Atlas registration and ensemble deep convolutional neural network-based prostate segmentation using magnetic resonance imaging

Haozhe Jia, Yong Xia, Yang Song, Weidong Cai, Micheal Fulham, David Dagan Feng

Paper link: https://www.sciencedirect.com/science/article/pii/S0925231217316132

Presented by: Tahereh Hassanzadeh

Abstract

- Propose a coarse-to-fine segmentation strategy.
- Segment endorectal coil prostate images and non-endorectal coil prostate images separately.
- present a registration-based coarse segmentation.
- Train deep neural networks as pixel-based classifier to predict whether the pixel in the potential boundary region is prostate pixel or not.
- A boundary refinement is used to eliminate the outlier and smooth the boundary.

Introduction

- 220,800 men were diagnosed with prostate cancer in the United States in 2015.
- Magnetic resonance (MR) imaging, due to its superior spatial resolution and tissue contrast, is the main imaging modality used to evaluate the prostate gland.
- The challenges mainly relate to the variability in size/shape/contours of the gland, heterogeneity in signal intensity around endorectal coils (ERCs), imaging artifacts and low contrast between the gland and adjacent structures.

Introduction

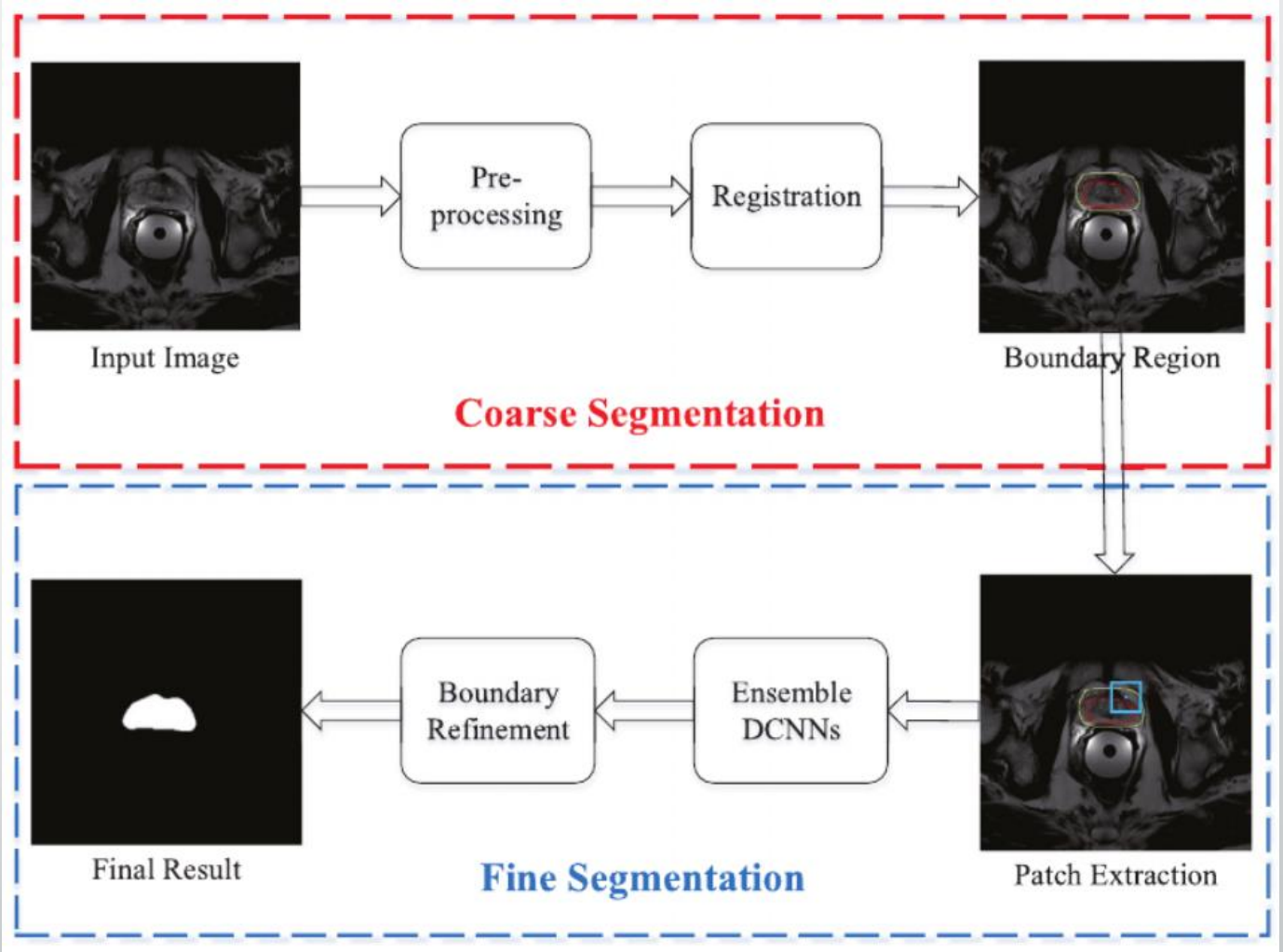
- Two contribution
  - First, we show that the use of pre-trained VGG-19 can alleviate overfitting and transfer the knowledge about image representation learned on the ImageNet dataset to characterizing prostate images.
  - Second, the experimental results demonstrate the use of ensemble learning can substantially improve the performance of prostate segmentation.

Introduction

- Dataset
  - Prostate MR Image Segmentation Challenge 2012 (PROMISE12).
  - https://promise12.grand-challenge.org/
  - SPIE-AAPM-NCI PROSTATEx Classification Challenge 2017 (PROSTATEx17) datasets.
  - https://wiki.cancerimagingarchive.net/display/Public/SPIE-AAPM-NCI+PROSTATEx+Challenges

Method

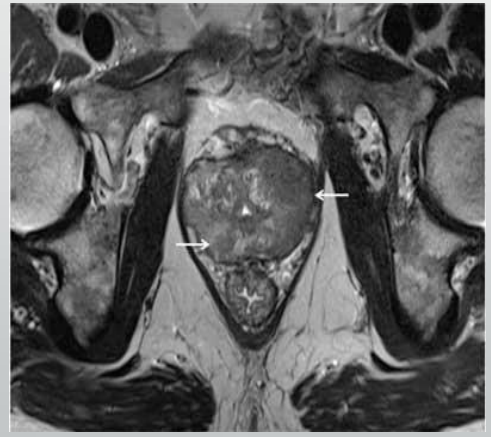
- Voxel value normalization
- Atlas- based coarse segmentation
- Ensemble DCNN-based fine segmentation
- Boundary refinement

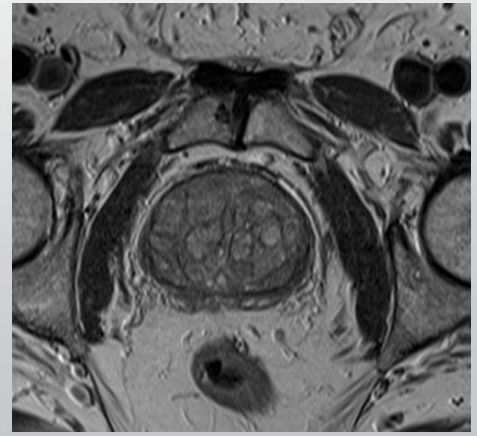


Voxel value normalization

- Uniform voxel size
  - 0.65 × 0.65 × 1.5 mm 3
  - The re-slicing procedure in the Statistical Parametric Mapping (SPM) software.
  - https://www.sciencedirect.com/topics/neuroscience/statistical-parametric-mapping
- Normalizing voxel values
  - non-ERCs
  - ERC

Voxel value normalization





Voxel value normalization

- non-ERC

Equation 1

$$I_x' = \begin{cases} 255 * (I_x - I_{\min}) / (I_{\max} - I_{\min}) & , I_x \leqslant \tau \\ 255 & , I_x > \tau \end{cases}$$

  - τ is truncate threshold
  - τ set to 4096 if $I_{max} > 4096$ and 1024 otherwise.

Voxel value normalization

- ERC
  - Poisson image editing
  - https://dl.acm.org/citation.cfm?doid=1201775.882269

Voxel value normalization

- Poisson image editing
  - It is a seamless editing and cloning tool.
  - Cloning allows the user to remove and add objects seamlessly.
  - This approach is based on Poisson partial differential equation and Dirichlet boundary condition which specifies the Laplacian of the unknown function over domain of interest.

Voxel value normalization

- **Step 1:** The region near the ERC that contains spikes was extracted by a threshold.
- **Step 2:** The voxel value normalization problem was converted into seeking an adjusted image f: Ω→R
  - Ω is spike region
  - f: Ω→R adjusted image intensity
  - f = I on the boundary of Ω
  - R set of real number
  - R^2 is two dimensional real number vector space
  - $g(x) = (I - G_\sigma * I)(x)$ is the high pass filtered image
  - By minimization of equation 2 is the solution for Poisson equation

Equation 2

$$E(f) = min \int_\Omega |\nabla f - \nabla g|^2 dx,$$

Equation 3

$$\nabla^2 f = \nabla^2 g$$

Voxel value normalization

- **Step 3:** Voxel values in the spike region were replaced by the corresponding values on the adjusted image f.
- **Step 4:** The spike suppressed image is applied to equation 1 to further normalize the voxel values.

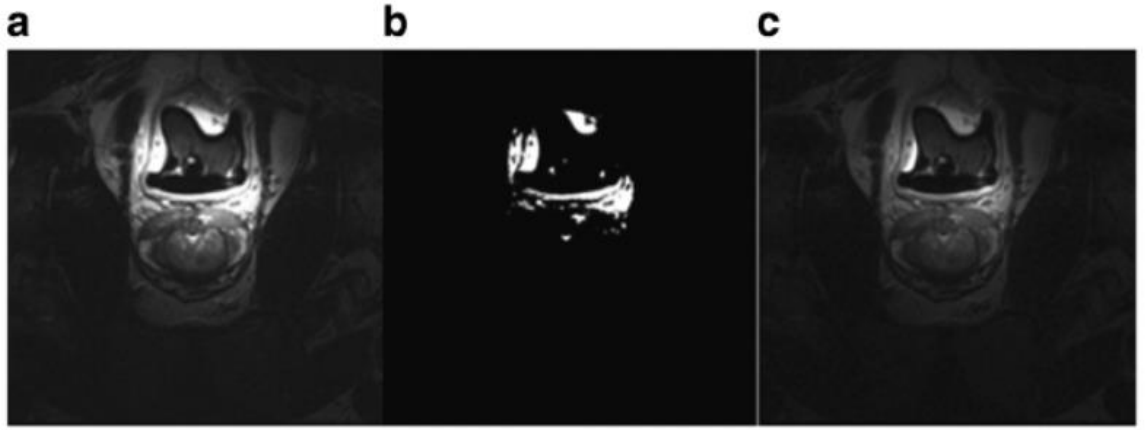


Fig. 2. MR images, using the ERC, showing: (a) the transaxial image, (b) detected spike region and (c) result from voxel value normalization.

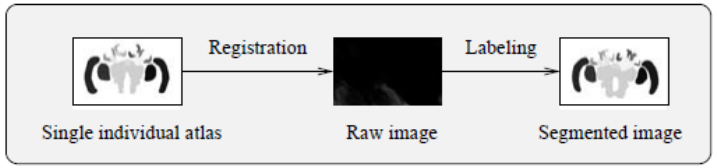
Atlas-based coarse segmentation

- The coarse segmentation of the gland was achieved via an atlas-based joint registration comparison analysis.
  - S: target image
  - I_i : training MR scan
  - L_i : corresponding ground truth
- The deformable registration via attribute matching and mutual- saliency weighting (DRAMMS) applied for registration to estimate a nonlinear transformation T that maps the training scan I_i to the target scan S.
- The estimated transformation T is applied to the ground truth L_i, and thus generates a prostate atlas A(S).
- Finally probabilistic atlas is constructed by averaging all atlases.

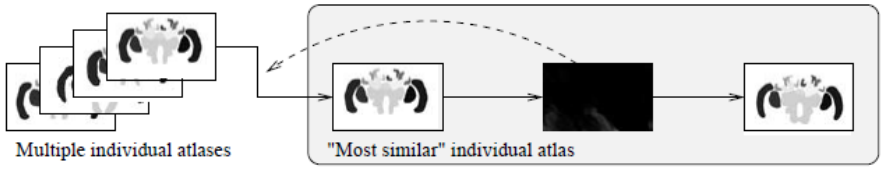
Equation 4

$$A^{(S)} = \frac{1}{2} \sum_{i=1}^{N} A_i^{(S)},$$

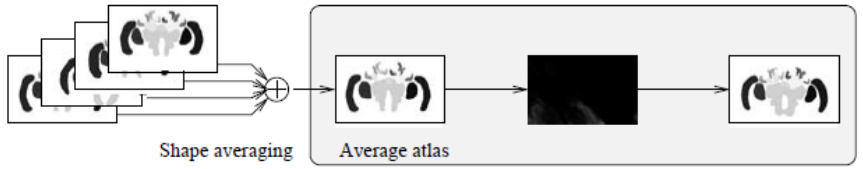
Atlas-based coarse segmentation



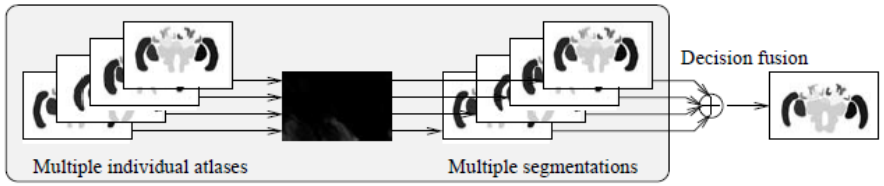
IND: Segmentation using a single individual atlas.



SIM: Segmentation using the “most similar” individual atlas.



AVG: Segmentation using an average shape atlas.



MUL: Independent segmentation using multiple individual atlases with decision fusion.

Atlas-based coarse segmentation

The target scan was partitioned into positive, boundary, and negative volumes by applying a low threshold 0.25 and a high threshold 0.75 to the probabilistic atlas.

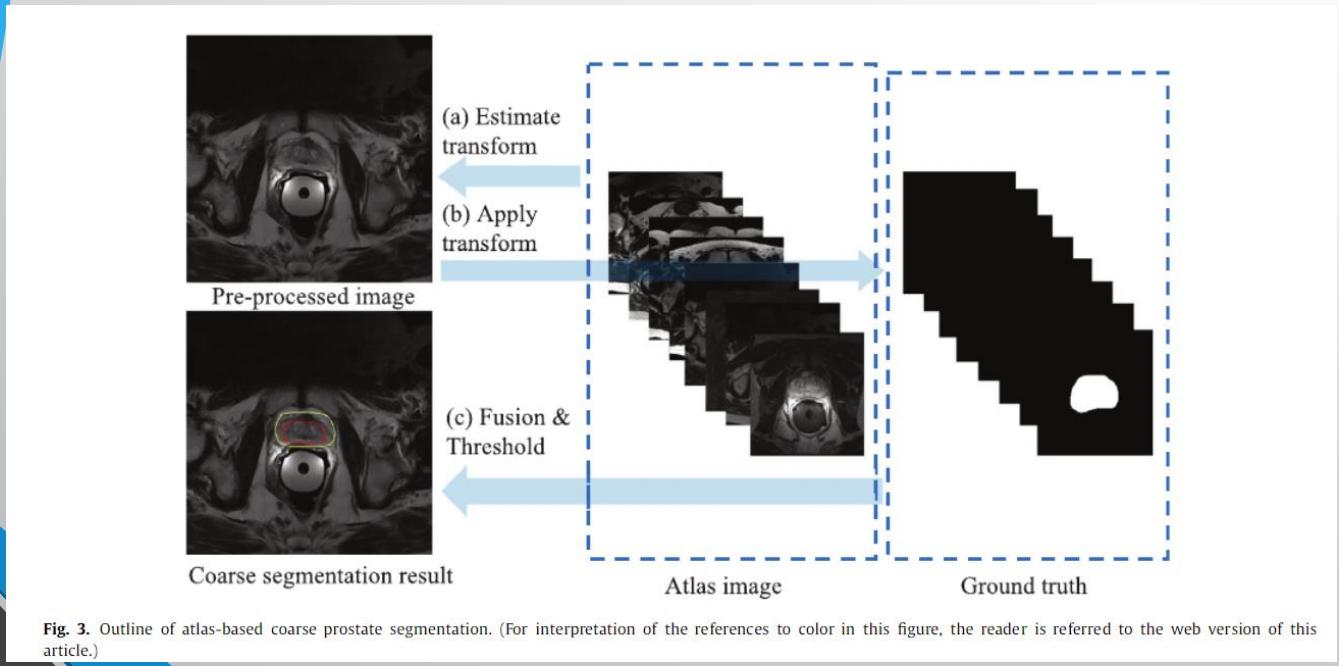
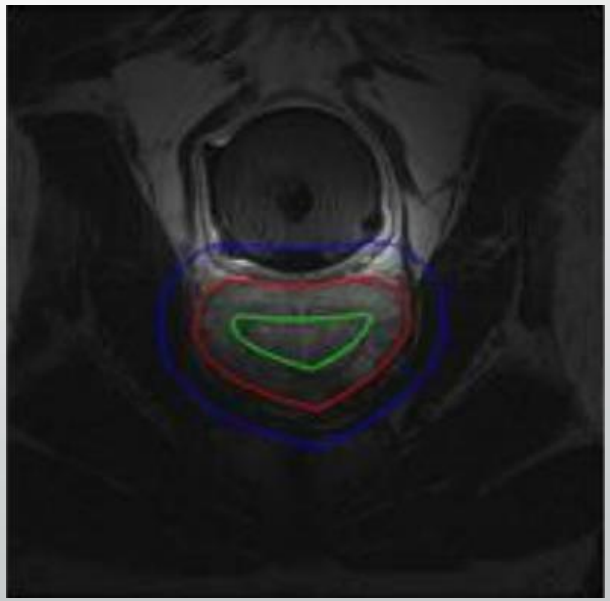


Fig. 3. Outline of atlas-based coarse prostate segmentation. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)



Ensemble DCNN-based fine segmentation

- The fine segmentation step further classifies each voxel in the boundary volume into prostate or non-prostate using the ensemble DCNN classifier.
- Fine segmentation is performed on a slice-by-slice basis from the axial view.

Ensemble DCNN-based fine segmentation

VGG-19

- 16 convolutional layers
  - 3*3 kernels
- 3 fully connected layers
  - 4096, 4096 and 1000 neurons
- 5 max pooling layers
  - 2*2 receptive fields
- ReLU
- Number of kernels from 64 to 512
- Dropout= 0.5 in fully connected layers
- Softmax- loss layer
- Previously trained by ImageNet
  - a 1000-category natural image database

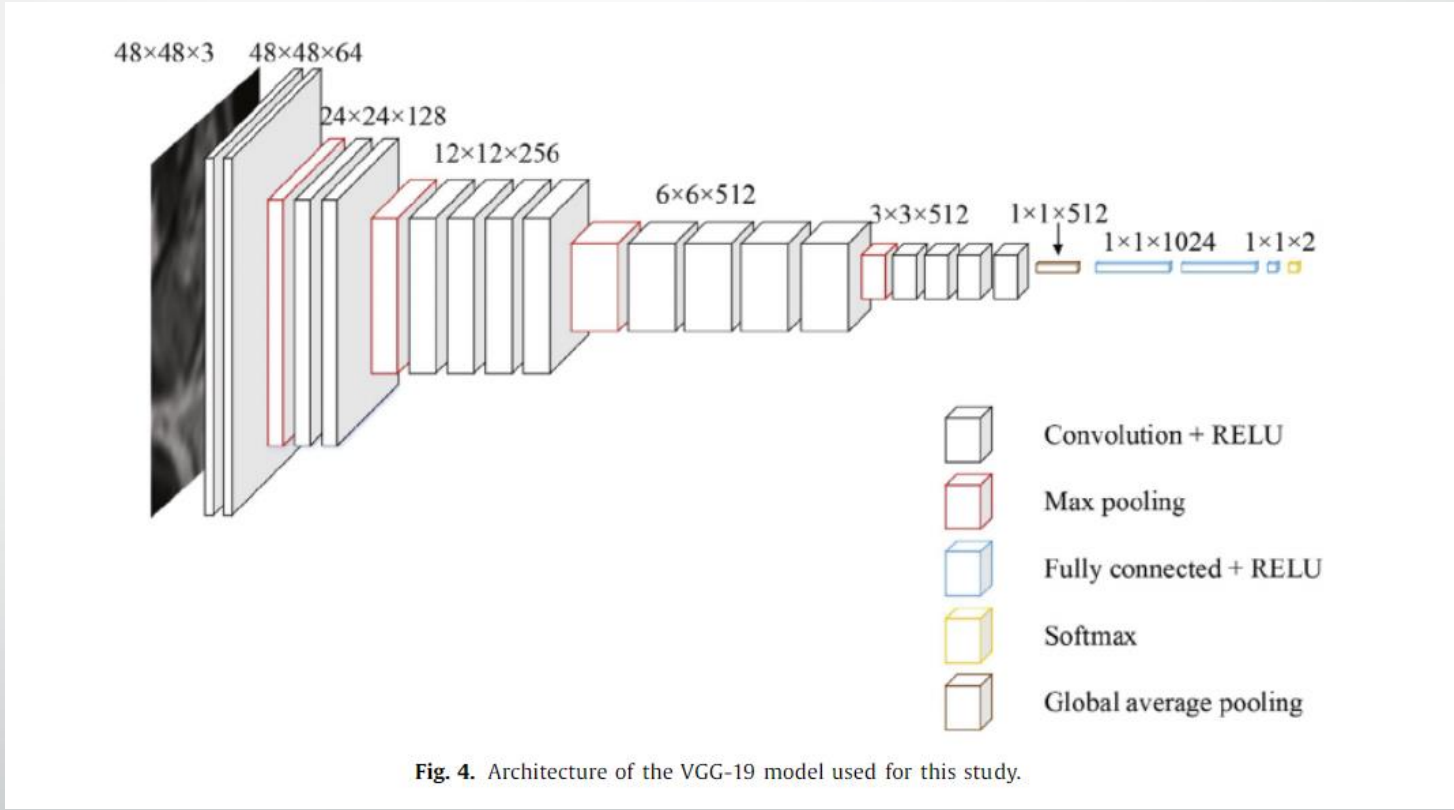
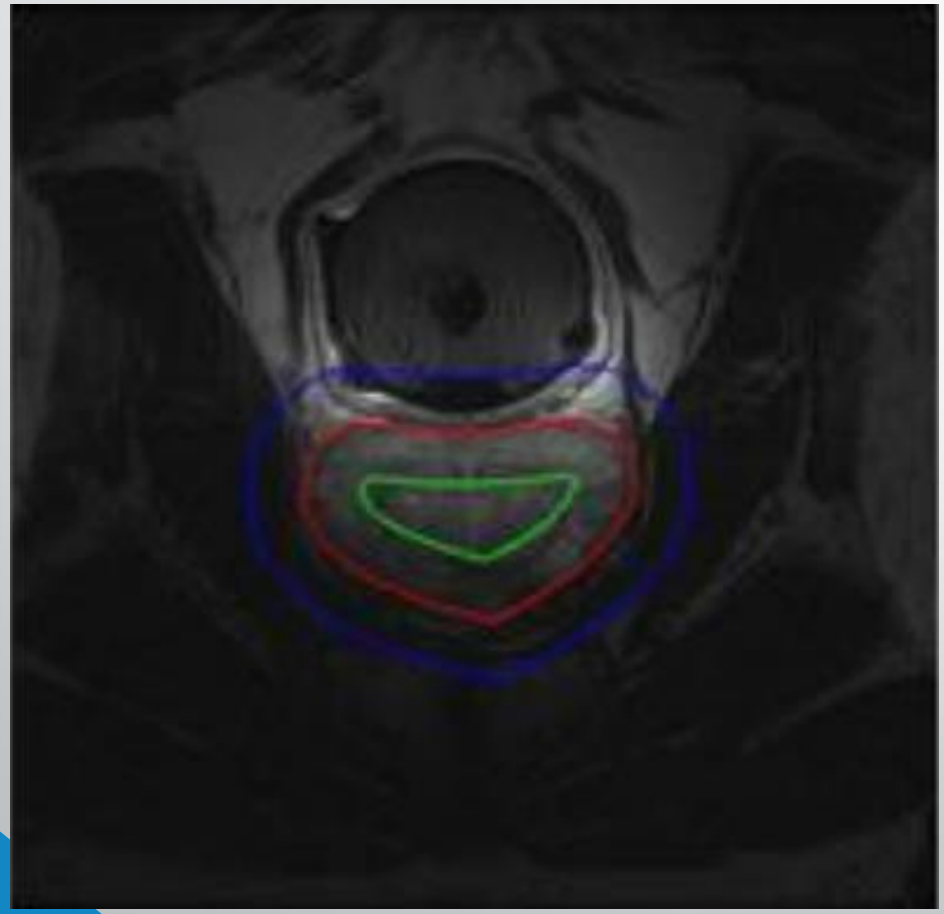


Fig. 4. Architecture of the VGG-19 model used for this study.

Ensemble DCNN-based fine segmentation

- Adapt VGG-19 for prostate segmentation
  - Randomly selected two neurons in the last fully connected layer and removed other output neurons and the weights attached to them.
  - Fine-tuned by using image patches extracted from the training studies.
  - A boundary region was defined as the difference between the dilation and erosion of the ground truth slice using a disk whose radius was 20 pixels.
  - Seed pixels were sampled with a 5 × 5 sliding window with a stride of 5.
  - Extracted 48 × 48 image patch cantered in seed pixel.

Ensemble DCNN-based fine segmentation



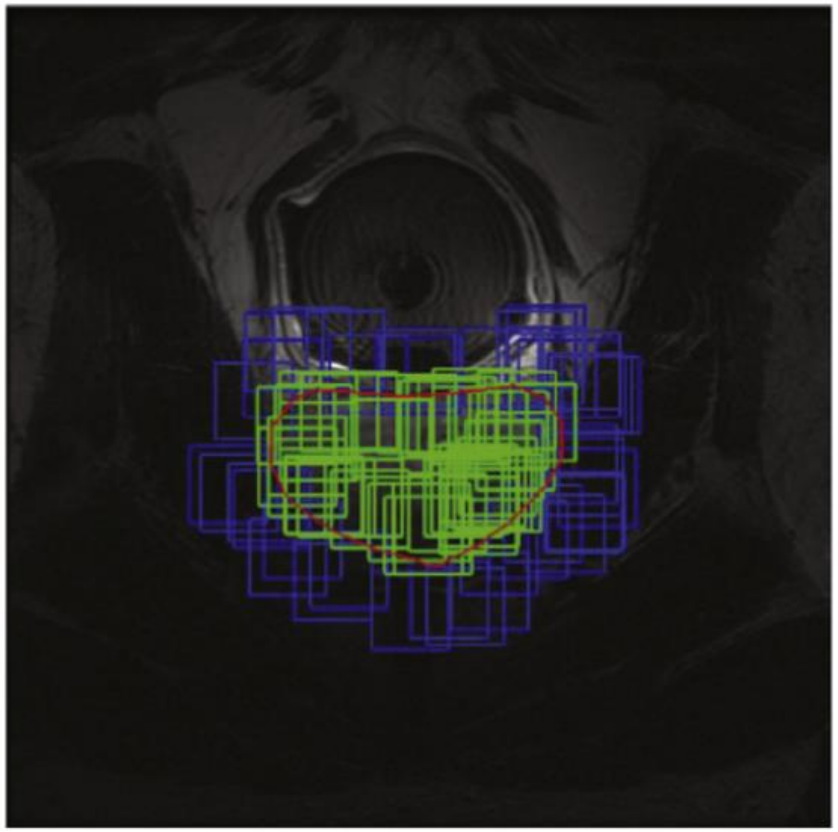


Fig. 5. Prostate patches (green) and non-prostate patches (blue) on a training slice. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Ensemble DCNN-based fine segmentation

Learning rate to 0.00001
Batch size to 100
7 individual VGG-19 models

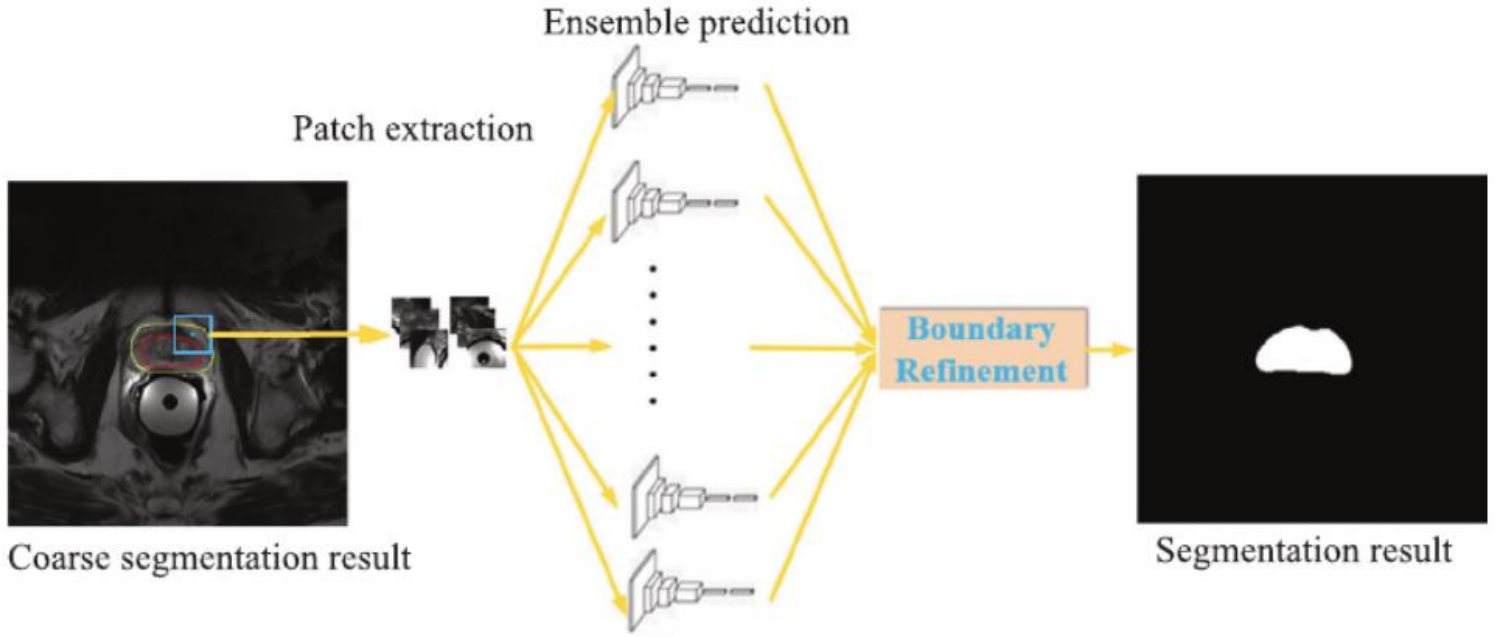


Fig. 6. Diagram for ensemble DCNN-based fine segmentation. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Boundary Refinement

- This process included 3 × 3 median filtering.
  - First calculated the distances between consecutive boundary points and the centroid.
  - Then removed 10% boundary points whose distance was most different from the mean distance.
  - Finally fitted a cubic B-spline to the remaining boundary points to obtain the refined segmentation.

Experiments and results

- **Data sets**
  - PROMISE12- 50 volumes for training and 30 volumes for testing.

Table 1
Details of the dataset acquisition protocols.

Center	Field strength	ERC	Resolution	Manufacturer
Hk	1.5	Y	0.625/3.6	Siemens
BIDMC	3	Y	0.25/2.2-3	GE
UCL	1.5/3	N	0.325–0.625/3-3.6	Siemens
RUNMC	3	N	0.5–0.75/3.6–4.0	Siemens

  - The PROSTATEx17 database has 204 training MR.
    - *T2- weighted*
    - *Ktrans*
    - *Apparent Diffusion coefficient Images*

Experiments and results

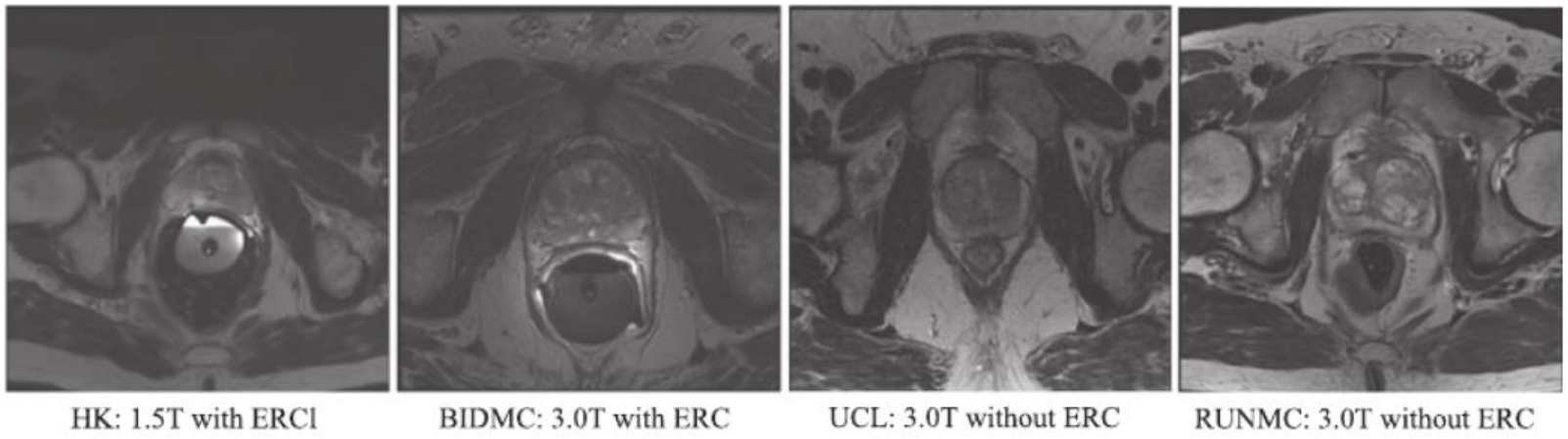


Fig. 7. Example image slices from PROMIS 12 dataset (left two scans with ERC; right: two scans without ERC).

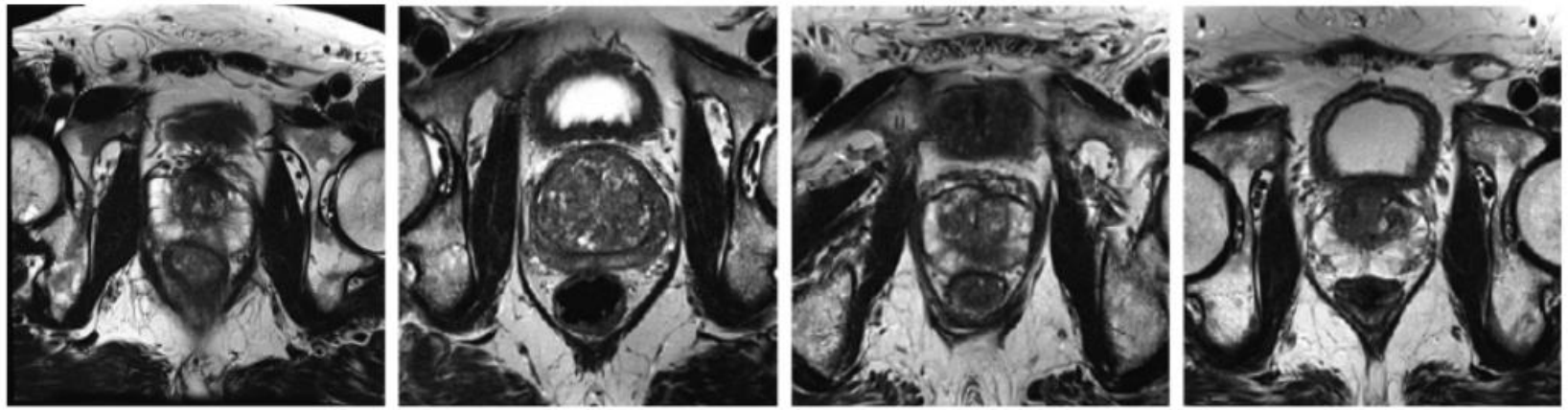


Fig. 8. Example image slices from the PROSTATEx dataset.

Experiments and results

- **Experiment setting and evaluations**
  - Four-fold cross-validation (each fold has ERC and non-ERC images)
- **Evaluation**
  - Dice Similarity Coefficient (DSC)
    - DSC ranges from 0 to 1
    - a higher value representing a more accurate segmentation result

$$D(X,Y) = \frac{2|X \cap Y|}{|X| + |Y|}$$

  - Relative Volume Difference (RVD)
    - A positive RVD reflects under-segmentation
    - A negative RVD reflects over-segmentation

$$RVD(X,Y) = 100 \times \left(\frac{|X|}{|Y|} - 1 \right)$$

Experiments and results

- **Evaluation**
  - Average Boundary Distance (ABD)

$$ABD(X_s + Y_s) = \frac{1}{|X_s| + |Y_s|}\left(\sum_{x \in X_s} \min_{y \in Y_s} \|x - y\| + \sum_{y \in Y_s} \min_{s \in X_s} \|x - y\|\right),$$

  - 95% Hausdorff Distance (95%HD)
  - Hausdorff Distance (HD)

$$HD_a sym(X_s, Y_s) = \max_{x \in X_s}(\min_{y \in Y_s} \|x - y\|)$$

- ABD and HD are classical shape distance-based evaluation metrics

Results

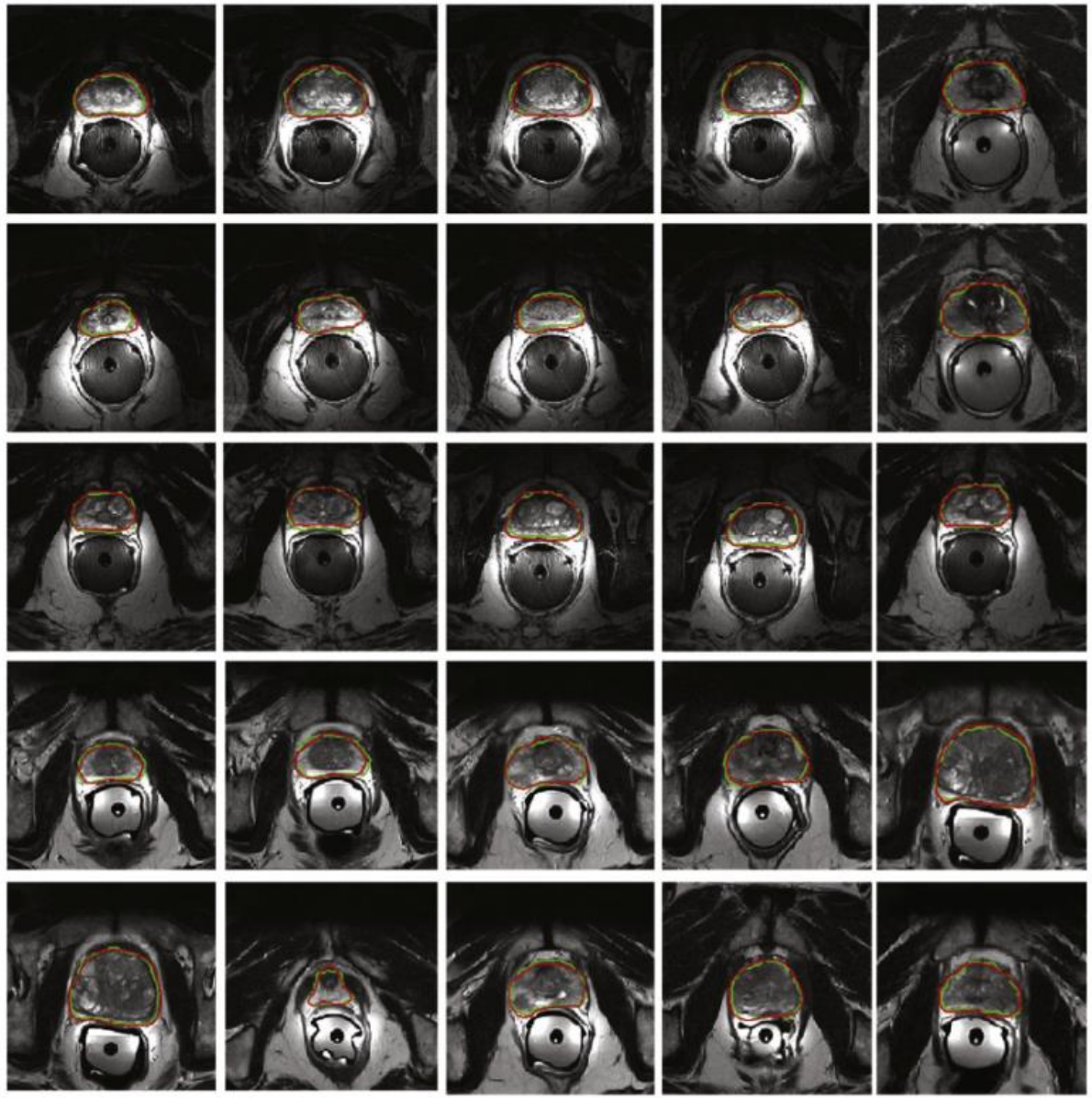


Fig. 9. ERC segmentation results with our algorithm; obtained boundaries are highlighted in green; the ground truth is outlined in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Results

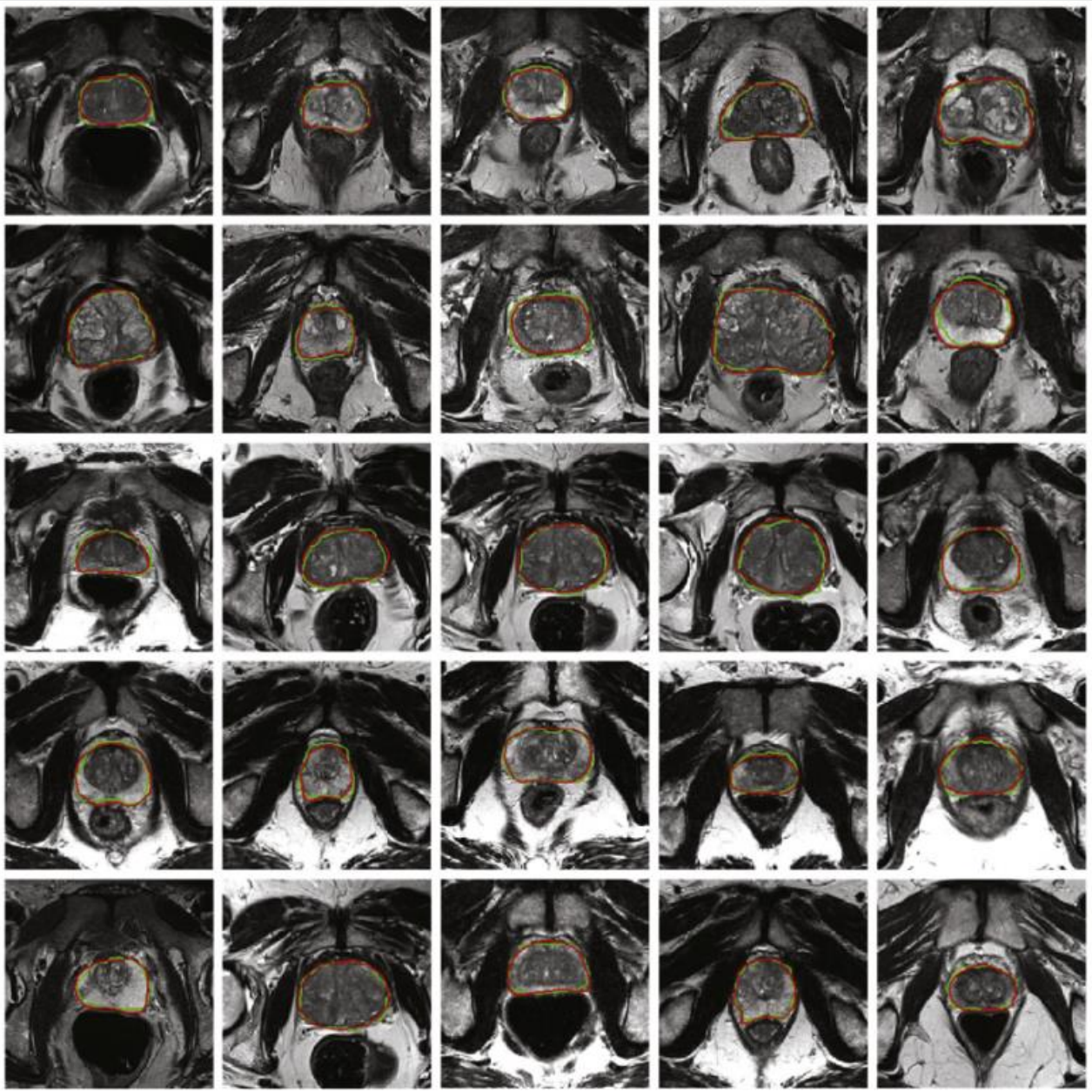


Fig. 10. Non-ERC segmentation results of our algorithm; obtained boundaries are highlighted in green; the ground truth is outlined in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Results

Table 2
Mean ± standard deviation of quantitative results for segmentations obtained by different algorithms.

Algorithm	DSC	RVD(%)	ABD(mm)	HD(mm)	95%HD(mm)
3D AAM (one shape model) [54]	0.784 ± 0.120	/	/	7.320 ± 4.910	/
3D AAM (two shape model) [54]	0.810 ± 0.120	/	/	6.430 ± 4.630	7.300 ± 4.900
Atlas fusion (local atlases and patch weighting) [55]	0.847 ± 0.044	/	/	/	/
Probabilistic ASM [56]	0.860 ± 0.006	/	1.600 ± 0.630	/	9.510 ± 2.730
Automated AAM [37]	0.880 ± 0.030	/	/	4.170 ± 1.350	/
Our algorithm	**0.910 ± 0.036**	**4.674 ± 9.401**	**1.583 ± 0.441**	**2.813 ± 1.292**	**4.579 ± 1.791**

Results

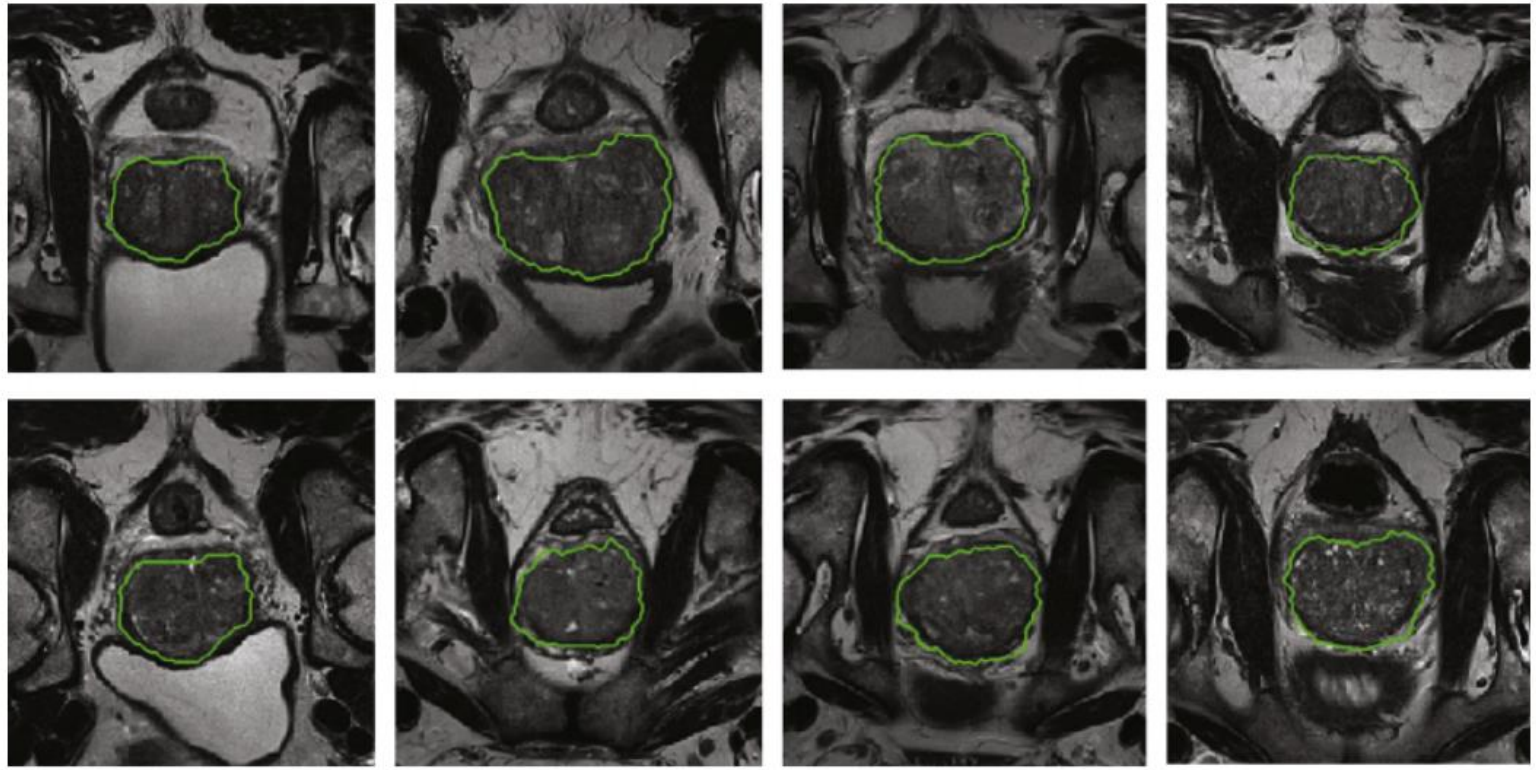


Fig. 11. Segmentation results using our approach on examples from the PROSTATEx dataset. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Results

Table 3
Comparison of segmentation performance with different patch sizes.

Patch size	DSC	RVD(%)	ABD(mm)	HD(mm)	95%HD(mm)
48	**0.887 ± 0.041**	6.367 ± 13.745	**1.715 ± 0.356**	**3.788 ± 1.153**	**5.736 ± 1.574**
56	0.882 ± 0.042	**5.927 ± 15.658**	1.883 ± 0.400	5.043 ± 1.260	7.579 ± 1.791
64	0.867 ± 0.054	9.174 ± 21.419	2.056 ± 0.520	6.352 ± 1.450	8.991 ± 1.626

Table 4
Comparison of segmentation performance with different ensemble numbers.

Ensemble number	DSC	RVD(%)	ABD(mm)	HD(mm)	95%HD(mm)	Total training time(h)
3	0.891 ± 0.040	6.300 ± 12.666	1.700 ± 0.354	3.715 ± 1.232	4.890 ± 1.733	**24**
5	0.903 ± 0.034	5.410 ± 12.700	1.667 ± 0.335	3.295 ± 1.122	4.612 ± 1.593	40
7	**0.910 ± 0.036**	**4.674 ± 9.407**	**1.583 ± 0.441**	**2.813 ± 1.292**	**4.579 ± 1.791**	56

Results

- Pre-trained versus fully-trained DCNN
  - Replaced the pre-trained VGG-19 model with the LeNet-5 model
  - fully-trained by using extracted image patches

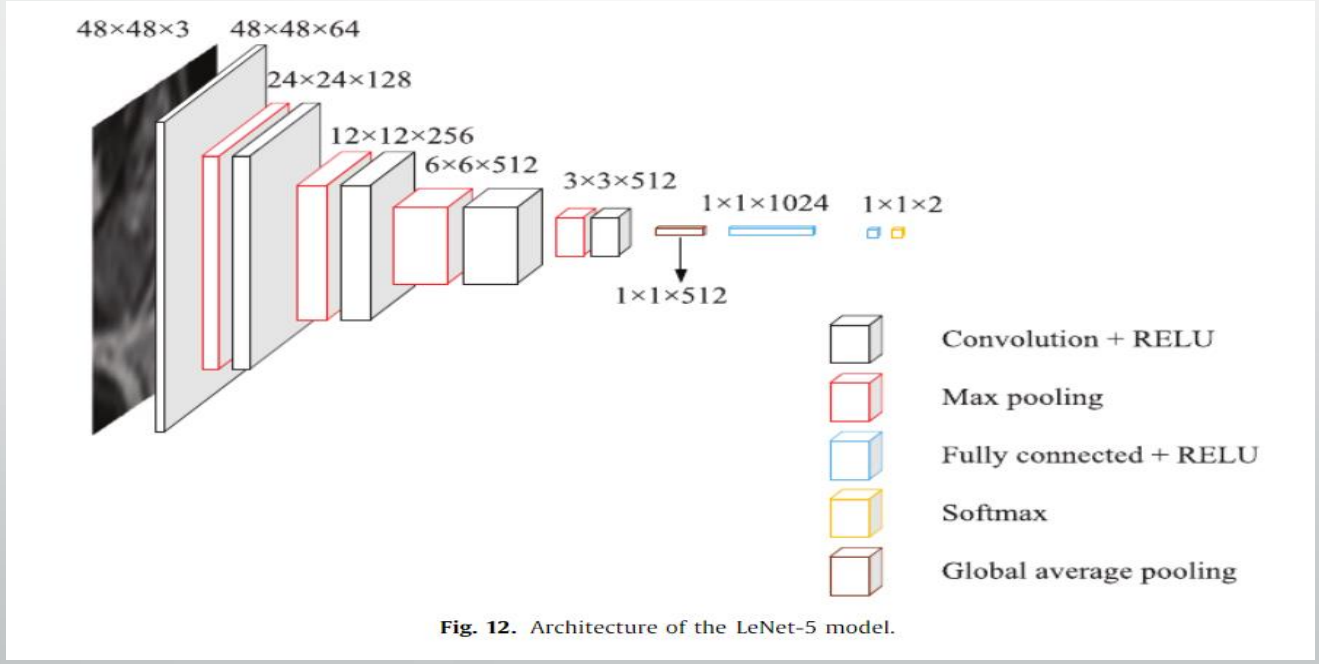


Fig. 12. Architecture of the LeNet-5 model.

Results

- Pre-trained versus fully-trained DCNN

Table 5
Mean ± standard deviation of quantitative results for segmentations obtained by different algorithms.

Algorithm		DSC	RVD(%)	ABD(mm)	HD(mm)	95%HD(mm)
Single model	LeNet-5	0.859 ± 0.065	10.079 ± 14.695	1.858 ± 0.626	5.320 ± 1.612	7.251 ± 2.756
	VGG-19	0.887 ± 0.041	6.367 ± 13.745	1.715 ± 0.356	3.788 ± 1.153	5.736 ± 1.574
Ensemble model	LeNet-5	0.877 ± 0.072	7.320 ± 10.268	1.753 ± 0.687	5.071 ± 1.698	7.091 ± 2.862
	VGG-19	**0.910 ± 0.036**	**4.674 ± 9.401**	**1.583 ± 0.441**	**2.813 ± 1.292**	**4.579 ± 1.791**

Results

- Computational Complexity

Table 6
Details and times for our approach.

Parameter		Value	
		Registration	Other work
Algorithm	Platform:	Mac OS X 10.10 system	Ubuntu 14.04 64 bit system
	Language:	C++	MATLAB 2016 b, Python
	Libraries:	DRAMMS	Keras, Tensorflow
	Multi-threaded:	None	None
	User interaction:	None	None
Machine	CPU:	Intel core i5	Intel core i7-4790
	CPU clock speed:	CPU 1.4 GHz	CPU 3.2 GHz
	Machine memory:	8 GB	32 GB
	GPU:	None	Nvidia GTX titan X
Time	Normalization and enhancement:	$\leq$ 1s (per study)	
	Registration:	30–40 min (per study registration by 10 atlas)	
	Label fusion:	$\approx$ 1 s (per study)	
	Off-line DCNNs training:	3.5 h (randomly initialization)	2 h (pre-trained weights)
	DCNNs prediction:	$\approx$ 2 min (per study)	
	Boundary refinement:	$\leq$ 1 s (per study)	
	Total segmentation:	$\approx$ 40 min (per study)	

Conclusion

- Present an automated coarse-to-fine segmentation.
- The coarse segmentation was achieved by using a probabilistic atlas.
- The fine segmentation was done using a cohort of trained DCNNs.
- Results suggest that ensemble DCNNs initialized with pre-trained weights substantially improve segmentation accuracy.

Thank You