.9	96.3 ± 4	96.6% ± 3.1
		GT2	97.1 ± 4.6		96.8 ± 4.9	
		GT3	97.2 ± 3.9		97 ± 4.2	
	Mid-gland region	Best GT	97.2 ± 3.4		96.6 ± 3.1	
		GT1	97.2 ± 3.5		97 ± 3.6	
		GT2	97.6 ± 4.1		97.5 ± 3.9	
		GT3	97.3 ± 3.6		97.1 ± 3.5	
		Best GT	97.5 ± 2.2		97.1 ± 2.5	
	Base region	GT1	95.9 ± 4.8		95.5 ± 4.7	
		GT2	96.4 ± 4.4		96.3 ± 4.1	
		GT3	95.7 ± 4		95.6 ± 3.7	
		Best GT	96.8 ± 3.5		96 ± 3.4	

90%). Furthermore, in contrast to the H-ProSeg method, the competing algorithms only used the CKPC method to obtain the data sequence and ignored the influence of abnormal data, which resulted in the values of all metrics decreasing by as much as 4.1% ([Table 3](#)). Meanwhile, the differential evolution neural network-based method also increased the performance of the H-ProSeg method.

The H-ProSeg method and state-of-the-art methods: In this section, we discuss the following aspects: (1) the impact of seed points, (2) computational efficiency, and (3) the degree of difficulty with ultrasound prostate tasks.

Impact of seed points: Here, we mainly discuss the effect of using seed points on the performance of the U-Net method. Compared with the DSC value when seed points were not used (90.5 ± 10.6, [Table 5](#)), the DSC value obtained with the U-Net method using less than 1% of manually delineated contour points as the prior points was lower, at 87.3 ± 6.3% ([Table 4](#)). The main reason for this result is that some seed points located in the shadowed regions affected the prediction made by the U-Net method. However, the robustness was improved, which shows that the introduced prior points assisted and guided the U-Net prediction.

Computational efficiency view: In contrast to hybrid algorithms, deep learning-based algorithms are fully automatic. However, the performance of deep learning-based algorithms depends on the amount of training data [\[60\]](#). Due to the limited amount of data used in our study, we required the use of rotation as the data augmentation step [\[61\]](#) to increase the amount of training data. However, this data augmentation strategy is known to cause significant variance in end performance and can be challenging to select [\[62\]](#). In addition, the testing times of all the methods compared in [Table 5](#) were approximately 4–5 s, but the total execution times differed between models. The Mask-RCNN (1.5 days) and U-Net++ (1 day) methods required the greatest amount of training

time, whereas the U-Net method had approximately 3 h of training. The other hybrid methods (Hull-CKPC and H-ProSeg) spent approximately 1.5 h on training.

Degree of difficulty degree with ultrasound prostate tasks: As presented in [Section 3.5](#), the hybrid algorithms obtained good performance using a limited training data set. Furthermore, the Hull-CKPC method [\[14\]](#) has previously been tested on lower-resolution images from the Japanese Society of Radiological Technology lung database [\[63\]](#). Lower image resolution is known to lead to greater segmentation accuracy [\[64\]](#). However, the Hull-CKPC method was evaluated in this study and showed decreased accuracy metrics (i.e., DSC and Ω) compared with those previously reported, which demonstrates that prostate segmentation in ultrasound images is more challenging.

Future work: There are potential ways to improve the performance of the H-ProSeg model in future studies. Our proposed framework contains three key steps that increase the memory burden for prostate segmentation. Hence, model compression may be a good choice for real-time clinical applications [\[65\]](#). We used TRUS images with a constant resolution for our experiments. In the future, we plan to test the robustness of our method at various image resolutions and validate the accuracy of our method at a higher resolution. Considering that we only tested our framework on TRUS images, we will further validate our method using various imaging modalities, i.e., computed tomography and MRI images. Segmenting both the clinical target volume and the planning target volume in every TRUS image is necessary for radiation treatment planning. We will further investigate our framework in future studies. Due to the irregular and unpredictable motion of the prostate, there is some degree of difference in the shape or location of the prostate scanned in the same patient on different days. We aim to further evaluate the performance of our method under such circumstances.

5. Conclusions

Here, we propose a novel hybrid framework, termed H-ProSeg, for accurate and robust prostate segmentation in TRUS images. The three innovations of our method are (1) an improved principal curve method, (2) an improved differential evolution machine learning method, and (3) an explainability-guided mathematical expression of the prostate contour. To the best of our knowledge, we are the first group to propose a hybrid model that combines an improved principal curve model with an improved differential evolution machine learning model for prostate segmentation. Both qualitative and quantitative experimental results demonstrated that our segmentation model outperformed other state-of-the-art methods.

Author statement

The authors declare that they have no known competing financial interests or personal relationships that may appear to have influenced the work reported in this paper.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partly supported by Innovation and Technology Fund Projects, Hong Kong, No. ITS/080/19.

Appendix

A. Notation

B. Preliminary work

B.1 Identifying the principal curve

Fig. 11 shows the entire process used to identify the principal curve. In step (a), the process begins with a first principal component line consisting of v_1 and v_2 . In steps (b) to (c), the principal curve is updated because of the updated vertices when a new vertex v_2 is added. From step (d), we obtain the principal curve f consisting of line segments. A partial magnification of the principal curve is also shown. During the process of identifying the principal curve, the projection index $t(p)$ is used to denote the nearest distance at which the principal curve is closest to the data points p [20].

B.2 CKPC method

As the KPC method [20] uses a line segment as the starting curve and does not take into consideration shape constraint conditions, it cannot correctly describe the data sequence of a closed data set. To address this issue, we previously developed a CKPC method [14,30,23], as illustrated in the flowchart in **Fig. 12**. The CKPC method has several improvements compared to the KPC method, including initialization, normalization, constraints, and loop conditions.

B.2.1 KPC method

(1) *Projection step:* f is a polygon curve consisting of the vertex subset $V_i = \{v_1, v_2, \dots, v_{iv}\}$ and the segment subset $S_i = \{s_1, s_2, \dots, s_{is}\}$, where iv and is denote the total numbers of vertices and segments, respectively. We set the vertex v_i on f to be the nearest neighbor of data point p_i in P_n and assign p_i to the vertex subset V_i . If segment s_i on f is the near-

Table 7
A description of the notation used in Appendix A.

	Description	Symbols
RCKPC	Principal curve	f
	Data point set/data point/center point of the data set	$P_n/p/p'$
	Vertex subset/segment subset in principal curve	$V_i = \{v_1, v_2, \dots, v_{iv}\}/S_i = \{s_1, s_2, \dots, s_{is}\}$
	Vertex/segment of principal curve	v/s
	Total number of vertices/segments of principal curve	iv/is
	Projection index	t
	Penalty factor	λ
	Length of segment	l
	Data radius	r
	Maximum distance deviation	Δs
	Angle at intersection vertex of two segments of principal curve	γ_i
	Penalized / new penalized distance function	$G_n(v_i)/G_n^+(v_i)$
	Penalty / new penalty at the vertex	$P(v_i)/D(v_i)$
	Average squared distance function	$\Delta_n(v_i)$
MCSDE	Flag of vertex cleaning	$flag(v_i)$
	Number of populations	Np
	Objective function	$g(\bullet)$
	Individual/new generated individual	$pop/npop$
	Mutant vector / trial vector	vec/u
	Present/maximum iteration number	$G/Gmax$
	Mutation factor/mean mutation factor	F/uF
	Crossover rate/mean crossover rate	CR/uCR
	Predefined lower / upper bounds of the search space	U_{min}/U_{max}
	Set of all successful mutation factors / crossover probabilities	S_F/S_{CR}
ABPNN	Adjusted parameter	ap
	Probability of using the mutation operator	pro_G
	Number of input neurons	n
	Number of hidden neurons	q
	Number of output neurons	u
	Weight from the input layer to the hidden layer	$w1$
	Weight from the hidden layer to the output layer	$w2$
	Threshold of the i th hidden neuron	$a1$
	Threshold of the u -th output neuron	$a2$
	Output of output units	$c(\bullet)$

RCKPC, redefined closed K-segment principal curve; MCSDE, memory-based, cuckoo search-based differential evolution; ABPNN, adaptive learning-rate backpropagation neural network.

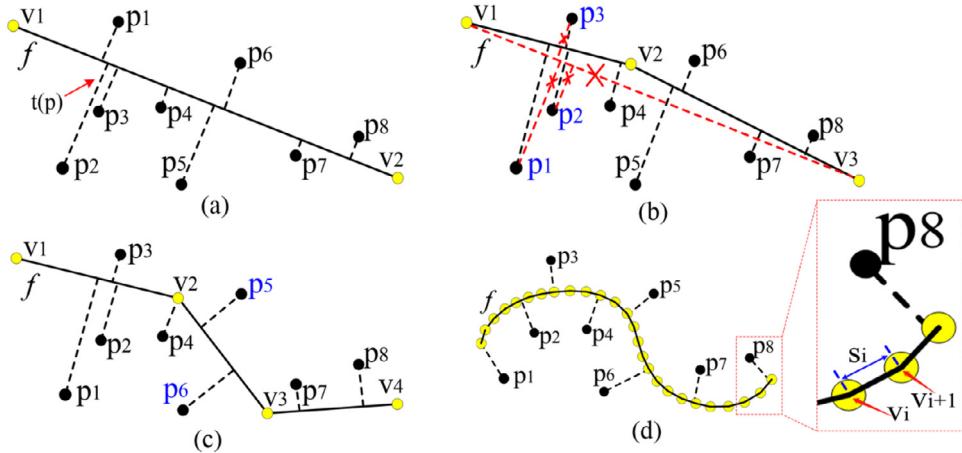


Fig. 11. Illustration of the process of identifying the principal curve. v denotes the vertex of the principal curve. The red dashed line indicates the deleted segment due to the updated principal curve after adding the new vertex, v . The blue font denotes the data points that changed the projected position of the principal curve.

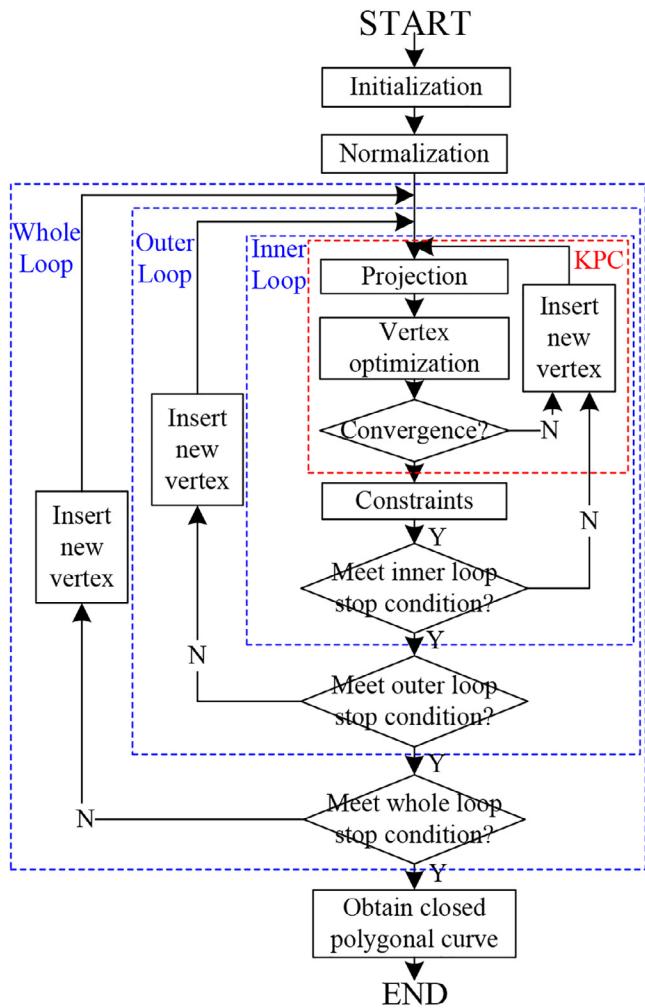


Fig. 12. The flowchart of the closed K-segment principal curve method.

est neighbor of data point p_i in P_n , we assign p_i to the segment subset S_i .

(2) *Vertex optimization step:* Using this step, the positions of all vertices v_i and segments s_i are updated according to the rule of the nearest distance from the data points p_i to the principal curve f [20]. Let $\pi(v_i) = r^2(1 + \cos \gamma_i)$, $\mu_+(v_i) = \|v_i - v_{i+1}\|^2$, and $\mu_-(v_i) = \|v_i - v_{i-1}\|^2$, where r is the data radius

and meets $r = \max_{p \in P_n} \|p - \frac{1}{n} \sum_{p' \in P_n} p'\|$ [29]. The data radius r denotes the maximum distance from the data point p to the center point p' of the data set P_n . If there is no center point p' in P_n , the nearest point in P_n is used as p' . Kegl and Krzyzak [29] used the $\pi(v_i)$ function to calculate the penalty $P(v_i)$ at vertex v_i , as shown in Eq. (19). To make the KPC method invariant to scaling, Kegl and Krzyzak [29] multiplied the cosines by the square of the data radius r . Furthermore, γ_i denotes the angle at the intersection vertex v_i of two segments, and the penalty $P(v_i)$ on vertex v_i is as follows:

$$P(v_i) = \begin{cases} \mu_+(v_i) + \pi(V_{i+1}) & \text{if } i = 1 \\ \mu_-(v_i) + \pi(V_i)\pi(V_{i+1}) & \text{if } i = 2 \\ \pi(V_{i-1}) + \pi(V_i)\pi(V_{i+1}) & \text{if } 2 < i < iv \\ \pi(V_{i-1}) + \pi(V_i) + \mu_+(v_i) & \text{if } i = iv \\ \pi(V_{i-1}) + \mu_-(v_i) & \text{if } i = iv + 1 \end{cases} \quad (19)$$

We set $\tau(v_i) = \sum_{x \in V_i} \Delta(x, v_i)$, $\sigma_+(v_i) = \sum_{x \in S_i} \Delta(x, s_i)$, and $\sigma_-(v_i) = \sum_{x \in S_{i-1}} \Delta(x, s_{i-1})$, and obtain the average squared distance function $\Delta_n(v_i)$ as:

$$\Delta_n(v_i) = \begin{cases} v(v_i) & \text{if } i = 1 \\ \sigma_-(v_i) + v(v_i) + \sigma_+(v_i) & \text{if } 1 < i < iv + 1 \\ \sigma_-(v_i) + v(v_i) & \text{if } i = iv + 1 \end{cases} \quad (20)$$

The penalized distance function $G_n(v_i)$ is expressed as

$$G_n(v_i) = \frac{1}{n} \Delta_n(v_i) + \lambda \frac{1}{k+1} P(v_i) \quad (21)$$

where the penalty factor is $\lambda > 0$ and $\lambda = \lambda' \frac{is}{n^{1/3}} \frac{\sqrt{\Delta_n(f_{in,n})}}{r}$. λ' is the parameter of the method and takes a constant value of 0.13 [20].

(3) *Insert a new vertex:* We start with the optimized principal curve f and select the segment s_i with the largest number of data points p_i projecting to it. When there exist more than one such segments, we choose the longest segment s_i , and use the midpoint of this segment s_i as the new vertex.

B.2.2 Improvements

(1) *Initialization and normalization:* Compared with the KPC method, which uses a line segment as the starting curve, our previously developed CKPC method uses a closed square as the starting curve.

(2) *Stop conditions:* The whole, outer, and inner loop conditions are included in this improvement.

Following the method described by Kegl et al [20], the whole loop must terminate when the number of segments is exceeds:

$$ns(n, \Delta_n(f_{in,n})) = \beta n^{1/3} \Delta_n(f_{in,n})^{-1/2} r \quad (22)$$

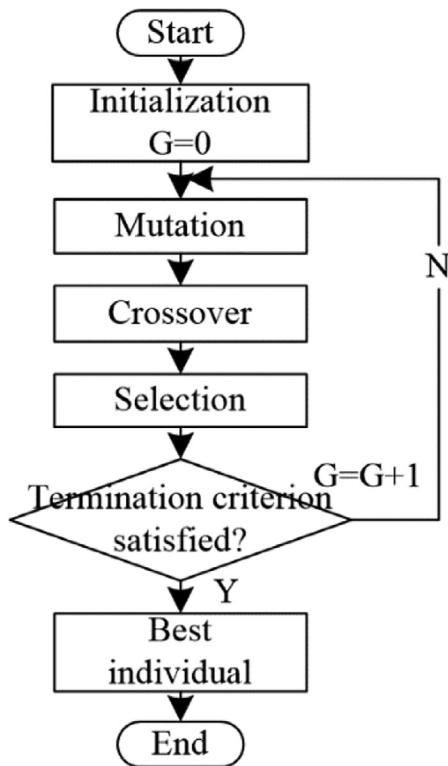


Fig. 13. The flowchart of the differential evolution method.

Based on previously published data [66], the penalty factor's weight β is set at 0.3, n is the number of data points p , and r is the data radius. Moreover, both the inner and outer loops satisfy the requirement that the distance deviation between the current distance and the last loop distance are smaller than the maximum distance deviation $\Delta s = 0.002$, as determined by repeated trials.

3) *Constraints:* When adding a new vertex, it follows that the line segment including the largest number of projected data points should be retained. If more than two segments satisfy the above rule, we choose the longest segment. In addition, the generated principal curve should remain closed, and the angle, α , between segments of the generated principal curve is an obtuse angle within the range of $[90^\circ, 180^\circ]$.

B.3 DE method

As the parameters (i.e., weight and threshold) of NNs are randomly initialized, the training results are always easily constrained by the local minimum [26]. To address this issue, some studies have used DE or DE-based methods to identify the optimal parameters of the NN, because the DE method is good at identifying the optimum global performance of the NN [27]. Fig. 13 shows the flowchart of the DE method.

First, we initialize the main parameters of the DE method, including the population size, N_p ; the mutation factor, F ; the crossover rate factor, CR ; the present iteration number, G ; and the maximum iteration number, $Gmax$. The individual pop_i is randomly initialized as: $pop_i = U_{min} + rand[0, 1] \times (U_{max} - U_{min})$, where $i = 1, 2, \dots, N_p$; $rand [0, 1]$ is a random number in the range of $[0, 1]$; and U_{min} and U_{max} are predefined lower and upper bounds of the search space, respectively.

We then obtain a mutant vector vec_i^{G+1} using a mutation step according to $vec_i^{G+1} = pop_{i1}^G + F \times (pop_{i2}^G - pop_{i3}^G)$, where i_1 , i_2 , and i_3 denote three random integers in the range of $[1, N_p]$. Furthermore, F is used to control the mutation degree and is set at 0.001, based on previously published data [67].

The trial vector u_i^{G+1} is then obtained using the crossover method, as follows,

$$u_i^{G+1} = \begin{cases} vec_i^{G+1}, & if rand[0, 1] \leq CR \\ pop_i^G, & otherwise \end{cases} \quad (23)$$

where CR affects the performance of the global search and is determined by the user within the range of $[0, 1]$. We set CR to 0.5 based on previously published data [67].

We then use a selection method to determine whether the trial vector, u_i^{G+1} , or the individual pop_i^G is a better individual for the next new generation, using the equation:

$$pop_i^{G+1} = \begin{cases} u_i^{G+1}, & if g(u_i^{G+1}) < g(pop_i^G) \\ pop_i^G, & otherwise \end{cases} \quad (24)$$

Where the objective function $g(x) = \sum_{i=1}^{N_p} pop_i^2$ is based on previously published data [38].

Furthermore, it is required to meet the termination criterion, $G \geq Gmax$.

B.4 Mathematical model

Our previous study [23] used a BPNN for training, as the main purpose of the training was to identify an appropriate mathematical map function (denoted by the parameters of the BPNN) to generate smooth lung contours, represented by the output of the BPNN (i.e., optimized vertices), that match the GT contours. The mathematical map function was as follows:

$$f(t) = (x(t), y(t)) = \left(\frac{1}{1 + e^{\sum_{i=1}^q (tw_{i1} - a_{i1})}}, \frac{1}{1 + e^{\sum_{i=1}^q (tw_{i2} - a_{i2})}} \right) \quad (25)$$

where the activation functions from both the input to the hidden layer and from the hidden layer to the output layer are sigmoid functions, and t is the projection index. w_{1i} ($i = 1, 2, \dots, q$) and $w_{2i,u}$ ($i = 1, 2, \dots, q$; $u = 1, 2$) are the weights from the input layer to the i th hidden neuron and from the i th hidden neuron to the u -th output neuron, respectively. a_{1i} ($i = 1, 2, \dots, q$) is the thresholds of the i th hidden neuron and $a_{2i,u}$ ($i = 1, 2, \dots, q$; $u = 1, 2$) is the threshold of the u -th output neuron.

C. Best model choice

We selected the optimal parameters (i.e., hidden neurons and epochs) for the H-ProSeg model, using the three metrics, DSC, Ω , and ACC, for evaluation. Fig. 14 and Fig. 15 show the training and validation results obtained using the H-ProSeg framework on hidden neurons and epochs, respectively. First, we set the number of epochs at 1000 and evaluated the effect of hidden neurons (Fig. 14). We then selected the optimal number of epochs using the determined optimal number of hidden neurons (Fig. 15). Overall, regardless of whether we performed the evaluation based on different numbers of hidden neurons or epochs, the trends of the three metrics were similar. Meanwhile, the trend in training results was similar to that of the validation results.

C.1 Evaluation on different neurons

The training and validation results obtained using the ProSeg method to evaluate the effects of different hidden neurons are shown in Fig. 14, with the number of epochs set at 1000. As seen in Fig. 14, the training results showed a similar trend to the validation results. The entire stage consisted of the rising stage and the saturated stage. At the rising stage, both the training and validation metrics were as low as 70% when the number of neurons was 1. One possible reason for this finding is that when the ABPNN has an insufficient number of hidden neurons, it fails to deal with complex issues. The model showed the best performance when the number of neurons was increased to 10, at which point the validation results gave DSC, Ω , and ACC values of 95.6%, 95.1%, and 94%,

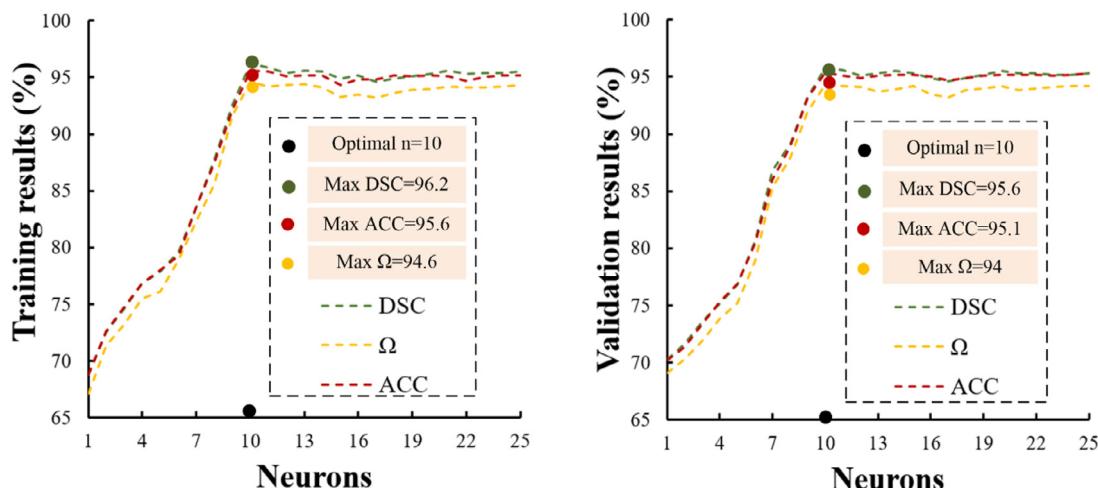


Fig. 14. Corresponding training and validation results for different numbers of hidden neurons. Green, yellow, and red curves indicate the Dice similarity coefficient (DSC), Jaccard similarity coefficient (Ω), and accuracy (ACC), respectively, according to the number of neurons. Meanwhile, the green, yellow, and red dots indicate the maximum DSC, Ω , and ACC values at the optimal number of neurons (10), respectively.

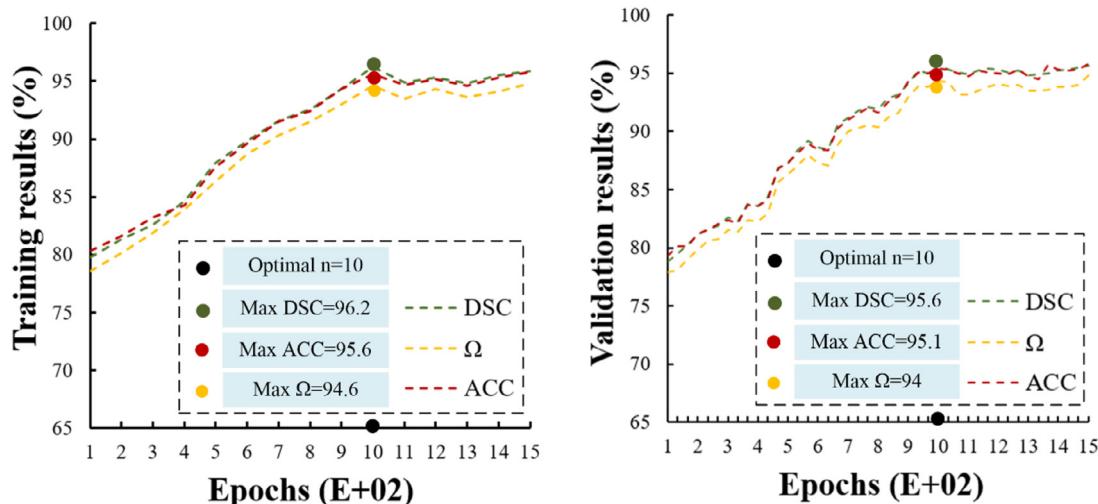


Fig. 15. Corresponding training and validation results for different numbers of epochs. DSC, Dice similarity coefficient; Ω , Jaccard similarity coefficient; ACC, and accuracy.

respectively. At 10 neurons, a saturated stage was reached, such that when the number of neurons further increased, the performance of the method began to decrease. The main cause of this phenomenon is that too many hidden neurons cause the model to become more complex, resulting in overfitting.

C.2 Evaluation on different epochs

In subsequent experiments, we used 10 neurons after combining them with the results achieved in the previous section. Fig. 15 shows the training and validation results for different numbers of epochs. When the number of epochs increased, the DSC, Ω , and ACC values increased. When the number of epochs was 1000, the model achieved the best performance. As the number of epochs further increased above 1000, the model performance became stable. Therefore, we set the number of epochs at 1000 and the number of neurons at 10.

References

- [1] J. Chen, Z. Wan, J. Zhang, W. Li, Y. Chen, Y. Li, Y. Duan, Medical image segmentation and reconstruction of prostate tumor based on 3D AlexNet, *Comput. Methods Progr. Biomed.* 200 (2021) 105878.
- [2] H. Bi, Y. Jiang, H. Tang, G. Yang, H. Shu, J.L. Dillenseger, Fast and accurate segmentation method of active shape model with Rayleigh mixture model clustering for prostate ultrasound images, *Comput. Methods Progr. Biomed.* 184 (2020) 105097.
- [3] H. Ma, Y. Zou, P.X. Liu, MHSU-Net: a more versatile neural network for medical image segmentation, *Comput. Methods Progr. Biomed.* 208 (2021) 106230.
- [4] X. Tang, J. Peng, B. Zhong, J. Li, Z. Yan, Introducing frequency representation into convolution neural networks for medical image segmentation via twin-Kernel Fourier convolution, *Comput. Methods Progr. Biomed.* 205 (2021) 106110.
- [5] D. Singh, V. Kumar, C.J. Das, A. Singh, A. Mehndiratta, Segmentation of prostate zones using probabilistic atlas-based method with diffusion-weighted MR images, *Comput. Methods Progr. Biomed.* 196 (2020) 105572.
- [6] K. Yan, X. Wang, J. Kim, M. Khadra, M. Fulham, D. Feng, A propagation-DNN: deep combination learning of multi-level features for MR prostate segmentation, *Comput. Methods Progr. Biomed.* 170 (2019) 11–21.
- [7] J. Funke, F. Tschopp, W. Grisaitis, A. Sheridan, C. Singh, S. Saalfeld, S.C. Turaga, Large Scale Image Segmentation with Structured Loss Based Deep Learning for Connectome Reconstruction, *IEEE Trans. Pattern Anal. Mach. Intell.* 41 (2019) 1669–1680.
- [8] X. Zhuang, Multivariate mixture model for myocardium segmentation combining multi-source images, *IEEE Trans. Pattern Anal. Mach. Intell.* 41 (2019) 2933–2946.
- [9] P. Luo, L. Lin, X. Liu, Learning Compositional Shape Models of Multiple Distance Metrics by Information Projection, *IEEE Trans. Neural Netw. Learn. Syst.* 27 (2015) 1417–1428.
- [10] Y. Lei, S. Tian, X. He, T. Wang, B. Wang, P. Patel, A.B. Jani, H. Mao, W.J. Curran, T. Liu, X. Yang, Ultrasound prostate segmentation based on multidirectional deeply supervised V-Net, *Med. Phys.* 46 (2019) 3194–3206.
- [11] H. Jia, Y. Xia, W. Cai, M. Fulham, D.D. Feng, Prostate segmentation in MR images using ensemble deep convolutional neural networks, in: Proceedings of the IEEE 14th International Symposium on Biomedical Imaging ISBI, Melbourne, Australia, 2017, pp. 762–765. IEEE.

- [12] G. Litjens, R. Toth, W. van de Ven, C. Hoeks, S. Kerkstra, B. van Ginneken, G. Vincent, G. Guillard, N. Birbeck, J. Zhang, R. Strand, F. Malmberg, Y. Ou, C. Davatzikos, M. Kirschner, F. Jung, J. Yuan, W. Qiu, Q. Gao, P. "Eddie" Edwards, B. Maan, F. van der Heijden, S. Ghose, J. Mitra, J. Dowling, D. Barratt, H. Huisman, A. Madabhushi, Evaluation of prostate segmentation algorithms for MRI: the PROMISE12 challenge, *Med. Image Anal.* 18 (2014) 359–373.
- [13] Y. Wang, H. Dou, X. Hu, L. Zhu, X. Yang, M. Xu, J. Qin, P.A. Heng, T. Wang, D. Ni, Deep attentive features for prostate segmentation in 3D transrectal ultrasound, *IEEE Trans. Med. Imaging* 38 (2019) 2768–2778.
- [14] T. Peng, Y. Wang, T.C. Xu, X. Chen, Segmentation of lung in chest radiographs using hull and closed polygonal line method, *IEEE Access* 7 (2019) 137794–137810.
- [15] F. Balsiger, Y. Soom, O. Scheidegger, M. Reyes, Learning shape representation on sparse point clouds for volumetric image segmentation, in: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Shenzhen, China, 2019, pp. 273–281.
- [16] X. Chen, B.M. Williams, S.R. Vallabhaneni, G. Czanner, R. Williams, Y. Zheng, Learning active contour models for medical image segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition CVPR, CA, USA, 2019, pp. 11624–11632. IEEE, Long Beach.
- [17] L. Tan, A. Liang, L. Li, W. Liu, H. Kang, C. Chen, Automatic prostate segmentation based on fusion between deep network and variational methods, *J. X Ray Sci. Technol.* 27 (2019) 821–837.
- [18] M. Bardis, R. Houshyar, C. Chantaduly, A. Ushinsky, J. Glavis-Bloom, M. Shaver, D. Chow, E. Uchio, P. Chang, Deep learning with limited data: organ segmentation performance by U-Net, *Electronics* 9 (2020) 1199.
- [19] Z. Tian, X. Li, Y. Zheng, Z. Chen, Z. Shi, L. Liu, B. Fei, Graph-convolutional-network-based interactive prostate segmentation in MR images, *Med. Phys.* 47 (2020) 4164–4176.
- [20] B. Kégl, T. Linder, K. Zeger, Learning and design of principal curves, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (2000) 281–297.
- [21] T. Tarpey, Self-consistency and principal component analysis, *J. Am. Stat. Assoc.* 94 (1999) 456–467.
- [22] T. Hastie, W. Stuetzel, Principal curves, *J. Am. Stat. Assoc.* 84 (1989) 502–516.
- [23] T. Peng, Y. Wang, T.C. Xu, L. Shi, J. Jiang, S. Zhu, Detection of lung contour with closed principal curve and machine learning, *J. Digit. Imaging* 31 (2018) 520–533.
- [24] G. Biau, A. Fischer, Parameter selection for principal curves, *IEEE Trans. Inf. Theory* 58 (2012) 1924–1939.
- [25] Y.H. Wang, Y. Guo, Y.C. Fu, Z.Y. Shen, An algorithm for learning principal curves with principal component analysis and back-propagation network, in: Proceedings of the 7th International Conference on Intelligent Systems Design and Applications ISDA, Rio de Janeiro, Brazil, 2007, pp. 447–453. IEEE2007.
- [26] Yi Shang, B.W. Wah, Global optimization for neural network training, *Computer* 29 (1996) 45–54 (Long Beach Calif.).
- [27] Y.R. Zeng, Y. Zeng, B. Choi, L. Wang, Multifactor-influenced energy consumption forecasting using enhanced back-propagation neural network, *Energy* 127 (2017) 381–396.
- [28] N. Orlando, D.J. Gillies, I. Gyacskov, C. Romagnoli, D. D'Souza, A. Fenster, Automatic prostate segmentation using deep learning on clinically diverse 3D transrectal ultrasound images, *Med. Phys.* 47 (2020) 2413–2426.
- [29] B. Kégl, A. Krzyzak, Piecewise linear skeletonization using principal curves, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (2002) 59–74.
- [30] T. Peng, T.C. Xu, Y. Wang, F. Li, Deep belief network and closed polygonal line for lung segmentation in chest radiographs, *Comput. J.* (2020).
- [31] G. Jing, W. Du, Y. Guo, Studies on prediction of separation percent in electro-dialysis process via BP neural networks and improved BP algorithms, *Desalination* 291 (2012) 78–93.
- [32] R. Chi, Y. Su, Z. Qu, X. Chi, A hybridization of Cuckoo search and differential evolution for the logistics distribution center location problem, *Math. Probl. Eng.* 2019 (2019) 1–16.
- [33] M.Z. Ali, N.H. Awad, P.N. Suganthan, R.G. Reynolds, An adaptive multipopulation differential evolution with dynamic population reduction, *IEEE Trans. Cybern.* 47 (2017) 2768–2779.
- [34] B.N. Jiang, On the least-squares method, *Comput. Methods Appl. Mech. Eng.* 152 (1998) 239–257.
- [35] C.A. Hall, W.W. Meyer, Optimal error bounds for cubic spline interpolation, *J. Approx. Theory* 16 (1976) 105–122.
- [36] H. Zhang, W. Pedrycz, D. Miao, C. Zhong, A global structure-based algorithm for detecting the principal graph from complex data, *Pattern Recognit.* 46 (2013) 1638–1647.
- [37] J. Zhang, D. Chen, U. Kruger, Adaptive constraint K-segment principal curves for intelligent transportation systems, *IEEE Trans. Intell. Transp. Syst.* 9 (2008) 666–677.
- [38] V. Feoktistov, S. Janaqi, Generalization of the strategies in differential evolution, in: Proceedings of the 18th International Parallel and Distributed Processing Symposium, Santa Fe, NM, USA, 2004, pp. 165–170. IEEE.
- [39] Y. Su, R. Chi, Multi-objective particle swarm-differential evolution algorithm, *Neural Comput. Appl.* 28 (2017) 407–418.
- [40] C. Cobos, H. Muñoz-Collazos, R. Urbano-Muñoz, M. Mendoza, E. León, E. Herrera-Viedma, Clustering of web search results based on the cuckoo search algorithm and balanced Bayesian information criterion, *Inf. Sci.* 281 (2014) 248–264.
- [41] U. Mlakar, B. Potočnik, J. Brest, A hybrid differential evolution for optimal multilevel image thresholding, *Expert Syst. Appl.* 65 (2016) 221–232.
- [42] Y. Sun, Q. Zhang, Optimization design and reality of the virtual cutting process for the boring bar based on PSO-BP neural networks, *Neural Comput. Appl.* 29 (2018) 1357–1367.
- [43] S. Amari, Backpropagation and stochastic gradient descent method, *Neurocomputing* 5 (1993) 185–196.
- [44] X. Li, L. Yu, H. Chen, C.W. Fu, L. Xing, P.A. Heng, Transformation consistent self-ensembling model for semi-supervised medical image segmentation, *IEEE Trans. Neural Netw. Learn. Syst.* 32 (2020) 523–534.
- [45] L. Zhang, D. Xu, Z. Xu, X. Wang, D. Yang, T. Sanford, S. Harmon, B. Turkbey, B. Wood, H. Roth, A. Myronenko, Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation, *IEEE Trans. Med. Imaging* 39 (2020) 2531–2540.
- [46] M.D. Abràmoff, P.T. Lavin, M. Birch, N. Shah, J.C. Folk, Pivotal trial of an autonomous AI-based diagnostic system for detection of diabetic retinopathy in primary care offices, *Npj Digit. Med.* 1 (2018) 39.
- [47] E.R. Arce-Santana, A.R. Mejía-Rodríguez, E. Martínez-Peña, A. Alba, M. Méndez, E. Scalco, A. Mastropietro, G. Rizzo, A new probabilistic active contour region-based method for multiclass medical image segmentation, *Med. Biol. Eng. Comput.* 57 (2019) 565–576.
- [48] S. Ali, A. Madabhushi, An Integrated region-, boundary-, shape-based active contour for multiple object overlap resolution in histological imagery, *IEEE Trans. Med. Imaging* 31 (2012) 1448–1460.
- [49] S. McKinley, M. Levine, Cubic spline interpolation, *Coll. Redw.* 45 (1998) 1049–1060.
- [50] O. Ronneberger, P. Fischer, T. Brox, N. Navab, J. Hornegger, W.M. Wells, A.F. Frangi, U-Net: convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention- MICCAI, Springer International Publishing, Cham, 2015, pp. 234–241. 2015.
- [51] K. He, G. Gkioxari, P. Dollar, R. Girshick, R.C.N.N. Mask, in: Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 2017, pp. 2961–2969.
- [52] Z. Zhou, M.M.R. Siddiquee, N. Tajbakhsh, J. Liang, UNet++: redesigning skip connections to exploit multiscale features in image segmentation, *IEEE Trans. Med. Imaging* 39 (2020) 1856–1867.
- [53] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, J. Dean, A guide to deep learning in healthcare, *Nat. Med.* 25 (2019) 24–29.
- [54] S.A. Taghanaki, Y. Zheng, S.Kevin Zhou, B. Georgescu, P. Sharma, D. Xu, D. Comaniciu, G. Hamarneh, Combo loss: handling input and output imbalance in multi-organ segmentation, *Comput. Med. Imaging Graph.* 75 (2019) 24–33.
- [55] X. Jia, L. Ren, J. Cai, Clinical implementation of AI technologies will require interpretable AI models, *Med. Phys.* 47 (2020) 1–4.
- [56] L. Xing, E.A. Krupinski, J. Cai, Artificial intelligence will soon change the landscape of medical physics research and practice, *Med. Phys.* 45 (2018) 1791–1793.
- [57] E. Ataer-Cansizoglu, E. Bas, J. Kalpathy-Cramer, G.C. Sharp, D. Erdogmus, Contour-based shape representation using principal curves, *Pattern Recognit.* 46 (2013) 1140–1150.
- [58] E.M.A. Anas, P. Mousavi, P. Abolmaesumi, A deep learning approach for real time prostate segmentation in freehand ultrasound guided biopsy, *Med. Image Anal.* 48 (2018) 107–116.
- [59] D.C. Stanford, A.E. Raftery, Finding curvilinear features in spatial point patterns: principal curve clustering with noise, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (2000) 601–609.
- [60] E. Gibson, W. Li, C. Sudre, L. Fidon, D.I. Shakir, G. Wang, Z. Eaton-Rosen, R. Gray, T. Doel, Y. Hu, T. Whynnie, P. Nachev, M. Modat, D.C. Barratt, S. Ourselin, M.J. Cardoso, T. Vercauteren, NiftyNet: a deep-learning platform for medical imaging, *Comput. Methods Progr. Biomed.* 158 (2018) 113–122.
- [61] E.D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q.V. Le, AutoAugment: learning augmentation strategies from data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 2019, pp. 113–123. IEEE.
- [62] A.J. Ratner, H.R. Ehrenberg, Z. Hussain, J. Dunnmon, C. Ré, Learning to compose domain-specific transformations for data augmentation, *Adv. Neural Inf. Process. Syst.* (2017) 3239–3249.
- [63] J. Shiraishi, S. Katsuragawa, J. Ikezoe, T. Matsumoto, T. Kobayashi, K. Komatsu, M. Matsui, H. Fujita, Y. Kodera, K. Doi, Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules, *Am. J. Roentgenol.* 174 (2000) 71–74.
- [64] S. Candemir, S. Jaeger, K. Palaniappan, J.P. Musco, R.K. Singh, Zhiyun Xue, A. Karargyris, S. Antani, G. Thoma, C.J. McDonald, Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration, *IEEE Trans. Med. Imaging* 33 (2014) 577–590.
- [65] F. Martínez-Martínez, M.J. Rupérez-Moreno, M. Martínez-Sober, J.A. Solves-Llorente, D. Lorente, A.J. Serrano-López, S. Martínez-Sanchis, C. Monserrat, J.D. Martín-Guerrero, A finite element-based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real-time, *Comput. Biol. Med.* 90 (2017) 116–124.
- [66] B. Kégl, T. Linder, K. Zeger, A polygonal line algorithm for constructing principal curves, in: Proceedings of the 11th International Conference on Neural Information Processing Systems NIPS, 1998, pp. 501–507.
- [67] N. Leema, H.K. Nehemiah, A. Kannan, Neural network classifier optimization using differential evolution with global information and back propagation algorithm for clinical datasets, *Appl. Soft Comput.* 49 (2016) 834–844.