COMPARISON OF OBJECTIVE FUNCTIONS IN CNN-BASED PROSTATE MAGNETIC RESONANCE IMAGE SEGMENTATION

Juhyeok Mun[†], *Won-Dong Jang*[†], *Deuk Jae Sung*[‡], *Chang-Su Kim*[†]

[†]School of Electrical Engineering, Korea University, Seoul, Korea
[‡]Department of Radiology, Anam Hospital, College of Medicine, Korea University, Seoul, Korea
E-mails: {jhmun, wdjang}@mcl.korea.ac.kr, {urorad, changsukim}@korea.ac.kr

ABSTRACT

We investigate the impacts of objective functions on the performance of deep-learning-based prostate magnetic resonance image segmentation. To this end, we first develop a baseline convolutional neural network (BCNN) for the prostate image segmentation, which consists of encoding, bridge, decoding, and classification modules. In the BCNN, we use 3D convolutional layers to consider volumetric information. Also, we adopt the residual feature forwarding and intermediate feature propagation techniques to make the BCNN reliably trainable for various objective functions. We compare six objective functions: Hamming distance, Euclidean distance, Jaccard index, dice coefficient, cosine similarity, and cross entropy. Experimental results on the PROMISE12 dataset demonstrate that the cosine similarity provides the best segmentation performance, whereas the cross entropy performs the worst.

Index Terms— Medical image segmentation, prostate segmentation, 3D convolutional neural networks, and objective functions

1. INTRODUCTION

In the prostate magnetic resonance (MR) image segmentation, prostate regions are segmented out from a 3D MR image, which consists of 2D image slices. The volume of a prostate can be estimated from the 3D segmentation result, which can be then used to assist the diagnosis of prostatism. A human expert can delineate a prostate in an MR image, but it demands much effort. It is hence necessary to develop an automatic algorithm that yields a precise segment of a prostate without human guidance or annotations. Fig. 1 shows examples of prostate image slices.

The prostate segmentation is a challenging problem due to three main difficulties. First, prostate MR images have severe intra-image and inter-image variations, caused by endorectal coils [1,2]. Second, in the view of the medical imaging, the appearances of prostates are similar to those of other organs, such as seminal vesicles or bladders. Third, only a limited number of prostate MR images are available due to patients' privacy, and thus it is difficult to learn general characteristics of prostates.

To address these difficulties, semi-supervised prostate segmentation algorithms [2, 3] have been proposed, which require user annotations about prostate regions in a few slices. Malmberg *et al.* [3] propagate user annotations from seed voxels to the others. Tian *et al.* [2] over-segment each image slice into superpixels, and then dichotomize the superpixels into either prostate or non-prostate class based on the graph-cut optimization. Although [2, 3] yield reliable segments, the user annotations can be burdensome. To minimize user effort, unsupervised algorithms [1,4] also have been developed. Vincent *et al.* [4] construct a generative prostate model using appearance, position, and texture features. Mahapatra and Buhmann [1] segment a prostate using a random forest classifier. However, [1,4] use hand-crafted features, which can be easily overfitted.

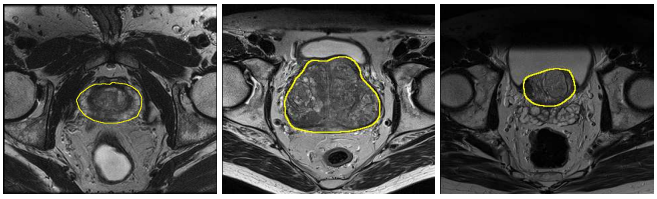


Fig. 1: Examples of 2D image slices of prostate MR images. The outlines of the prostates are depicted in yellow.

Recently, deep-learning-based algorithms [5–10] have demonstrated outstanding performances for natural or medical image segmentation. With this success of the deep-learning-based segmentation, automatic prostate segmentation algorithms have been proposed [11, 12]. Milletari *et al.* [11] separate prostate regions from an MR image via a CNN, which uses 3D convolutional filters. Yu *et al.* [12] adopt the residual feature forwarding [13] and also perform the sliding window sampling to obtain segments statistically. Although the Yu *et al.*'s algorithm [12] yields promising performances, it adopts the cross entropy as the objective function, without investigating other objective functions.

In this work, we analyze the impacts of six objective functions on the performance of CNN-based prostate segmentation: Hamming distance, Euclid distance, Jaccard similarity, dice coefficient, cosine similarity, and cross entropy. To this end, we develop a baseline CNN (BCNN), which can be trained to optimize each objective function. In the BCNN, we adopt the residual feature forwarding [13] and intermediate feature propagation [14] strategies for reliable and effective training. Then, we compare the segmentation results of the proposed BCNN using the six objective functions, from which important observations are made. Especially, the cosine similarity provides the best performance, whereas the widely-used cross entropy yields the worst performance.

Section 2 describes the BCNN and the six objective functions. Section 3 compares the objective functions quantitatively and qualitatively. Finally, Section 4 concludes this work.

This work was supported partly by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP)(No. NRF-2015R1A2A1A10055037), and partly by the Agency for Defense Development (ADD) and Defense Acquisition Program Administration (DAPA) of Korea (UC160016FD).

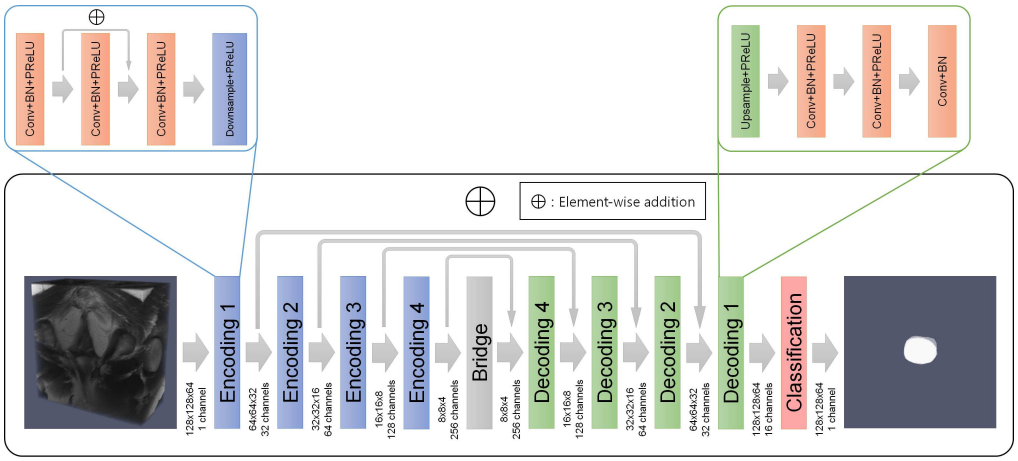


Fig. 2: The architecture of the proposed BCNN, which uses the encoding, bridge, decoding, and classification modules.

2. BCNN AND OBJECTIVE FUNCTIONS

Fig. 2 shows the architecture of the proposed BCNN, which accepts a 3D voxel image as input. We resize an input MR image into $128 \times 128 \times 64$. As a result, each voxel covers the physical volume of $0.625 \times 0.625 \times 1.5$ mm^3. For each voxel, the proposed BCNN infers the probability that it belongs to a prostate gland region.

2.1. Proposed BCNN Architecture

As shown in Fig. 2, the proposed BCNN consists of four kinds of modules: encoding, bridge, decoding, and classification modules. Each encoding module has three 3D convolutional layers (Conv) and a downsampling layer. The downsampling layer is implemented by a convolutional layer, the stride of which is 2. Thus, each downsampling layer reduces the spatial resolution by a factor of 2 axially, sagittally, and coronally. The batch normalization (BN) reduces the imbalance of inter-channel features for reliable training. We exploit the parametric rectified linear unit (PReLU) [15] as an activation function to prevent over-fitting. Also, the residual feature forwarding strategy [13] is used in the encoding modules to train the deep network effectively. The bridge module is the same as the encoding modules, except that it excludes the downsampling layer. Each decoding module performs upsampling, followed by three convolutional layers. We implement the upsampling layer using a deconvolutional layer, which enlarges the spatial resolution by a factor of 2 axially, sagittally, and coronally. The classification module predicts the probability that each voxel belongs to the prostate. Consequently, BCNN yields a segmentation map that has the same spatial resolution as the input image.

Deep features are extracted from the input image by the encoding modules. While these deep features contain prostate information, they may lose spatial details due to the downsampling layers. Hence, as done in [10], we exploit the intermediate features in the decoding phase, by employing the element-wise addition operator in Fig. 2. The intermediate features have low-level information, and thus can improve the qualities and details of segments [14].

2.2. Objective Functions

The proposed BCNN can be trained via the gradient descent method, in which the gradients are computed to optimize an objective function. Therefore, the attributes and the performance of the network are affected by the adopted objective function.

Table 1: Operational taxonomic units (OTUs) [17].

	Positive (Prediction)	Negative (Prediction)
Positive (Ground-truth)	$a = \sum_{i=1}^{N} p_i q_i$	$b = \sum_{i=1}^{N} (1 - p_i) q_i$
Negative (Ground-truth)	$c = \sum_{i=1}^{N} (1 - q_i) p_i$	$d = \sum_{i=1}^{N} (1 - p_i)(1 - q_i)$

We consider six objective functions [16]: Hamming distance, Euclidean distance, Jaccard similarity, dice coefficient, cosine similarity, and cross entropy. They are used to compute the similarity or dissimilarity between two binary signals. Note that the Hamming distance, the Eunclidean distance, and the cross entropy are dissimilarity measures and thus should be minimized, whereas the Jaccard similarity, the dice coefficient, and the cosine similarity are similarity measures to be maximized.

Except for the cross entropy, the objective functions can be compactly described by the operational taxonomic units (OTUs) [17]. Table 1 summarizes the four OTUs a, b, c, and d in terms of predicted and ground-truth labels. Let $P = \{p_1, \ldots, p_N\}$ be the set of predicted labels of voxels, obtained by a segmentation algorithm, and $Q = \{q_1, \ldots, q_N\}$ be the set of the ground-truth binary labels. Each OTU is computed by counting the number of voxels that have a certain pair of predicted and ground-truth labels. For example, a counts the number of voxels whose predicted and ground-truth labels are both 'positive.' In other words, it is the number of true positives. Similarly, b, c, and d count false negatives, false positives, and true negatives, respectively. Note that the sum of all four OTUs is $N = a + b + c + d$, where N is the number of voxels in an image.

2.2.1. *Hamming Distance*

The Hamming distance is the simplest distance metric, which considers only false negatives and false positives. It counts the number of wrong predictions and is commonly used to quantify mismatches between two binary signals. The Hamming distance is defined as

$$D_{\mathrm{H}} = b + c = \sum_{i=1}^{N} \left(p_i^2 + q_i^2 - 2 p_i q_i \right), \tag{1}$$

and its gradient is composed of the partial derivatives

$$\frac{\partial D_{\mathrm{H}}}{\partial p_j} = 2\left(p_j - q_j\right), \quad j = 1, \ldots, N. \tag{2}$$

In the training, since the Hamming distance is a dissimilarity measure, the gradient is directly employed in the backpropagation algorithm.

2.2.2. Euclidean Distance

The Euclidean distance is also widely used, since it coincides with the human conception of a distance. It is given by

$$D_{\mathrm{E}} = \sqrt{b + c} = \sqrt{\sum_{i=1}^{N} \left(p_i^2 + q_i^2 - 2p_i q_i\right)}, \tag{3}$$

and its partial derivatives are

$$\frac{\partial D_{\mathrm{E}}}{\partial p_j} = \frac{p_j - q_j}{\sqrt{\sum_i \left(p_i^2 + q_i^2 - 2p_i q_i\right)}}, \quad j = 1, \ldots, N. \tag{4}$$

As in the Hamming distance, the gradient is directly employed in the backpropagation.

2.2.3. Jaccard Index

The Jaccard index is a similarity measure but it does not consider true negatives. It is defined as

$$S_{\mathrm{J}} = \frac{a}{a + b + c} = \frac{\sum_{i=1}^{N} p_i q_i}{\sum_{i=1}^{N} \left(p_i^2 + q_i^2 - p_i q_i\right)}. \tag{5}$$

Notice that the Jaccard index is identical to the intersection-over-union (IoU) ratio, which is often used to evaluate segmentation algorithms. The gradient of the Jaccard index consists of the partial derivatives

$$\frac{\partial S_{\mathrm{J}}}{\partial p_j} = \frac{q_j \sum_i \left(p_i^2 + q_i^2 - p_i q_i\right) - \left(2p_j - q_j\right) \sum_i p_i q_i}{\left(\sum_i \left(p_i^2 + q_i^2 - p_i q_i\right)\right)^2} \tag{6}$$

where $j = 1, \ldots, N$. The negative of the gradient should be used in the backpropagation, since the objective function in (5) should be maximized.

2.2.4. Dice Coefficient

The dice coefficient (or Sørensen index) is a weighted version of the Jaccard index, which assigns bigger weights to true positives than to false positives or false negatives. It is given by

$$S_{\mathrm{D}} = \frac{2a}{2a + b + c} = \frac{2\sum_{i=1}^{N} p_i q_i}{\sum_{i=1}^{N} \left(p_i^2 + q_i^2\right)}. \tag{7}$$

Note that the conventional prostate segmentation algorithm [11] uses the dice coefficient as its objective function. Furthermore, the dice coefficient is used as the evaluation metric in the PROMISE12 challenge [18]. The partial derivatives are

$$\frac{\partial S_{\mathrm{D}}}{\partial p_j} = \frac{2q_j \sum_i \left(p_i^2 + q_i^2\right) - 4p_j \sum_i p_i q_i}{\left(\sum_i \left(p_i^2 + q_i^2\right)\right)^2} \tag{8}$$

where $j = 1, \ldots, N$. In the training, the negative of the gradient is used in the backpropagation.

Table 2: Segmentation scores of the proposed BCNN using different objective functions. The best and the second best results are bold-faced and underlined, respectively. For comparison, the score of [12] is also included.

Algorithm	Objective function	Score
BCNN	Hamming distance	0.8366
	Euclidean distance	0.8467
	Jaccard similarity	0.8291
	Dice coefficient	<u>0.8507</u>
	Cosine similarity	**0.8537**
	Cross entropy	0.8275
[12]	Cross entropy	0.8693

2.2.5. Cosine Similarity

The cosine similarity computes the cosine of the angle between two signals (or vectors), given by

$$S_{\mathrm{C}} = \frac{a}{\sqrt{(a + b)(a + c)}} = \frac{\sum_{i=1}^{N} p_i q_i}{\sqrt{\sum_{i=1}^{N} p_i^2 \sum_{i=1}^{N} q_i^2}}, \tag{9}$$

and its partial derivatives are

$$\frac{\partial S_{\mathrm{C}}}{\partial p_j} = \frac{q_j \sum_i p_i^2 \sum_i q_i^2 - p_j \sum_i p_i q_i \sum_i q_i^2}{\left(\sum_i p_i^2 \sum_i q_i^2\right)^{\frac{3}{2}}} \tag{10}$$

where $j = 1, \ldots, N$. Since it is another similarity measure, the negative of the gradient is used for the training.

2.2.6. Cross Entropy

The cross entropy is adopted as an objective function in many deep-learning-based classification algorithms [13, 19, 20]. As an information theoretic quantity, it is directly defined as

$$D_{\mathrm{C}} = -\sum_{i=1}^{N} q_i \log p_i - \sum_{i=1}^{N} (1 - q_i) \log(1 - p_i), \tag{11}$$

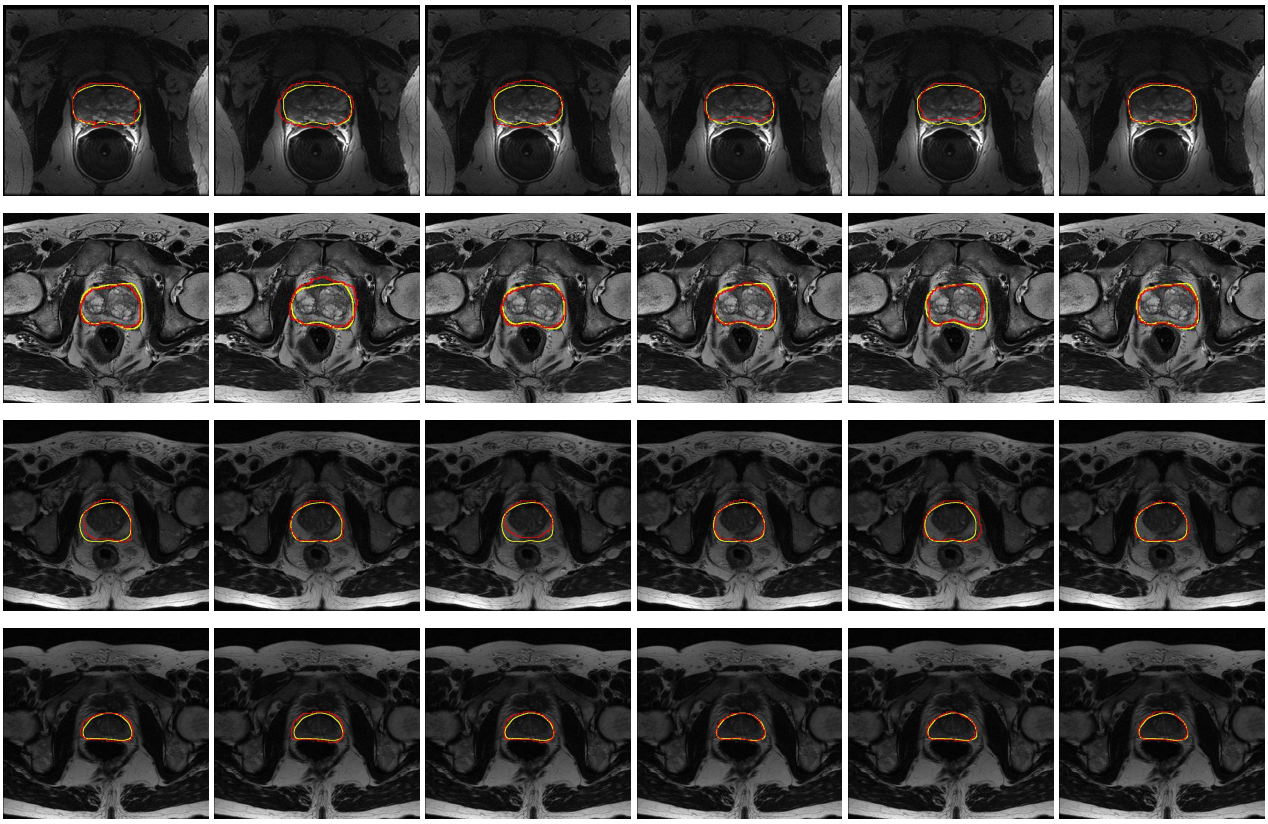
instead of being described with the OTUs. Its partial derivatives are

$$\frac{\partial D_{\mathrm{C}}}{\partial p_j} = -\frac{q_j}{p_j} + \frac{1 - q_j}{1 - p_j}, \quad j = 1, \ldots, N. \tag{12}$$

The cross entropy quantifies a dissimilarity, and thus the gradient is employed for the gradient descent.

2.3. Training Phase

As mentioned before, the lack of data makes it difficult to train a CNN for the prostate segmentation. Only the PROMISE12 training dataset [18] of 50 prostate MR images is available online. Hence, we augment the training dataset by applying the flipping and the deformable transformation [11] to the training images. For every training instance, we perform the flipping and the deformable transformation independently, both with a probability of 0.5. We use the TensorFlow [21] to implement the objective function layers, and train the proposed BCNN using the Adam optimizer. The initial learning rate is set to 0.005 and then reduced to 0.0005 after 10,000 iterations. We stop the training after 15,000 iterations, which requires about eight hours using an NVIDIA GTX Titan X GPU.



(a) Cross entropy (b) Jaccard similarity (c) Hamming distance (d) Euclidean distance (e) Dice coefficient (f) Cosine similarity

Fig. 3: Qualitative comparison of the six objective functions for training the proposed BCNN. Each row is from a different MR image. The yellow and red boundaries outline the ground-truth and predicted prostate segments, respectively.

3. EXPERIMENTAL RESULTS

We evaluate the proposed BCNN on the PROMISE12 training dataset [18], which is composed of 50 prostate images. Because of the lack of data, we adopt the 10-fold cross validation method: We first split the dataset into 10 subsets. Then, to measure the performance on a subset, we train the BCNN using the other nine subsets. We do this process 10 times to assess the BCNN on all the subsets. We use the dice coefficient score as the evaluation metric, as done in [11, 12]. To segment a prostate image, the proposed BCNN takes about 10 seconds on a PC with an Intel Xeon E5-2690 2.60GHz CPU and an NVIDIA GTX Titan X GPU.

Table 2 compares the segmentation performances (dice coefficient scores) of the proposed BCNN using different objective functions. Notice that the cosine similarity outperforms the other objective functions. It is worth pointing out that the cross entropy yields the worst performance, even though it has been used for training the conventional prostate segmentation algorithms [11, 12]. The Euclidean distance makes the BCNN perform better than the Hamming distance does. This is because the denominator in the derivative of the Euclidean distance in (4) normalizes the scale of the gradient. By comparing the Jaccard index with the dice coefficient, we see that the segmentation performance is improved by assigning bigger weights to true positives. For comparison, Table 2 also includes the performance of Yu *et al.*'s algorithm [12], which outperforms the proposed BCNN. However, the goal of this work is to analyze the objective functions for the prostate segmentation. Hence, we do not adopt complicated techniques, such as the sliding window sampling strategy [12], for training the BCNN. The performance of the BCNN can be further improved if we use these techniques as well.

Fig. 3 shows exemplar slices of prostate segmentation results. It is observable that the cosine similarity is more accurate in discovering prostate glands than the other objective functions are. This indicates that Yu *et al.*'s algorithm [12] also may yield more accurate prostate segments by employing the cosine similarity in stead of the cross entropy. When the source codes of [12] become available, we will investigate this possibility.

4. CONCLUSIONS

We proposed a deep-learning-based baseline algorithm, called BCNN, for the prostate segmentation, which uses the residual feature forwarding and intermediate feature propagation strategies. Then, we introduced the six objective functions and derived their gradients. We compared the performances of these objective functions on the PROMISE12 dataset [18]. Experimental results demonstrated that the BCNN using the cross entropy, which is commonly adopted in the conventional prostate segmentation algorithms [11, 12], provides the worst performance. On the other hand, the BCNN using the cosine similarity perfoms the best. Future research issues include the development of a more sophisticated deep-learning-based algorithm for medical image segmentation that uses the cosine similarity as the objective function.

5. REFERENCES

[1] D. Mahapatra and J. M. Buhmann, "Visual saliency-based active learning for prostate magnetic resonance imaging segmentation," *Journal of Medical Imaging*, vol. 3, no. 1, 2016. 1

[2] Z. Tian, L. Liu, Z. Zhang, and B. Fei, "Superpixel-based segmentation for 3D prostate MR images," *IEEE Trans. Medical Imaging*, vol. 35, no. 3, 2016. 1

[3] F. Malmberg, R. Strand, J. Kullberg, R. Nordenskjöld, and E. Bengtsson, "Smart Paint - A new interactive segmentation method applied to MR prostate segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention, Grand Challenge Workshop*, 2012. 1

[4] G. Vincent, G. Guillard, and M. Bowes, "Fully automatic segmentation of the prostate using active appearance models," in *International Conference on Medical Image Computing and Computer-Assisted Intervention, Grand Challenge Workshop*, 2012. 1

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE CVPR*, 2015. 1

[6] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE ICCV*, 2015. 1

[7] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE CVPR*, 2016. 1

[8] P. O. Pinheiro, R. Collobert, and P. Dollar, "Learning to segment object candidates," in *Advances in Neural Information Processing Systems*, 2015. 1

[9] N. Xu, B. Price, S. Cohen, J. Yang, and T. S Huang, "Deep interactive object selection," in *Proc. IEEE CVPR*, 2016. 1

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015. 1, 2

[11] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. IEEE International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2016. 1, 3, 4

[12] L. Yu, X. Yang, H. Chen, J. Qin, and P.-A. Heng, "Volumetric ConvNets with mixed residual connections for automated prostate segmentation from 3D MR images," in *AAAI Conference on Artificial Intelligence*, 2017. 1, 3, 4

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016. 1, 2, 3

[14] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *European Conference on Computer Vision*, 2016. 1, 2

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE ICCV*, 2015. 2

[16] S.-S. Choi, S.-H. Cha, and C. C. Tappert, "A survey of binary similarity and distance measures," *Journal of Systemics, Cybernetics and Informatics*, vol. 8, no. 1, 2010. 2

[17] G. Dunn and B. S. Everitt, *An Introduction to Mathematical Taxonomy*, Courier Corporation, 2004. 2

[18] PROMISE12. [Online]. Available: https://promise12.grand-challenge.org. 3, 4

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. 3

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015. 3

[21] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., "TensorFlow: A system for large-scale machine learning," in *Proceedings of the USENIX Symposium on OSDI*, 2016. 3