IEEE Transactions on Image Processing Volume: 28 , Issue: 8 , Aug 2019

Crossbar-Net: A Novel Convolutional Neural Network for Kidney Tumor Segmentation in CT Images

Qian Yu, Yinghuan Shi*, Jinquan Sun, Yang Gao*, Jianbing Zhu, Yakang Dai

***Abstract*—Due to the unpredictable location, fuzzy texture and diverse shape, accurate segmentation of the kidney tumor in CT images is an important yet challenging task. To this end, we in this paper present a cascaded trainable segmentation model termed as Crossbar-Net. Our method combines two novel schemes: (1) we originally proposed the crossbar patches, which consists of two orthogonal non-squared patches (*i.e.*, the vertical patch and horizontal patch). The crossbar patches are able to capture both the global and local appearance information of the kidney tumors from both the vertical and horizontal directions simultaneously. (2) With the obtained crossbar patches, we iteratively train two sub-models (*i.e.*, horizontal sub-model and vertical sub-model) in a cascaded training manner. During the training, the trained sub-models are encouraged to become more focus on the difficult parts of the tumor automatically (*i.e.*, mis-segmented regions). Specifically, the vertical (horizontal) sub-model is required to help segment the mis-segmented regions for the horizontal (vertical) sub-model. Thus, the two sub-models could complement each other to achieve the self-improvement until convergence. In the experiment, we evaluate our method on a real CT kidney tumor dataset which is collected from 94 different patients including 3,500 CT slices. Compared with the state-of-the-art segmentation methods, the results demonstrate the superior performance of our method on the Dice similarity coefficient, true positive fraction, centroid distance and Hausdorff distance. Moreover, to exploit the generalization to other segmentation tasks, we also extend our Crossbar-Net to two related segmentation tasks: (1) cardiac segmentation in MR images and (2) breast mass segmentation in X-ray images, showing the promising results for these two tasks. Our implementation is released at https://github.com/Qianyu1226/Crossbar-Net.**

This work was supported by the NSFC (61432008, 61673203), Young Elite Scientists Sponsorship Program by CAST (YESS 2016QNRC001), CCF-Tencent Open Research Fund (RAGR 20180114), Projects of Shandong Province Higher Educational Science and Technology Program (J18KA370, J15LN58), Project of Shandong Medicine and Health Science Technology Development Plan (2017WSB04071), and Shandong Province Science and Technology Development Plan Project (2014GSF118086). This work was supported by Zhejiang Key Technology Research Development Program (2018C03024), Jiangsu Key Technology Research Development Program (BE2017664), Suzhou Science and Technology Projects for People's Livelihood (SYS2018010), Suzhou Science and Technology Development Project (SZS201818), and SND Medical Plan Project (2016Z010, 2017Z005).

Corresponding Authors: Yang Gao and Yinghuan Shi.

Qian Yu, Yinghuan Shi, Jinquan Sun, and Yang Gao are with the State Key Laboratory for Novel Software Technology, the National Research Institute for Big Data Science in Health and Medicine, and the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University, China. Qian Yu is also with School of Data and Computer Science, Shandong Women's University, China. (e-mail: yuqian@sdwu.edu.cn, syh@nju.edu.cn, jinquansun@gmail.com, gaoy@nju.edu.cn)

Jianbing Zhu is with the Suzhou Science and Technology Town Hospital, China. (e-mail: zeno1839@126.com)

Yakang Dai is with the Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences, China. (e-mail: daiyk@sibet.ac.cn)

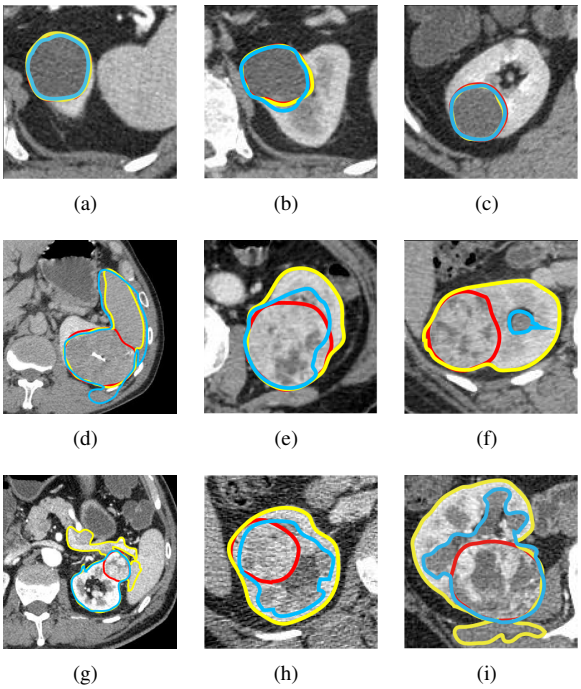


Fig. 1. Typical images of kidney tumors. The red, yellow and light blue contours denote the ground truth, predictions of the energy-based method [1] and traditional learning-based method [2], respectively. Extensive comparisons with other state-of-the-art models are reported in Section IV.

***Index Terms*—Deep Convolutional Neural Network, Kidney Tumors, Crossbar-Net, Image Segmentation, CT Images.**

I. Introduction

RENAL cell carcinoma is a common urologic cancer arising from renal cortex [3]–[5]. Accurate quantification and correct classification of tumors could largely influence the effect of the following computer-aided treatment of renal cell carcinoma [6]. In this meaning, for the quantification and classification, the accurate kidney tumor segmentation is a significant prerequisite. Traditional human-based manual delineation for kidney tumor segmentation is not desirable in clinical practice, due to both the subjective (*e.g.*, incorrect delineation) and objective (*e.g.*, a large number of images) factors. Thus, computer-aided automatic segmentation methods for kidney tumors (in CT images) are in high demand.

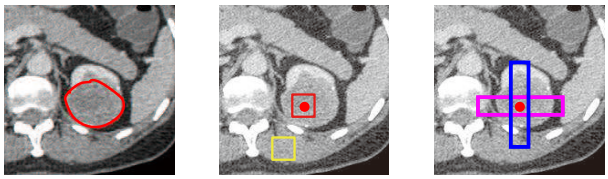
However, segmenting the kidney tumors automatically in CT images is a very challenging task. According to the clinical and experimental observation,

- The location of different kidney tumors in medical images is difficult to predict since the tumors could possibly appear in very different places between different patients.
- Different tumors between different patients usually show very diverse shape appearance and volumetric size according to the different growth stages.
- The tumors and their surrounding tissues are with very similar texture information due to the low contrast of CT images.

Although several works have been proposed recently [1], [2], [6]–[9], their segmentation performance could not be robustly guaranteed in different cases (see Fig. 1). Both (1) the intensity dissimilarity within different parts of tumors and (2) the similar appearance between kidney tumors and their surrounding tissues pose the great technical challenges for developing robust segmentation models. In order to visually illustrate these challenges, we have segmented several typical kidney tumors in CT images by introducing two representative models: the energy minimization-based model [1] and the traditional learning-based model [2]. Please note that the extensive comparison with other state-of-the-art models is reported in our experimental part. As shown in Fig. 1(a) - 1(c), the tumors with high contrast and clear boundaries could be well segmented by traditional segmentation methods [1], [2]. However, [1], [2] will fail in more difficult cases. For example, the tumor in Fig. 1(d) is strongly connected to its surrounding tissue and meanwhile shows a similar intensity with that of the tissue, which leads [1], [2] fail to segment. Similarly, all the tumors in Fig. 1(e) - Fig. 1(f) show very similar visual characteristics with the outside renal parenchyma. In addition, the tumors in Fig. 1(g) - Fig. 1(i) are challenging cases because they are with intensity dissimilarity within different parts inside the tumor. Therefore, the advanced efforts on accurate kidney tumor segmentation are still required to meet the clinic requirement.

The key issue of accurate segmentation is how to well distinguish the tumor and non-tumor boundary by extracting (or learning) the informative and discriminative features. Recent trends of deep convolutional neural network (CNN) have demonstrated the superior performance on learning-based segmentation tasks in different imaging modalities for different organs, *e.g.*, prostate [10], [11], heart [12]–[14], brain [15]–[17]. Hence, in this paper, we present a CNN-based model for CT kidney tumor segmentation. Previous CNN-based segmentation methods could be roughly classified into two categories: the image-based CNN models [18]–[22] and the patch-based CNN models [10], [23], [24]. Both of these previous methods treated either whole images or squared patches as the training samples to first learn the segmentation model and then employ the obtained models to segment the new coming testing images.

Unlike the traditional image- or patch-based CNN models, we originally propose the new findings for kidney tumor segmentation in CT images. Specifically, in CT images, the kidney tumors normally appear as the subrounded shape with a certain degree of symmetry. This observation inspires us that we could leverage the shape information for kidney tumor segmentation to achieve a promising performance. Unfortunately, this specific shape information is usually ignored in previous CNN-based methods. For example, if we extract squared patches as the existing patch-based CNN models, as shown in Fig. 2(b), the red and yellow patches with only captured local information cannot distinguish the tumor and its surrounding organs obviously. Alternatively, we can enlarge the size of the squared patches or apply the whole image-based methods directly to cover the whole tumor and its context. However, in this case, the irrelevant noise might be brought in at the same time, which might cause that the local details, especially the boundary details, might be ignored. The following results in Fig. 9 and Table III support this point. Basically, it is known that, in a same region area, compared with the squared patch, the non-squared rectangular patch could capture more information typically from one direction (*i.e.*, horizontal or vertical). If we sample non-squared patches (named as crossbar patches in this paper) as Fig. 2(c) to fully cover the whole tumor along one direction from side-to-side, we indeed could integrate more contextual and symmetrical information simultaneously.



(a) (b) (c)

Fig. 2. An example of the kidney tumors and different types of patches. (a) Ground truth. (b) Squared patches. (c) Our crossbar patches.

Thus, we innovatively in this paper propose crossbar patches which consist of the vertical patch and horizontal patch, aiming to jointly capture (1) the local detail information and (2) global contextual information from vertical and horizontal directions, respectively. In addition, on the obtained crossbar patches, we originally present a cascaded training framework to iteratively train the sub-models (namely vertical sub-model and horizontal sub-model) from these two directions. It is noteworthy that our training and testing are performed on the pixel-wise since we convert the segmentation task to a pixel-wise classification problem as the traditional setting [10], [23], [24].

In particular, during the training process, the trained vertical and horizontal sub-models are encouraged to help each other in a way of asking the other one to help segment its difficult parts. Taking the vertical sub-model as an example, if it cannot segment a region correctly in the vertical direction, the horizontal sub-model could complement the unsatisfactory segmentation in the horizontal direction. Also, the vertical sub-model is required to help the horizontal sub-model in the same way. Thus, the two sub-models could complement each other to achieve the self-improvement until convergence. Additionally, to make the training process more effective and efficient, we propose two sampling strategies (*i.e.*, the *basic sampling strategy* and the *covering re-sampling strategy*). The former samples the discriminative patches and balances the different classes (*i.e.*, tumor or non-tumor) to allow the

efficient model training with less patch redundancy, while the latter guarantees the complementary help between different sub-models. These two strategies facilitate self-improvement for these sub-models together.

Since our proposed method involves the sampled crossbar patches from two directions, and the cascaded training process to iteratively train the vertical and horizontal sub-models, we name our method as Crossbar-Net in the following parts. Overall, the contributions of our work can be summarized in the following four folds:

- Our crossbar patches could capture both the local detail information and global contextual information. Also, these patches are easy to sample without introducing any additional parameters to train.
- Our cascaded training process could help provide complementary information between different sub-models to enhance the final segmentation. In our training process, the sub-models can perform the self-improvement iteratively.
- Our model is easy to implement and extend. Beyond kidney tumor segmentation in CT images, we have evaluated our method on cardiac segmentation in MR images and breast mass segmentation in X-ray images, showing promising results and good generalization ability.
- Our method is fast to train and test, although it is trained in a cascaded manner. Taking kidney tumor as an example, the training time is about 1h and the testing time is about 1.5s on a regular GPU.

The rest of this paper is organized as follows. Related works about kidney tumor segmentation in recent years are briefly introduced in Section II. We then describe the framework and technical details of Crossbar-Net in Section III. Experimental results are reported in Section IV. Finally, we conclude our paper in Section V.

II. Related Work

For kidney tumor segmentation, according to the way of feature representation, most of the previous methods belong to the low-level methods. The low-level methods either employ the energy minimization-based models or learn the segmentation model on the extracted hand-crafted features. For example, Skalski *et al.* [2] first located the kidney region through a hybrid level set method with the ellipsoidal shape constraint, then calculated the low-level features (*e.g.*, mean value, standard deviation, histogram of oriented gradients [25]) on the obtained region, and finally performed the decision tree to distinguish the kidney tumor and arterial blood vessel regions. Linguraru *et al.* [1] first extracted kidney tumors by the region growing for initial segmentation, and then applied geodesic active contours to refine the segmentation result. Also, Linguraru *et al.* [9] described the kidney tumors using the low-level visual features (*e.g.*, histograms of curvature features). Similarly, Lee *et al.* [7] first detected region-of-interest by analyzing the textural and contextual information, and then extracted the mass candidates with the region growing and active contours. Hodgdon *et al.* [8] first extracted the texture features (*i.e.*, gray-level histogram mean and variance, gray-level co-occurrence and run-length matrix features [26]), and then trained a support vector machine (SVM) to segment the fat-poor angiomyolipoma from renal cell carcinoma in CT images. These aforementioned low-level methods perform well in the simple case when the tumors show different appearance with surrounding tissues. However, their performance could not be fully guaranteed when the shape and texture of the tumors are close to the surrounding tissues.

Recent trends of using deep features or deep models have demonstrated the effectiveness in several segmentation tasks. Although the attempts of developing the specific deep feature-based methods for segmenting kidney tumor are very limited, the related deep feature-based segmentation methods for segmenting other medical organs [23], [10], [15], [24], [27], can be borrowed to segment the kidney tumor. For instance, Ciresan *et al.* [23] employed multiple deep networks to segment biological neuron membranes by extracting the squared patches in multi-scales with sliding-window. Prasoon *et al.* [27] designed a triplanar CNN model to segment cartilage with multiple view patches. Moeskops *et al.* [15] proposed a multi-scale CNN model to segment brain MR images. Wang *et al.* [24] devised a multi-branch CNN model to segment lung nodules. Shi *et al.* [10] proposed a cascaded deep domain adaptation model to segment the prostate in CT images. However, we notice that the squared patches used in these above works are one of the major bottlenecks when borrowing them to segment the kidney tumors which are experimentally demonstrated in our experiment. To address the limitation of the local squared patch, several attempts about the image-level segmentation are exploited in recent years. Mortazi *et al.* [12] presented a novel multi-view CNN model to fuse the information from the axial, sagittal, and coronal views of cardiac MRI. This model performed well in the task of the left atrium and proximal pulmonary veins segmentation. Ronneberger *et al.* [19] proposed a widely-used model in medical image segmentation tasks, namely U-Net, to solve the cell tracking problem. Recently, a new image-based model, SegCaps [22] was developed which achieved a promising result in the task of segmenting pathological lungs from low dose CT scans. Also, He *et al.* [28] introduced a fully convolutional network with distinctive curve to segment the pelvic organ.

In summary, compared with the previous segmentation methods, all of them employ either the image-level or squared patch-level segmentation, while our Crossbar-Net involves the crossbar patches (non-squared patches) to capture the both (1) local detail information and (2) global context information from vertical and horizontal directions.

Recently, training deep models in a cascaded or boosting-like manner for better performance have aroused considerable interests [29]–[33]. For example, Shwartz *et al.* [32] proposed a SelfieBoost model, which boosted the performance of a single network based on minimizing the maximal loss. Karianakis *et al.* [30] proposed an object detection framework according to the boosted hierarchical features. Walach *et al.* [29] introduced a boosted-CNN model where the latter CNN was added according to the error of the former CNN, and finally all CNNs were joined via a sum layer. Similarly, Havaei

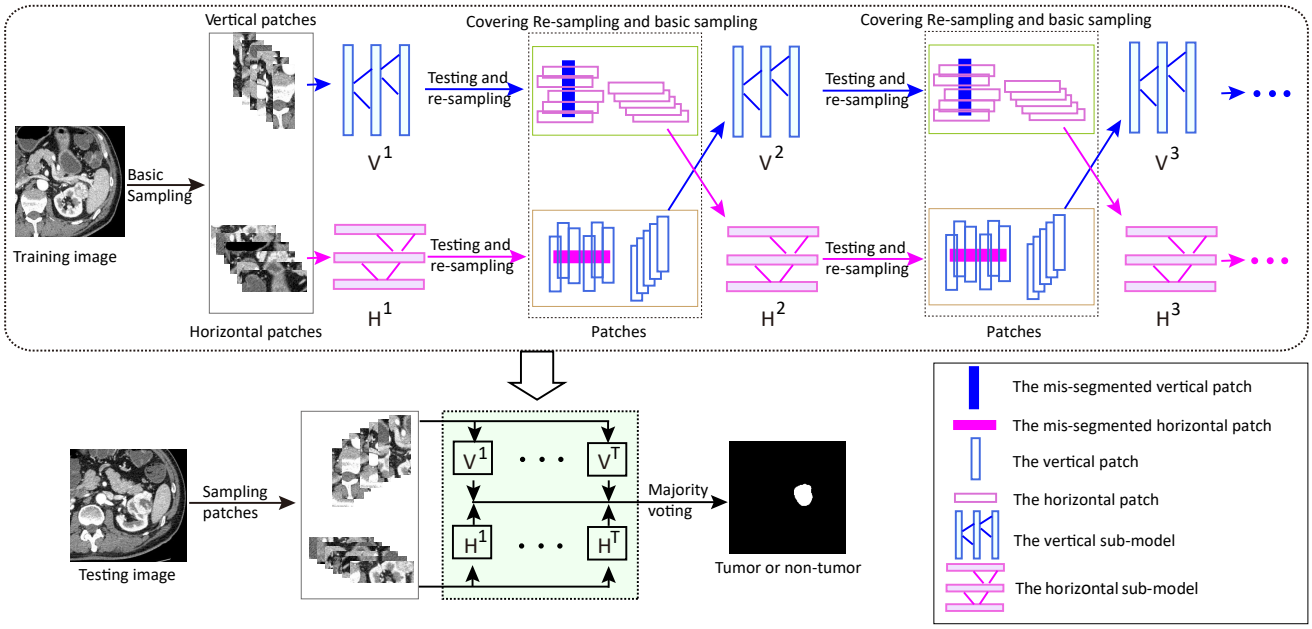


Fig. 3. Framework of the proposed method. $\mathbf{V}^i$ and $\mathbf{H}^i$ are vertical sub-model and horizontal sub-model of the i-th round, respectively, $i = 1, \cdots, T$.

et al. [33] presented a cascaded architecture consisting of two CNNs to segment glioblastomas in MR images, where the output probabilities of the first CNN was added to the layers of the second CNN. For these above methods, multi-modal fusion and enhancement in a boosting-like manner are the common choices. Also, our Crossbar-Net consists of the fusion from vertical and horizontal sub-models. As for what to enhance, in addition to hierarchical features [30], [33], the misclassified samples are enhanced in the next rounds to raise the concerns from the classifier. Adaboost [34] and co-training models [35]–[39] are the typical misclassified-samples-enhancing methods. Inspired by this, we train a cascaded model composed of vertical and horizontal sub-models by enhancing the region around misclassified pixels.

Compared with the previous cascaded methods, the major distinctions of Crossbar-Net are (1) our model learns both local detail features and global context features from two directions simultaneously, and (2) our model is composed of the sub-models from two directions, in which the sub-models can perform the self-improvement during different rounds to complement each other iteratively.

III. Method

In this section, we first introduce our methodology and sampling strategy of crossbar patch, then present the sub-model setup and illustrate the training process, and finally discuss the testing process.

A. Our Methodology

The framework of Crossbar-Net is schematized in Fig. 3, which includes the training and testing stages. Note that, we convert our segmentation task to a pixel-wise classification problem, which intends to predict a patch to be a tumor or non-tumor class.

In the training stage, firstly, the crossbar patches are initially extracted from the training CT images under the *basic sampling strategy* (detailed in Section III-B) with the manual segmentation available as the ground truth. Also, the vertical and horizontal sub-models are initially trained in the 1-st round, denoted as $\mathbf{V}^1$ and $\mathbf{H}^1$, respectively. Then, regarding the cascaded training process, at the t-th round, we evaluate the segmentation performance of the current trained vertical and horizontal sub-models, and select the mis-segmented regions of each sub-model. Formally, the mis-segmented region indicates the vertical or horizontal patch whose central pixel is misclassified, *i.e.*, if a central pixel along with its located vertical patch is misclassified by a vertical sub-model, then its corresponding vertical patch is a mis-segmented region. And then, we re-sample the mis-segmented regions using the *covering re-sampling strategy* (detailed in Section III-B) to obtain the corresponding re-sampling patches. Then, we feed these re-sampling patches to another sub-model for its model training. Meanwhile, beyond the aforementioned re-sampling patches, in each round, the patches sampled under the *basic sampling strategy* are also feed to the same sub-model. We keep repeating the above process until the training error converges or the maximum round number reaches.

In the testing stage for segmenting a new coming CT image, the trained sub-models in each round are gathered together to perform a majority voting on this image to obtain the final segmentation.

B. Crossbar Patch Sampling Strategy

1) Basic Sampling Strategy: We develop the *basic sampling strategy* with the goal of making the segmentation model more focus on the region surrounding the tumor boundary which is considered to be hard to segment in practice [40]. Therefore, the principle we adhere to is to increase the patches close to

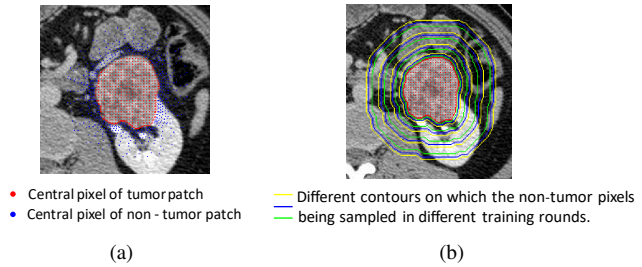


Fig. 4. The typical example of *basic sampling strategy*.

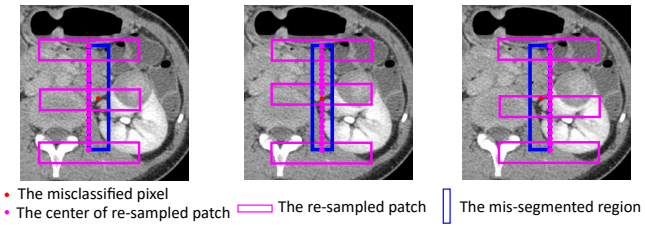


Fig. 5. Illustration of *covering re-sampling strategy*. The blue patch is the mis-segmented region with the red central pixel, and the magenta rectangles are the re-sampling patches with central pixels being located at the left, middle and right line of the blue patch.

the tumor and reduce the redundant patches that are far from the tumor. This sampling strategy is used in each round of the cascaded training with different sampling intervals.

By using this strategy, we select a part of total pixels according to the distance between the current pixel and the center of the tumor. We first extract crossbar patches uniformly in the tumor region as the training samples belonging to the tumor class (*i.e.*, tumor patch), and then sample non-tumor patches densely near the tumor and sparsely in the far region as the training samples belonging to the non-tumor class (*i.e.*, non-tumor patch). As shown in Fig. 4(a), the red pixels are the center of the tumor patches with certain intervals and the blue pixels are the center of the non-tumor patches. More details are described in Section III-D.

2) Covering Re-sampling Strategy: As shown in Fig. 5, we assume that the vertical patch is a mis-segmented region in vertical sub-model $\mathbf{V}^t$ at the t-th round. Our purpose is to borrow the horizontal sub-model to well segment this region using the horizontal patches. In particular, we extract the horizontal patches by fully covering the mis-segmented region, according to the location of the misclassified central pixel in the vertical sub-model. In order to cover this region, we sample the horizontal patches with the central pixel being located at three columns: the center, right and left column of the vertical patch. To avoid sampling redundant horizontal patches, we perform re-sampling by every three pixels on each column. Normally, for a vertical patch, we can roughly obtain ~ 40 horizontal patches. Thus, the horizontal sub-models can provide a complement to the vertical sub-models. Moreover, in this way, if both sub-models, *i.e.*, $\mathbf{V}^t$ and $\mathbf{H}^t$, fail on a same pixel, the role of the region around this pixel will definitely be enhanced in the next round. Thus, with the combination of the re-sampling patches and the aforementioned basic-sampling patches as the training samples, the performance of $\mathbf{V}^{t+1}$ is expected to be superior to $\mathbf{V}^t$, and $\mathbf{H}^{t+1}$ is also expected to outperform $\mathbf{H}^t$.

C. Sub-Model Architecture

The architecture of sub-models is designed according to a preliminary study on our CT kidney tumor dataset. The number of layers, kernel size and the amount of feature maps are all experimentally determined by inner cross-validation. Basically, both vertical and horizontal sub-models consist of eight convolutional layers, two max-pooling layers, and one softmax layer. Details of sub-model are illustrated in Table I. For the vertical sub-model, the input is a 100×20 vertical patch. Regarding the patch is non-square, the sizes of convolutional kernels in the first four convolutional layers are all set to 5×3, while that in the last four convolutional layers are set to 6×1, 6×1, 7×1 and 1×1, respectively. Each convolutional layer is followed by the rectified linear unit (ReLU) [41] activation and performed with 1 stride and 0 padding. The kernel size of each pooling layer is 2×2 with 2 strides and 0 padding. In addition, the dropout [42] after the last convolutional layer is applied to avoid the possible over-fitting. For each sub-model, we minimize the following softmax loss L:

$$L = -\sum_{i=1}^{N} y_i \log \widehat{y}_i \tag{1}$$

where N is the number of training patches (vertical or horizontal patches) in the current sub-model, y_i and $\widehat{y}_i$ are the ground truth and the predicated label of the central pixel in the i-th patch, respectively. The weights of the filters are initialized randomly with the Gaussian distribution [43] and updated by the stochastic gradient descent (SGD) algorithm.

Similarly, we can also obtain the architecture of the horizontal sub-model as illustrated in Table I, with the input of the sub-model as a 20×100 horizontal patch. Please note that the architecture could be adjusted according to different segmentation scenarios. Typically, we use the same architecture in the cardiac segmentation and different architecture in the breast mass segmentation.

D. Training Sub-Models

We now discuss how to train our Crossbar-Net in a cascaded manner, as a boosting-like training style. Formally, we denote the vertical and horizontal sub-models in the i-th round as $\mathbf{V}^i$ and $\mathbf{H}^i$, respectively. The training process can be detailed as follows:

Firstly, extracting crossbar patches with the *basic sampling strategy* and training the initial vertical and horizontal sub-models, *i.e.*, $\mathbf{V}^1$ and $\mathbf{H}^1$, respectively. We use an example to illustrate the detailed implementation process of *basic sampling strategy*:

As shown in Fig. 4(b), in this round, we first sample pixels on the green contours as the centers of non-tumor patches. Then, we select the red points as the centers of tumor patches on the odd rows inside the tumor. The sampling interval on the contours near the tumor is smaller than that on the contours which are far away from the tumor, with the goal of sampling

TABLE I
DETAILS OF SUB-MODEL ARCHITECTURES. CONV AND POOLING DENOTE CONVOLUTIONAL LAYER AND POOLING LAYER, RESPECTIVELY.

	Layer	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	# 9	# 10	# 11
Vertical sub-model	Layer type	Conv	Conv	Pooling	Conv	Pooling	Conv	Conv	Conv	Conv	Conv	Softmax
	Feature maps	16	36	36	64	64	64	64	64	500	2	2
	Kernel size	5×3	5×3	2×2	5×3	2×2	5×3	6×1	6×1	7×1	1×1	-
Horizontal sub-model	Layer type	Conv	Conv	Pooling	Conv	Pooling	Conv	Conv	Conv	Conv	Conv	Softmax
	Feature maps	16	36	36	64	64	64	64	64	500	2	2
	Kernel size	3×5	3×5	2×2	3×5	2×2	3×5	1×6	1×6	1×7	1×1	-

more patches near the tumor boundary. Here, we denote the sets of these pixels (*i.e.*, the location of these pixels) and their ground truth labels as X and Y, and the corresponding vertical and horizontal training patches as $P^{V^1}_{\text{basic}}$ and $P^{H^1}_{\text{basic}}$, respectively.

Secondly, we continuously update our Crossbar-Net based on the currently obtained sub-models. Specifically, in the i-th round ($i > 1$),

- Performing the evaluation on $\mathbf{H}^{i-1}$. In particular, we input the $P^{H^1}_{\text{basic}}$ to $\mathbf{H}^{i-1}$ for the evaluation at the i-th round, by predicting the label of central pixel in each patch from $P^{H^1}_{\text{basic}}$. In this meaning, the mis-segmented regions in $\mathbf{H}^{i-1}$ are determined according to the predicated labels. Formally, assuming that all these predicted labels as $\widehat{Y}_H$, we define the misclassified pixels as:

$$C^{i-1}_H = \left\{ x_j | x_j \in X \wedge I(\widehat{y}_j \neq y_j), j = 1, ..., N \right\} \quad (2)$$

  where x_j is the central pixel of the j-th patch, $\widehat{y}_j$ and y_j are the predicted and ground truth label of x_j, $\widehat{y}_j \in \widehat{Y}_H$ and $y_j \in Y$. N is the number of horizontal training patches. $I(s)$ is an indicator function. $I(s) = 1$ if and only if the statement s is true and $I(s) = 0$ otherwise.
- Performing both *covering re-sampling strategy* (on the mis-segmented regions in $\mathbf{H}^{i-1}$) and the *basic sampling strategy*, to obtain the newly generated vertical patches. Firstly, the covering re-sampling is performed. We also record the position of the patches obtained by *covering re-sampling strategy* in the current round, denoted as $P^{V^i}_{\text{re}}$. Then, the *basic sampling strategy* is sequentially performed. Specifically, as i varies, as shown in Fig. 4(b), we sample central pixels for non-tumor patches on different contours which are labeled by different colors. For the tumor patches, we sample the red points on different rows or columns as the central pixels. The sampling intervals for these two types of patches are different from that in the previous rounds. Meanwhile, if a patch has already been extracted in the covering re-sampling, it cannot be sampled in the basic sampling in the same round. The advantage of the above way is to avoid redundant sampling which might cause over-fitting. Here, the *covering re-sampling strategy* aims to enhance the role of mis-segmented regions, while the *basic sampling strategy* wishes to control the amount and distribution of training samples and prevent the sub-model from over-emphasizing the misclassified pixels.
- Employing these newly generated patches to train $\mathbf{V}^{i-1}$ to obtain $\mathbf{V}^i$.
- Updating the $\mathbf{H}^i$ from $\mathbf{H}^{i-1}$, similarly.

Finally, we repeat the aforementioned steps by updating $\mathbf{H}^i$ and $\mathbf{V}^i$ from $\mathbf{H}^{i-1}$ and $\mathbf{V}^{i-1}$ iteratively, until (1) maximum training round number reaches or (2) the training error of each sub-model could not be reduced significantly anymore.

Overall, the advantages of our cascaded training can be summarized as: The vertical and horizontal sub-models could complement each other during the cascaded training. When the features in one direction are not very discriminative for segmentation, the features in another direction could make up for the current direction. For example, if the boundaries are blurred in the vertical direction, they might be sharp in the horizontal direction. In the remaining rounds, the vertical (horizontal) sub-model iteratively feeds the generated crossbar patches using *covering re-sampling strategy* to the horizontal (vertical) sub-model in the same round, until the convergence. This can emphasize the learning on the mis-segmented regions and guarantee the sub-models to complement each other.

As a boosting-like algorithm, both the horizontal and vertical sub-model can perform self-improvement with both *basic sampling patches* and *covering re-sampling patches* as the training samples. Here, the self-improvement means that the performance of sub-model in one direction can be improved along with the increase of rounds. We claimed that the *basic sampling patches* in the current round are sampled at different intervals with the previous rounds, which is equivalent to adding new training data to the sub-model. This might be a major cause of the self-improvement. In addition, if both sub-models in the same round fail on a same pixel, the corresponding mis-segmented region of this pixel in the current round will be definitely enhanced in both vertical and horizontal sub-models in the next round. In this meaning, the region around this misclassified pixel could have a larger chance to be emphasized in the next round compared with the current round, which causes the segmentation model in the next round more cares about the segmentation error on this region. Thus, a better segmentation performance on this region is expected as the similar weight-updating way in AdaBoost.

E. Testing

In the testing phase, for a new coming image, we first extract the crossbar patches for each pixel. Then, we input these extracted patches to the trained vertical and horizontal sub-models in each round, to predict the central pixel of each patch belongs to the tumor region or not. Each sub-model outputs a segmentation result, and the final result

is generated by a majority voting on all obtained results. Formally, assuming T as the number of maximum round, we can obtain $2T$ sub-models as $\mathbf{V}^1, \cdots, \mathbf{V}^T$ and $\mathbf{H}^1, \cdots, \mathbf{H}^T$. The result is determined by these $2T$ sub-models.

IV. Experimental Results

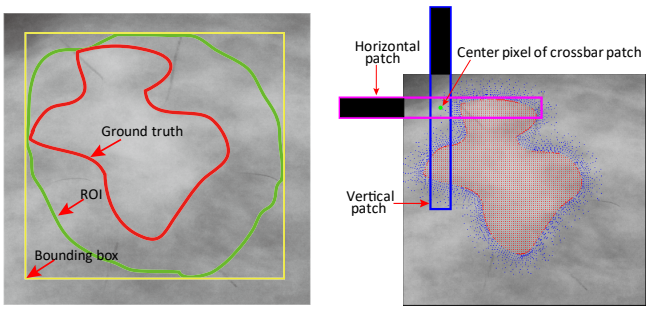
Now we validate the advantages of Crossbar-Net qualitatively and quantitatively. After the introduction of datasets, evaluation criteria, and implementation details, we first investigate the characteristics of Crossbar-Net. Then, we present comparisons between Crossbar-Net and baseline methods on kidney tumor dataset. Finally, we apply Crossbar-Net to the cardiac and breast mass segmentation task to show that our model could be extended to other organ segmentation.

A. Data

Kidney tumor dataset. This dataset is independently collected from Suzhou Science and Technology Town Hospital. 3,500 CT slices of 94 subjects in total are used for performance evaluation, with one tumor per slice. The resolution of the image is 512×512 with 1×1 mm^2/pixel, and the spacing between slices is 1 mm. For each image, the diameter of tumors ranges from 7 pixels to 90 pixels, and the tumor is manually annotated by the physician as the ground truth for training. We randomly partitioned the dataset into three subsets including the training, validation and testing sets which consist of 50, 8 and 36 subjects respectively. The sub-models in the 1-st round is trained using about 580,000 patches being extracted by the *basic sampling strategy*. The epoch number of each sub-model is automatically determined by the validation set, and the performance of each sub-model in each round is still evaluated by the training set.

Breast masses dataset. A subset of DDSM [44], [45] is introduced to investigate the performance of our method. The image in this dataset is LJPEG format and we convert them into PNG format as [46]. There are 1,923 malignant and benign cases in DDSM, and each case includes two images of each breast. The Regions of Interesting (ROI) are given in images containing the suspicious areas. Since the ROIs are not the accurate boundary of a tumor, the boundary of each tumor is annotated again as the ground truth by the experienced radiologists. There are in total 1,000 selected images, among which 600 and 100 images are training and validation set and the remaining 300 images are testing set. In most of the breast mass segmentation methods, the image is cropped to the bounding box of ROI [47], [48] or 1.2 times of the bounding box [49]. We follow the cropping manner in [49]. An example is shown in Fig. 6(a). According to the *basic sampling strategy*, some patches might be extracted outside the image (Fig. 6(b)). For these patches, the parts outside the image are filled with black. When implementing the *covering re-sampling strategy*, the black part of mis-segmented regions would not be re-sampled. Most tumors in these images are smaller than 450 pixels in diameter with very few of them whose diameter is about 600 pixels.

Cardiac dataset. This dataset is available from [50] and comprised of cardiac MRI sequences for 33 subjects with total 7,980 MR slices. The image resolution is 256×256 with the in-plane pixel size as 0.9-1.6 mm^2, and the inter-slice thickness as 6-13 mm. In each image, endocardial and epicardial contours of the left ventricle (LV) are provided as the ground truth. We randomly select 20 and 3 subjects as the training and validation set to train sub-models. The images of the remaining 10 subjects form a testing set for evaluation. The amount of training patches is about 350,000 in the 1-st round, and about 100,000 and 50,000 samples are sampled in the 2-nd and 3-rd round respectively.



(a) (b)

Fig. 6. Example of the cropped breast mass image and the extracted crossbar patch. (a) is the cropped image. (b) shows the crossbar patch extracted outside the image.

B. Evaluation Criteria

We employ the Dice similarity coefficient (DSC) and the true positive fraction (TPF) as the primary evaluation criteria for assessing the segmentation performance. DSC is usually employed to measure the overlap between the prediction and manual segmentation. A large DSC indicates a high segmentation accuracy. TPF indicates the percentage of correctly predicted tumor pixels in the manually segmented region. The higher the TPF is, the larger the coverage of the true tumor region is. We also introduce the centroid distance (CD) and the Hausdorff distance (HD) to evaluate the segmentation accuracy. CD indicates the distance between the central pixels of the final segmentation and manual segmentation which is used to indicate the Euclidean distance between two central points in a 3-D space. Similar to CD, a smaller HD indicates higher proximity between the final segmentation and manual segmentation, which is introduced in quite a bit of detail in Zhang *et al.* [51]. More details about DSC, TPF, and CD are introduced in [52].

C. Implementation Details

For the scale of crossbar patch on all datasets, we set the size of the horizontal patch as 20×100 and vertical patch as 100×20 on kidney data and cardiac data, and 50×500 and 500×50 on DDSM, respectively. We implement our networks on MatConvnet toolbox [53]. In order to improve the credibility of segmentation, the training and testing procedure are repeated 3 times in all experiments. In each time, the training, validation, and testing subsets are selected randomly, and we report the final average performance. Each sub-model

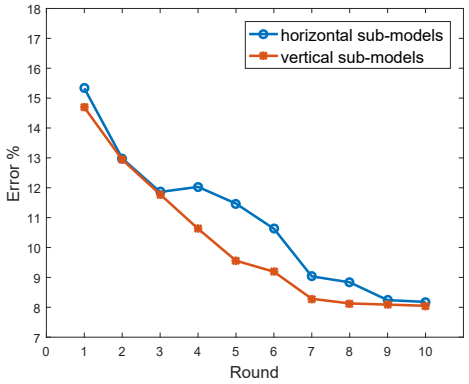


Fig. 7. Illustration of training error of each sub-model. The sub-models are trained separately.

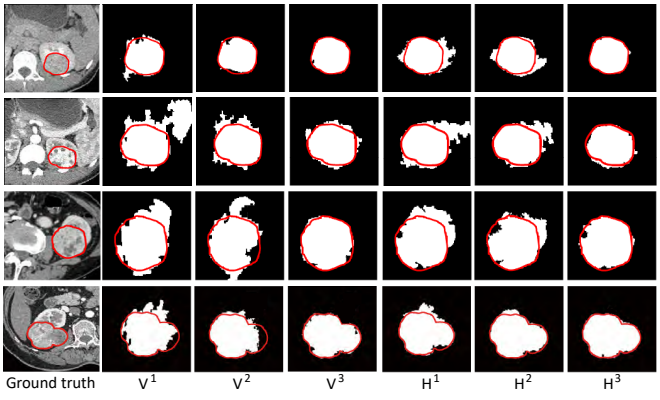
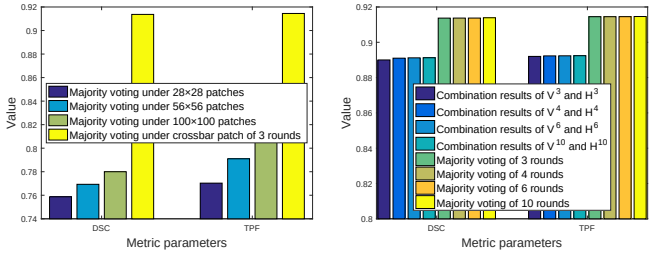


Fig. 8. Performance of each sub-model on kidney tumor. The left column indicates the ground truth image, the second to forth columns indicate the results of three vertical sub-models, and the last three columns indicate the results of three horizontal sub-models, respectively.

reaches convergence within 20, 25 and 25 epoches on kidney tumor, cardiac and DDSM, respectively. We run all deep models on NVIDIA GTX 1080 Ti.

D. Characteristics of Crossbar-Net

We analyze four aspects about the characteristics of the proposed Crossbar-Net, which including: (1) the sub-model can perform self-improvement, (2) the sub-models in the same training iteration can complement and benefit each other,



(a) under different patches (b) under crossbar patches

Fig. 9. DSC and TPF of Crossbar-Net.

TABLE II
AVERAGE METRICS FOR EACH SUB-MODEL ON TESTING SET

	$\mathbf{V}^1$	$\mathbf{V}^2$	$\mathbf{V}^3$	$\mathbf{H}^1$	$\mathbf{H}^2$	$\mathbf{H}^3$
DSC	0.853	0.871	0.882	0.847	0.871	0.881
TPF	0.844	0.876	0.883	0.834	0.869	0.884
HD (mm)	10.200	9.586	9.223	11.315	10.001	9.890
CD (mm)	4.346	3.424	2.790	4.403	3.560	2.909

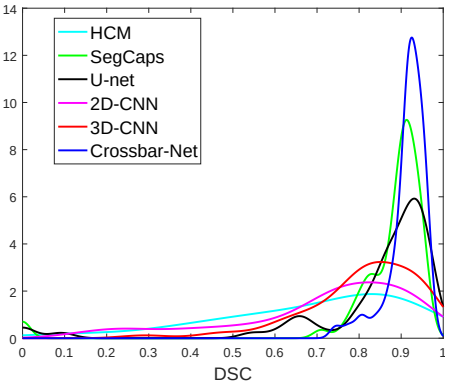
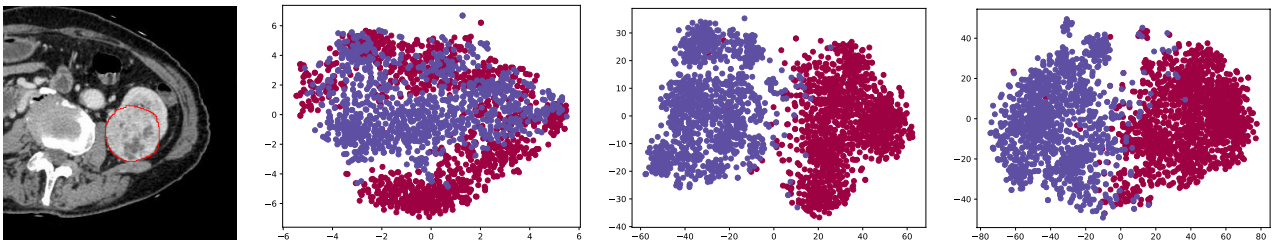


Fig. 11. Distribution of DSC on 600 kidney tumors.

(3) the advantage of combining all the sub-models for final segmentation, and (4) the effectiveness of crossbar patches.

Self-improvement. The purpose of this experiment is to show that if the vertical sub-model can self-improve without the involvement of the horizontal sub-model, and vice versa. We first train the vertical sub-model separately to obtain $\mathbf{V}^1$. Then, we re-sample the mis-segmented regions with the *covering re-sampling strategy*, where the re-sampling patches are the vertical patches instead of the horizontal patches. Then, we train $\mathbf{V}^1$ with these patches together with those gotten from the *basic sampling strategy* to get $\mathbf{V}^2$. Similarly, we can obtain the following $\mathbf{V}^3$, $\mathbf{V}^4$, $\cdots$ in a same way. We repeat training the sub-model (10 times here) until the training segmentation error converges. The horizontal sub-model is excluded throughout the process. The same process is also employed on horizontal sub-model. As illustrated in Fig. 7, the training error of each sub-model decreases with the increase of rounds. This experiment also shows that although each sub-model can self-improve, it takes 10 rounds for vertical (horizontal) sub-model to its convergence. In fact, only 3 rounds are needed if we train the sub-models in our manner shown in Fig. 3 instead of using this separate manner.

Complement between Sub-models. In this experiment, we train sub-models in the manner as illustrated in Fig. 3. After 3 rounds, the training error of the vertical and horizontal sub-models reaches its convergence. Thus, the sub-models in these 3 rounds, which are $\mathbf{V}^1$, $\mathbf{V}^2$, $\mathbf{V}^3$ and $\mathbf{H}^1$, $\mathbf{H}^2$, $\mathbf{H}^3$, will be used in all the following kidney tumor experiments. As shown in Fig. 8, we illustrate a typical case for visualization in the first row. In this case, we can observe $\mathbf{V}^1$ fails to properly segment the upper and lower parts of the tumor from vertical direction while $\mathbf{H}^1$ performs well from the horizontal direction. Similarly, $\mathbf{H}^1$ cannot segment the tumor with its left and right background correctly while $\mathbf{V}^1$ is better. At the same time, the degree of disagreement is reduced



(a) The test image (b) Cluster of squared patches (c) Cluster of vertical patches (d) Cluster of horizontal patches

Fig. 10. t-SNE visualization of the high level representations in Crossbar-Net.

between $\mathbf{V}^2$ and $\mathbf{H}^2$. $\mathbf{V}^3$ and $\mathbf{H}^3$ both achieve the promising results eventually. For other tumors in the remaining rows, in addition to complementary regions, there are more or less common misclassified pixels. For example, in the second tumor, the upper right boundary and the lower right boundary are incorrectly segmented by both sub-models in the first two rounds. In this case, $\mathbf{V}^2$ and $\mathbf{H}^2$ are obviously superior to their former sub-models. Also, $\mathbf{H}^3$ and $\mathbf{V}^3$ achieves the best performance. This observation verifies that sub-models can benefit and complement each other. Basically, in all cases, the performances of later sub-models are superior to the former sub-models.

The Way of Combination. Table II is the corresponding quantitative result of Fig. 8, which confirms that the latter sub-model works better than the previous sub-model. However, we cannot simply combine the results of the two last sub-models. Intuitively, the latter sub model pays more attention to the difficult regions (*i.e.*, hard to be segmented by the former sub-models). If we directly use the last sub-models as the segmentation model, we might have a strong attention-bias on the original data. Therefore, the majority voting of all sub-models is adopted, in which the weights of the last sub-models are greater than that of others. Experimentally, we indeed find the effectiveness of the majority voting. As shown in Fig. 9(b), the DSC and TPF of $\mathbf{V}^3$ and $\mathbf{H}^3$ combination are lower than the majority voting result of all sub-models in the 3 rounds around 2%. Theoretically, our *covering re-sampling strategy* is to some extent to being similar to the sampling strategy in AdaBoost. Also, in AdaBoost, the final strong ensemble classifier is obtained by a combination of all the previous weak classifiers since the latter weak classifier also assigns the higher weight to the difficult samples. We empirically set the weights of the last two models ($\mathbf{V}^3$ and $\mathbf{H}^3$) to 1.5 and the remaining to 1.

Although the training error of both vertical and horizontal sub-models converges after 3 rounds, to further show the benefit of combining all the sub-models with the majority voting, we continue the training process until reaching 10 rounds. The DSC and TPF of the 4, 6 and 10 rounds are listed in Fig. 9(b) together with that of the 3 rounds. Obviously, as the number of training rounds increasing, the majority voting still remains superior to the simple combination of the last two sub-models.

Effectiveness of Crossbar Patches. We illustrate the advantages of crossbar patch by additionally comparing the crossbar patch with the squared patch (*i.e.*, 28×28 and 56×56). For the experiment setting about these two types of patches, both their training strategy and network structure are maintained to be consistent for a fair comparison. For the training strategy, both of them adopt the cascaded training to make the network more focus on the mis-segmented regions, and their final segmentation models are the corresponding combination of their respective sub-models. For the network structure, it is impossible to make their networks be identical due to the different input sizes. However, for fair comparison, we tried our best to make their network structures to be almost same with only the difference on $1-3$ convolutional layers. Specifically, the structure in each path of the 28×28 and 56×56 patch is CCCPCCCCCS and CCPCCPCCCCCS respectively. Here C denotes convolutional layer, P is pooling layer and S is softmax layer. Since the filter is set to be rectangular for the rectangular patch, it is natural to set the filter to be square if the patch is square. Therefore, for the kernel size in squared patches, the 3×3 filter is adopted in all convolutional layers of all sub-models. The training data of both vertical and horizontal sub-models are sampled on the same images, while sampling intervals are different.

The results are shown in Fig. 9(a). The DSC and TPF of the model using squared patches are much lower than that using crossbar patches. In order to verify the unappealing result is not caused by the small size of the input patch, we turn the size of patches into 100×100. The structure of sub-model in each path under 100×100 patches is CCPCCPCCPCCCCS. The results are shown in Fig. 9(a), without any significant improvement. This observation indicates that, compared with non-squared crossbar patch, the large squared patches may include more information, while they might also bring in irrelevant noise to distinguish boundary.

Furthermore, in order to highlight the effectiveness of the non-squared patch, we also compare the features learned from crossbar patch and squared patch. We use t-SNE (t-distributed Stochastic Neighbour Embedding) [54] to evaluate the features. We take the 500-dimensional features in $\mathbf{V}^3$ from the vertical patch and the 56×56 patch, in $\mathbf{H}^3$ from the horizontal patch, respectively. As shown in Fig. 10, each point represents a patch projected from 500 dimensions into two dimensions, where the purple one is tumor case and the red one is non-tumor. The positive and negative cases represented by squared patch features are almost indivisible in Fig. 10(b), while the cases in Fig. 10(c) and Fig. 10(d) could be separated

well, which are represented by our vertical and horizontal patches.

E. Comparison to Other Methods on Kidney Tumor

We extensively compare Crossbar-Net with the low-level methods of kidney tumor segmentation, the multi-scale 2D-CNN model with squared patches, the 3D patch-based CNN model and the image-based CNN models.

As mentioned in Section II, the low-level feature methods [1], [2], [7]–[9] have their own specific goals and are difficult to be applied directly here, so we apply their basic operations to our data set: first extracting the whole kidney area with tumors being included firstly, then calculating features manually, and finally classifying or segmenting tumors with non-CNN methods. These methods are termed as hand-crafted based methods (**HCM**). Besides, our Crossbar-Net is essentially a patch-based multi-scale CNN model with local and contextual information being considered. Hence, the second and third compared methods are the multi-scale **2D-CNN** and multi-scale **3D-CNN**. The 2D-CNN model was modified from [15], in which all parameters are kept except for the nodes of the output layer are changed from 9 to 2. The basic 3D-CNN in [55] is adopted as the 3D model, in which the input 3D patches are $50 \times 50 \times 5$ and $100 \times 100 \times 10$ and the size of the convolutional kernels is $3 \times 3 \times 3$. The patches are extracted with *basic sampling strategy* in 2D-CNN and 3D-CNN models.

As for the image-based CNN methods, we investigate four models which are representative in medical image segmentation: the **FCN-G** [21], the **U-Net** [19], the **V-Net** [20] and the **SegCaps** [22]. The FCN-G is a combination of FCN model and graph model, and the later is a post-process of the former segmentation results. Similar to the original literature [21], we also adopt VGGnet [56] as basic architecture of the FCN and apply transfer learning on it. The U-Net, a popular image-based CNN model, is a promising model in medical image segmentation. We implement it with Python 3.6.8 [57] and PyTorch 0.4.0 [58]. The best test results are obtained after 28 epoches in this model. The V-net is another typical image-based model, which is also a 3D segmentation model. The SegCaps is also a representative segmentation model and we get the code from Github [59]. We implemented this model with Tensorflow 1.11.0 [60] on 4 GPUs. The segcapsr3 is chosen as network and other parameters are kept to be consistent with the original code. To adjust our dataset to this framework, we modified the code that is relevant to reading and converting images. We fed the original images to U-Net and SegCaps, and fed the cropped images to the FCN-G and V-Net to fit the respective input sizes of the two models.

We sampled testing images from one testing set which consists of 30 subjects with totally 600 images being randomly sampled. To fully observe all these 600 tumors, we depicted the kernel density estimation of DSC of all compared methods in Fig. 11. It can be observed that Crossbar-Net achieves promising DSC (larger than 0.9) on most tumors. Many low DSC distributed in the multi-scale 2D-CNN and the HCM. The U-Net performs better than these two methods and multi-scale 3D-CNN. SegCaps achieves the second best performance. There are some cases of 0 DSC in the results of U-Net and SegCaps, which indicates that some tumors are missed by these two models.

TABLE III
COMPARISON AMONG DIFFERENT METHODS ON KIDNEY TUMORS.

	DSC	TPF	HD (mm)	CD (mm)
HCM	0.686	0.788	25.991	11.231
FCN-G [21]	0.736	0.752	20.153	9.372
U-Net [19]	0.838	0.832	13.1	4.510
V-Net [20]	0.887	0.891	10.341	3.895
SegCaps [22]	0.879	0.882	10.451	4.132
2D-CNN [15]	0.718	0.709	20.982	9.853
3D-CNN [55]	0.812	0.820	14.225	6.039
Crossbar-Net	**0.913**	**0.915**	**8.891**	**2.624**

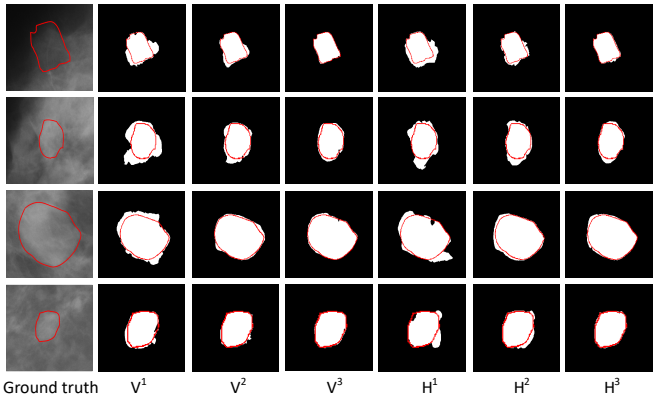


Fig. 13. Performance of each sub-model on DDSM. The left column is ground truth image, the second to forth columns are three vertical sub-models, and the last three columns are horizontal sub-models, respectively.

In Table III, we list average values of DSC, TPF, HD and CD of all test sets of different methods. It is obvious that Crossbar-Net outperforms other methods in terms of the higher DSC and TPF. Moreover, as shown in Table III, it is predominant that Crossbar-Net obtains the smallest value of HD and CD measurements which reflect high quality segmentation. The multi-scale 3D-CNN obtains a higher DSC than the 2D-CNN model since it takes the spatial information into account. Performance of U-Net is slightly better than that of the multi-scale 3D-CNN. In the FCN-G case, the graph model depends on the result of the FCN model [18] and the performance is not very competitive. Although the SegCaps is a 2D model, it performs competitively with V-Net, both significantly superior to other methods except to our Crossbar-Net. The FCN-G, U-Net and SegCaps and V-Net are all developed from FCN model. In order to explore the reasons why these methods are not very effective, we have also applied FCN directly in our task. The result is not desirable (the DSC is even < 0.6) which ignores the local details especially on the boundary of the small tumors. This may be the reason of 0 DSC cases occurring in the U-Net and SegCaps in Fig. 11.

As shown in Fig. 12, we illustrate several typical segmentation examples of Crossbar-Net, image-based method and 3D

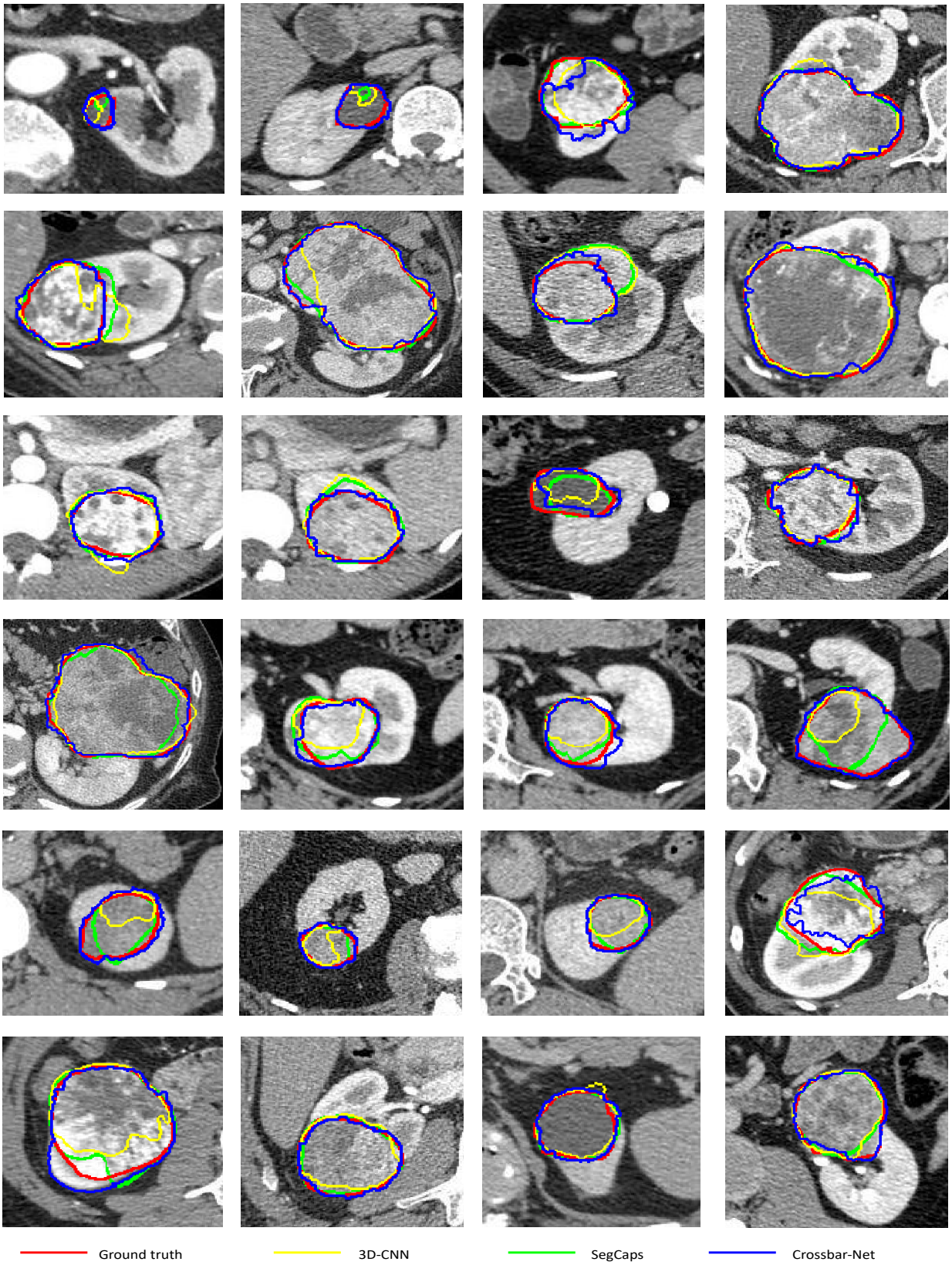


Fig. 12. Examples of segmentation results with ground truth on kidney tumor dataset. The red, blue, yellow and green curves are manual annotation, Crossbar-net, multi-scale 3D-CNN [55], and SegCaps [22] contour, respectively.

TABLE IV
AVERAGE DSC OF EACH SUB-MODEL IN LV SEGMENTATION AND BREAST MASS SEGMENTATION.

	$\mathbf{V}^1$	$\mathbf{V}^2$	$\mathbf{V}^3$	$\mathbf{H}^1$	$\mathbf{H}^2$	$\mathbf{H}^3$
Breast mass	0.853	0.875	0.902	0.849	0.872	0.897
LV	0.875	0.881	0.903	0.869	0.883	0.908

TABLE V
DSC OF EACH METHOD IN BREAST MASS SEGMENTATION.

Method [47]	Cross-sensor [49]	AM-FCN [48]	Method [61]	Crossbar-Net
0.8700	0.9000	0.9130	0.9118	**0.9122**

patch-based model. Obviously, Crossbar-Net segmentation is similar to the ground truth in most cases. SegCaps performs well in big tumors while fails in small cases, even though the tumor has a distinctive texture. The first and second image in the first row of Fig. 12 are the unsatisfactory cases of the SegCaps. The multi-scale 3D-CNN is competitive with SegCaps especially on small tumors, which may be related to its $3 \times 3 \times 3$ small convolutional kernels.

In addition, we have recorded the computation cost of Crossbar-Net, U-Net, and SegCaps: with the implementation on one GPU, our model (including six sub-models) takes $\sim$ 1h for training and less than 1.5s for segmenting a new patient (about 35 slices). The U-Net is very close to our method in training and testing time. The SegCaps running on 4 GPUs takes about 110 minutes for training one epoch and reaches to convergence after 24 epoches. Thanks to our sampling and boosting-like-training, many correctly segmented patches will not feed into the later rounds, which helps reduce the training patches and training time. Specifically, about 580,000 crossbar patches (*i.e.*, 580,000 vertical patches and 580,000 horizontal patches) are input to the vertical and horizontal sub-model respectively in the 1-st training round, to obtain the corresponding H^1 and V^1. In the 2-nd round, about 150,000 patches are fed to H^1 and V^1 respectively, including $\sim$ 90,000 re-sampling patches and 60,000 basic sampling patches. In the 3-rd round, about 70,000 patches are totally sampled.

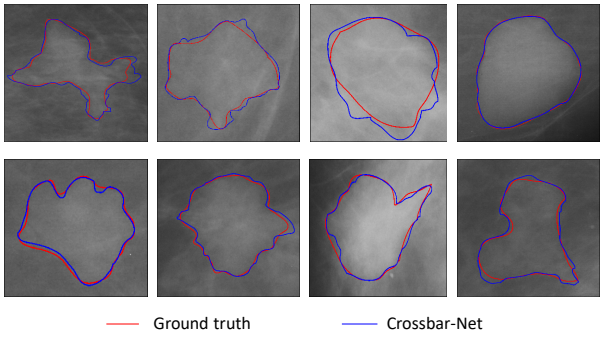


Fig. 14. Typical segmentation results on DDSM dataset.

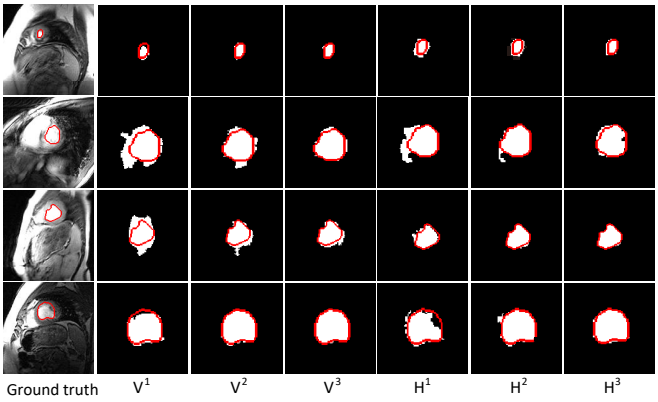


Fig. 15. Performance of each sub-model in LV cave segmentation on the cardiac dataset. The left column is ground truth image, the second to forth columns are three vertical sub-models, and the last three columns are horizontal sub-models, respectively.

TABLE VI
PREDICTION PERFORMANCE COMPARISON IN CARDIAC SEGMENTATION.

Method	LV			MYO	
	DSC	AD (mm)	HD (mm)	DSC	HD (mm)
DMWDP [62]	0.859	2.10	-	-	-
ASAMM [63]	0.856	2.30	-	-	-
DC-FCN [13]	0.915	-	12.08	0.855	14.98
3D-CNN [14]	0.925	-	14.65	0.855	38.12
GridNet [64]	**0.955**	-	5.85	0.885	8.01
Crossbar-Net	0.925	**1.82**	**3.60**	**0.892**	**4.63**

F. Crossbar-Net for Breast Mass Segmentation

We segment the breast mass in mammograph for evaluating the generalization to the related tasks. We illustrate the performance of each sub-models on DDSM in Fig. 13 and Table IV, confirming the characteristics of self-improvement and mutual help again. In Table V, we list the DSC of Crossbar-Net and several state-of-the-art methods which are implemented on DDSM dataset. In this table, we report the best records of [47]–[49], [61] as reported in the original manuscripts. The results in Table V demonstrate that Crossbar-Net is slightly superior to others. As shown in Fig. 6(b), it is possible that the black filled in some crossbar patches (especially the non-tumor patches) contributes to improving the discrimination of the patches. We also show several segmented visualization results of Crossbar-Net (Fig. 14).

Noted that the cost of training and testing on this dataset is larger than that on the kidney and cardiac datasets for the large patch of mammography. This is because that under the 500×50 and 50×500 patch size, the structure of the vertical and horizontal sub-models consists of 11 layers of convolution, 4 layers of pooling and 1 softmax layer, respectively. Also, about 500,000 patches are extracted to train $\mathbf{V}^1$ and $\mathbf{H}^1$. Meanwhile, about 120,000 and 50,000 patches are fed to the sub-models in the 2-nd and 3-rd round, respectively. Thus, regarding the more parameters in the segmentation model compared with the kidney tumor datasets, all six sub-models take about 6h for training and about 15s for segmenting a new subject.

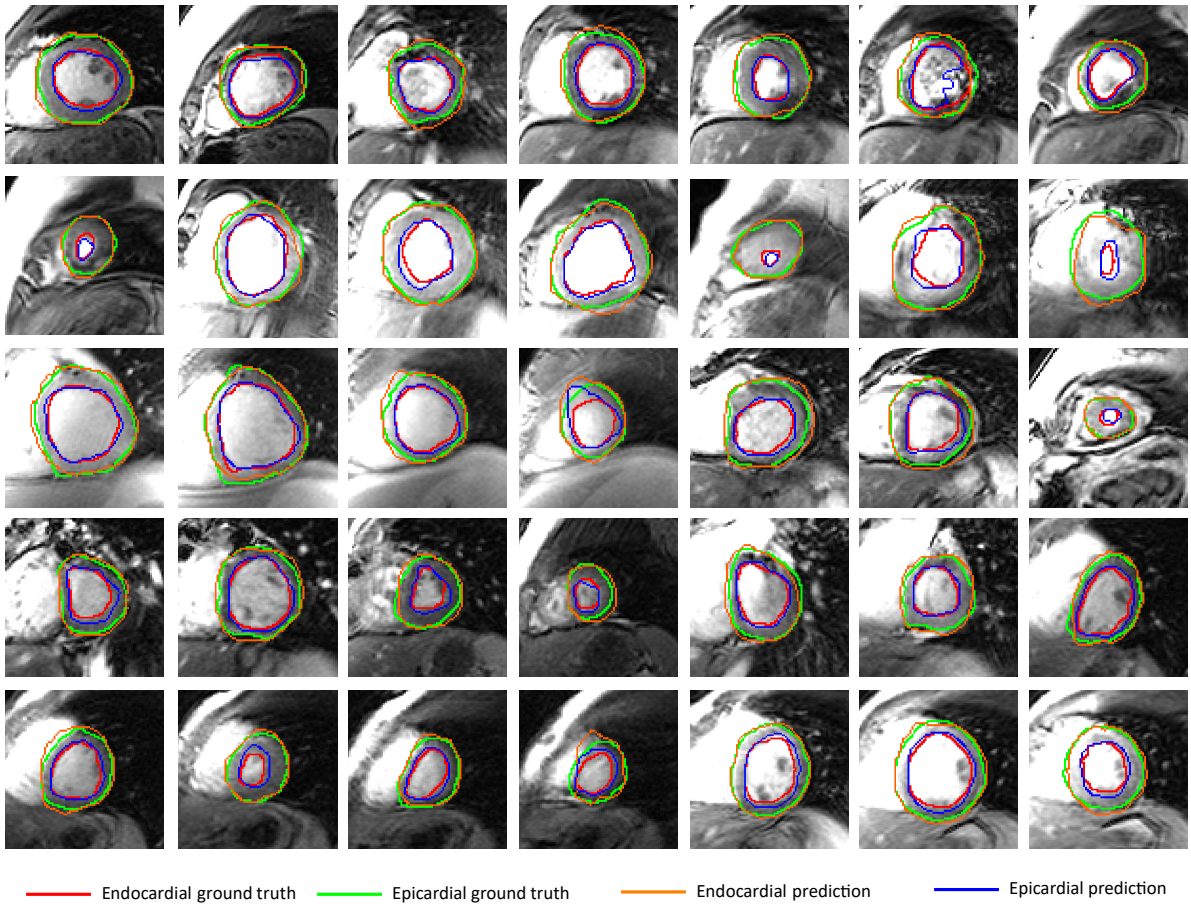


Fig. 16. Examples of resulting segmentations for subjects 24-33 of the cardiac dataset [65].

G. Crossbar-Net for Cardiac Segmentation

In this experiment, the LV and myocardium (MYO) in cardiac MRI dataset are segmented. MYO is determined by endocardia and epicardium together, while LV is determined by the endocardia. The performances of sub-models in LV segmentation are visualized in Fig. 15 and quantified in Table IV. Both the figure and the table indicate a gradual improvement in performance among sub-models in the same direction of different rounds.

Also, we report the DSC and distance metrics of our method and the other five methods proposed in [13], [14], [62]–[64] in Table VI. The deformable model (DMWDP) in [62] and Active Shape and Motion Model (ASAMM) in [63] are non-CNN models, both of which are applied on the same dataset with Crossbar-Net. The remaining three methods are CNN-based models applied on other dataset. All results are directly reported from their original articles for comparison. The average perpendicular distance (**AD**) in Table VI corresponds to the average distance between each pixel in the predicted boundary and the closest ground truth pixel. It is observed that DSC of Crossbar-Net in LV segmentation stands the second best overall result, with GridNet [64] being the most accurate. However, our HD and AD are superior to that of other methods. For the MYO segmentation, our method outperforms all other methods. The highest DSC of MYO and slightly low DSC of LV means that Crossbar-Net is superior to other methods in epicardium segmentation. Therefore, our method is competitive compared with the state-of-the-art methods in cardiac segmentation.

We also show some of the segmented visualization results of our method. As shown in Fig. 16, several representative samples from each sequence of subject 24-33 are illustrated. Note that the performance of Crossbar-Net is better in cardiac segmentation than on kidney tumors because the shape of the LV cavity and the myocardium are more regular.

V. Conclusion

In this paper, we propose a novel segmentation model named as Crossbar-Net, in which the innovations focus on the shape of patches, the way of patch sampling and the style of cascaded training. For the shape of patches, the crossbar patches cover the kidney tumor in both horizontal and vertical directions and capture the local and contextual information simultaneously. For the way of sampling patches, the *basic sampling strategy* and *covering re-sampling strategy* are proposed. The combination of these two strategies not only enhances the role of mis-segmented regions but also prevents sub-models from being over-emphasized on the mis-segmented regions. For the cascaded training style, the segmentation result of sub-models in one direction can be complemented by sub-models in the other direction, and each sub-model can perform self-improvement with re-sampling the

mis-segmented region. Our model can simultaneously learn a variety of information and achieve promising segmentation results on different size, shape, contrast and appearance of kidney tumors. Moreover, the successful application on cardiac and breast mass segmentation shows that Crossbar-Net has a wide range of application. The future work is to extend the direction of symmetric information from horizontal and vertical axes to the other axes.

REFERENCES

[1] M. G. Linguraru, S. Wang, F. Shah, R. Gautam, J. Peterson, W. M. Linehan, and R. M. Summers, "Automated noninvasive classification of renal cancer on multiphase ct," *Medical physics*, vol. 38, no. 10, pp. 5738–5746, 2011.

[2] A. Skalski, J. Jakubowski, and T. Drewniak, "Kidney tumor segmentation and detection on computed tomography data," *Imaging Systems and Techniques (IST), 2016 IEEE International Conference on*, pp. 238–242, 2016.

[3] J. J. Oh, J. K. Lee, B. D. Song, H. Lee, S. Lee, S. Byun, S. E. Lee, and S. K. Hong, "Accurate risk assessment of patients with pathologic t3an0m0 renal cell carcinoma," *Scientific Reports*, vol. 8, no. 1, p. 13914, 2018.

[4] "Cancer facts & figures," *American Cancer Society*, 2018. [Online]. Available: https://www.cancer.org/research/cancer-facts-statistics/all-cancer-facts-figures/cancer-facts-figures-2018.html

[5] D. Xiang, U. Bagci, C. Jin, F. Shi, W. Zhu, J. Yao, M. Sonka, and X. Chen, "Cortexpert: A model-based method for automatic renal cortex segmentation," *Medical Image Analysis*, vol. 42, pp. 257–273, 2017.

[6] D.-Y. Kim and J.-W. Park, "Computer-aided detection of kidney tumor on abdominal computed tomography scans," *Acta radiologica*, vol. 45, no. 7, pp. 791–795, 2004.

[7] H. S. Lee, H. Hong, and J. Kim, "Detection and segmentation of small renal masses in contrast-enhanced ct images using texture and context feature classification," *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on*, pp. 583–586, 2017.

[8] T. Hodgdon, M. D. McInnes, N. Schieda, T. A. Flood, L. Lamb, and R. E. Thornhill, "Can quantitative ct texture analysis be used to differentiate fat-poor renal angiomyolipoma from renal cell carcinoma on unenhanced ct images?" *Radiology*, vol. 276, no. 3, pp. 787–796, 2015.

[9] M. G. Linguraru, S. Wang, F. Shah, R. Gautam, J. Peterson, W. M. Linehan, and R. M. Summers, "Computer-aided renal cancer quantification and classification from contrast-enhanced ct via histograms of curvature-related features," *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pp. 6679–6682, 2009.

[10] Y. Shi, W. Yang, Y. Gao, and D. Shen, "Does manual delineation only provide the side information in ct prostate segmentation?" *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 692–700, 2017.

[11] B. He, D. Xiao, Q. Hu, and F. Jia, "Automatic magnetic resonance image prostate segmentation based on adaptive feature learning probability boosting tree initialization and cnn-asm refinement," *IEEE Access*, vol. 6, pp. 2005–2015, 2018.

[12] A. Mortazi, R. Karim, K. S. Rhode, J. Burt, and U. Bagci, "Cardiacnet: Segmentation of left atrium and proximal pulmonary veins from mri using multi-view cnn," *Medical Image Computing and Computer-assisted Intervention*, pp. 377–385, 2017.

[13] M. Khened, V. Alex, and G. Krishnamurthi, "Densely connected fully convolutional network for short-axis cardiac cine mr image segmentation and heart diagnosis using random forest," *International Workshop on Statistical Atlases and Computational Models of the Heart*, pp. 140–151, 2018.

[14] J. Patravali, S. Jain, and S. Chilamkurthy, "2d-3d fully convolutional neural networks for cardiac mr segmentation," *International Workshop on Statistical Atlases and Computational Models of the Heart*, pp. 130–139, 2018.

[15] P. Moeskops, M. A. Viergever, A. M. Mendrik, L. S. de Vries, M. J. Benders, and I. Išgum, "Automatic segmentation of mr brain images with a convolutional neural network," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1252–1261, 2016.

[16] G. Wang, W. Li, M. A. Zuluaga, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin *et al.*, "Interactive medical image segmentation using deep learning with image-specific fine tuning," *IEEE Transactions on Medical Imaging*, vol. 37, no. 7, pp. 1562–1573, 2018.

[17] R. Zhang, L. Zhao, W. Lou, J. Abrigo, V. Mok, W. C. Chu, D. Wang, and L. Shi, "Automatic segmentation of acute ischemic stroke from dwi using 3d fully convolutional densenets," *IEEE Transactions on Medical Imaging*, vol. 37, no. 9, pp. 2149–2160, 2018.

[18] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015.

[20] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," *3D Vision (3DV), 2016 Fourth International Conference on*, pp. 565–571, 2016.

[21] L. Zhang, M. Sonka, L. Lu, R. M. Summers, and J. Yao, "Combining fully convolutional networks and graph-based approach for automated segmentation of cervical cell nuclei," *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on*, pp. 406–409, 2017.

[22] R. LaLonde and U. Bagci, "Capsules for object segmentation," *arXiv preprint arXiv:1804.04241*, 2018.

[23] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," *Advances in neural information processing systems (NIPS2012)*, pp. 2843–2851, 2012.

[24] S. Wang, M. Zhou, Z. Liu, Z. Liu, D. Gu, Y. Zang, D. Dong, O. Gevaert, and J. Tian, "Central focused convolutional neural networks: Developing a data-driven model for lung nodule segmentation," *Medical image analysis*, vol. 40, no. 40, pp. 172–183, 2017.

[25] G. Tsai, "Histogram of oriented gradients," *University of Michigan*, vol. 1, no. 1, pp. 1–17, 2010.

[26] X. Tang, "Texture information in run-length matrices," *IEEE transactions on image processing*, vol. 7, no. 11, pp. 1602–1609, 1998.

[27] A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen, "Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network," *International conference on medical image computing and computer-assisted intervention*, pp. 246–253, 2013.

[28] K. He, X. Cao, Y. Shi, D. Nie, Y. Gao, and D. Shen, "Pelvic organ segmentation using distinctive curve guided fully convolutional networks," *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 585–595, 2019.

[29] E. Walach and L. Wolf, "Learning to count with cnn boosting," *European Conference on Computer Vision*, pp. 660–676, 2016.

[30] N. Karianakis, T. J. Fuchs, and S. Soatto, "Boosting convolutional features for robust object proposals," *arXiv preprint arXiv:1503.06350*, 2015.

[31] H. Schwenk and Y. Bengio, "Boosting neural networks," *Neural Computation*, vol. 12, no. 8, pp. 1869–1887, 2000.

[32] S. Shalev-Shwartz, "Selfieboost: A boosting algorithm for deep learning," *arXiv preprint arXiv:1411.3436*, 2014.

[33] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, "Brain tumor segmentation with deep neural networks," *Medical image analysis*, vol. 35, no. Jan, pp. 18–31, 2017.

[34] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 5, p. 1612, 1999.

[35] A. Blum and T. M. Mitchell, "Combining labeled and unlabeled data with co-training," *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100, 1998.

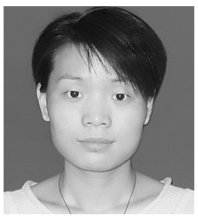
[36] S. Goldman and Y. Zhou, "Enhancing supervised learning with unlabeled data," *Proceedings of the 17th International Conference on Machine Learning*, pp. 327–334, 2000.

[37] I. Muslea, S. Minton, and C. A. Knoblock, "Selective sampling with redundant views," *national conference on artificial intelligence*, pp. 621–626, 2000.

[38] Z. Zhou, K. Chen, and H. Dai, "Enhancing relevance feedback in image retrieval using unlabeled data," *ACM Transactions on Information Systems*, vol. 24, no. 2, pp. 219–244, 2006.

[39] A. Ehsan M. and C. Aron, "Co-training for demographic classification using deep learning from label proportions," *arXiv preprint arXiv:1709.04108*, 2017.

[40] Y. Shi, Y. Gao, S. Liao, D. Zhang, Y. Gao, and D. Shen, "Semi-automatic segmentation of prostate in ct images via coupled feature representation and spatial-constrained transductive lasso," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 11, pp. 2286–2303, 2015.
[41] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.
[42] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.
[43] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," *Neural networks: Tricks of the trade*, pp. 9–48, 2012.
[44] H. Michael, B. Kevin, K. Daniel, M. Richard, and K. W. Philip, "The digital database for screening mammography," *Proceedings of the Fifth International Workshop on Digital Mammography*, pp. 212–218, 2000.
[45] H. Michael, B. Kevin, K. Daniel, K. W. Philip, M. Richard, C. Kyong, and M. S., "Current status of the digital database for screening mammography," *Proceedings of the Fourth International Workshop on Digital Mammography*, pp. 457–460, 1998.
[46] A. Sharma, "Ddsm utility," https://github.com/trane293/DDSMUtility, 2015.
[47] N. Dhungel, G. Carneiro, and A. P. Bradley, "Deep structured learning for mass segmentation from mammograms," *Image Processing (ICIP), 2015 IEEE International Conference on*, pp. 2950–2954, 2015.
[48] W. Zhu, X. Xiang, T. D Tran, G. Hager, and X. Xie, "Adversarial deep structured nets for mass segmentation from mammograms," *IEEE International Symposium on Biomedical Imaging (ISBI2018)*, pp. 847–850, 2018.
[49] J. S. Cardoso, N. Marques, N. Dhungel, G. Carneiro, and A. P. Bradley, "Mass segmentation in mammograms: A cross-sensor comparison of deep and tailored features," *International Conference on Image Processing*, pp. 1737–1741, 2017.
[50] A. Andreopoulos and J. K. Tsotsos, "Efficient and generalizable statistical models of shape and appearance for analysis of cardiac mri," *Medical Image Analysis*, vol. 12, no. 3, pp. 335–357, 2008.
[51] W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, and D. Shen, "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation," *NeuroImage*, vol. 108, pp. 214–224, 2015.
[52] Y. Shi, Y. Gao, S. Liao, D. Zhang, Y. Gao, and D. Shen, "A learning-based ct prostate segmentation method via joint transductive feature selection and regression," *Neurocomputing*, vol. 173, no. 2, pp. 317–331, 2016.
[53] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 689–692, 2015.
[54] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
[55] R. Dey, Z. Lu, and Y. Hong, "Diagnostic classification of lung nodules using 3d neural networks," *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on*, pp. 774–778, 2018.
[56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[57] R. G. Van and F. Drake, "Python 3 reference manual," *Paramount (CA): CreateSpace*, 2009.
[58] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, and Z. DeVito, "Automatic differentiation in pytorch," *31st Conference on Neural Information Processing Systems*, 2017.
[59] "Segcaps," *GitHub*, 2018. [Online]. Available: https://github.com/lalonderodney/SegCaps
[60] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv: Distributed, Parallel, and Cluster Computing*, 2015.
[61] J. Zhang, B. Chen, M. Zhou, H. Lan, and F. Gao, "Photoacoustic image classification and segmentation of breast cancer: a feasibility study," *IEEE Access*, vol. 7, pp. 5457–5466, 2019.
[62] C. Santiago, J. C. Nascimento, and J. S. Marques, "Fast and accurate segmentation of the lv in mr volumes using a deformable model with dynamic programming," *International conference on image processing*, pp. 1747–1751, 2017.
[63] ——, "Combining an active shape and motion models for object segmentation in image sequences," *International conference on image processing*, pp. 3703–3707, 2018.
[64] C. Zotti, Z. Luo, O. Humbert, A. Lalande, and P. Jodoin, "Gridnet with automatic shape prior registration for automatic mri cardiac segmentation," *International Workshop on Statistical Atlases and Computational Models of the Heart*, pp. 73–81, 2018.
[65] C. Santiago, J. Nascimento, and J. Marques, "A new robust active shape model formulation for cardiac mri segmentation," *Image Processing (ICIP), 2016 IEEE International Conference on*, pp. 4112–4115, 2016.



Qian Yu received her Master's Degree from School of Computer Science and Technology, Shandong University, China, in 2009. Currently, she is working toward the PhD degree at Nanjing University, China. Her research interests include computer vision and medical image analysis.

Yinghuan Shi is currently an Associate Professor in the Department of Computer Science and Technology of Nanjing University, China. He received his Ph.D. and B.Sc. degrees from Department of Computer Science of Nanjing University in 2013 and 2007, respectively. He was a visiting scholar in University of North Carolina at Chapel Hill, and University of Technology Sydney, respectively. His research interests include computer vision and medical image analysis. He has published more than 50 research papers in related journals and conferences. He was elected as the Young Elite Scientist by China Association for Science and Technology in 2016, the ACM Rising Star (Nanjing) in 2017, and Science and Technology Award for Youth by Jiangsu Computer Society in 2017.

Jinquan Sun received his B.Sc. degree from School of Computer Science and Technology, Harbing Institute of Technology, China, in 2016. Currently, He is working toward the Master's Degree at Nanjing University, China.

Yang Gao received the Ph.D. degree from the Department of Computer Science and Technology, Nanjing University, China, in 2000. Currently, he is a Professor, and also the Deputy Director in the Department of Computer Science and Technology, Nanjing University. He is currently directing the Reasoning and Learning Research Group in Nanjing University. He has published more than 100 papers in top-tired conferences and journals. His current research interests include artificial intelligence and machine learning. He also serves as Program Chair and Area Chair for many international conferences.

Jianbing Zhu is a chief physician and the associate director of Radiology Department, Suzhou science and Technology Town Hospital (The Affiliated Suzhou Hospital of Nanjing Medical University), with a practical and research experience more than twenty years, including diagnosis of CT and MRI, and more focus in abdomen imaging, tumor imaging and Medical Image Analysis. He has published more than 30 papers.

Yakang Dai is a professor and the associate director of the Medical Imaging Department, Suzhou Institute of Biomedical Engineering and Technology (SIBET), Chinese Academy of Sciences. His research interests include Medical Image Analysis (MRI, CT, PET, MEG/EEG, *etc.*). He has published more than 40 papers and 30 China innovation patents. In addition, he has developed several open medical image/signal analysis software toolboxes (including MITK, 3DMed, eConnectome, iBEAT, aBEAT etc).