ACNN: a Full Resolution DCNN for Medical Image Segmentation

Xiao-Yun Zhou[1*] and Jian-Qing Zheng[1*] and Peichao Li[1] and Guang-Zhong Yang[1,2]

Abstract—**Deep Convolutional Neural Networks (DCNNs) are used extensively in medical image segmentation and hence 3D navigation for robot-assisted Minimally Invasive Surgeries (MISs). However, current DCNNs usually use down sampling layers for increasing the receptive field and gaining abstract semantic information. These down sampling layers decrease the spatial dimension of feature maps, which can be detrimental to image segmentation. Atrous convolution is an alternative for the down sampling layer. It increases the receptive field whilst maintains the spatial dimension of feature maps. In this paper, a method for effective atrous rate setting is proposed to achieve the largest and fully-covered receptive field with a minimum number of atrous convolutional layers. Furthermore, a new and full resolution DCNN - Atrous Convolutional Neural Network (ACNN), which incorporates cascaded atrous II-blocks, residual learning and Instance Normalization (IN) is proposed. Application results of the proposed ACNN to Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) image segmentation demonstrate that the proposed ACNN can achieve higher segmentation Intersection over Unions (IoUs) than U-Net and Deeplabv3+, but with reduced trainable parameters.**

I. INTRODUCTION

Medical image segmentation which predicts the class, anatomy, or prosthesis of each pixel in an image is important for robot-assisted Minimally Invasive Surgeries (MISs). For example, the segmentation of Right Ventricle (RV) is important for instantiating the intro-operative 3D RV shape for navigating robot-assisted Radio-frequency Cardiac Ablation (RFCA) in 3D [1], [2]. The segmentation of markers in [3] is essential for instantiating the intra-operative 3D stent graft shape at fully-compressed [4], partially-deployed [5] and fully-deployed [6] state for 3D navigation in Fenestrated Endovascular Aortic Repair (FEVAR).

Conventional methods are based on ad hoc, expert-designed feature extractors and classifiers. Recently, the use of Deep Convolutional Neural Networks (DCNNs) has shown promising results for many vision-based tasks including image classification [7], object detection [8], and semantic image segmentation [9]. In DCNN, features are extracted and classified automatically by training multiple non-linear modules [10]. Unlike traditional fully-connected neural networks where each output node is linked to all input nodes, an output node of DCNN only links to regional input nodes, known as the receptive field (the input nodes that an output node sees). Multiple convolutional layers, as shown in Fig. 1a, and down sampling layers, i.e., pooling layers

shown in Fig. 1b, are cascaded to achieve a large receptive field coverage. The use of this kind of DCNN means that the feature map is also down sampled, which can be detrimental to pixel-level tasks, i.e., image segmentation. For medical images with focal lesions, local features with small sizes may be discarded due to the down sampling.

In order to compensate for decreased dimension of feature maps, various techniques have been proposed. For example, deconvolutional layers and non-linear up-sampling are used respectively in Fully Convolutional Neural Network (FCNN) [11] and SegNet [12] to recover the down-sampled feature map to the input image size. An alternative is to use atrous convolution [9], also known as dilated convolution [13], to replace the down sampling layer in traditional DCNNs to increase the receptive field. Atrous convolution inserts zeros between non-zero filter taps to sample the feature map as shown in Fig. 1c. It increases the receptive field with the atrous rate but maintains the spatial dimension of feature maps without increasing the computational complexity. However, applying atrous convolution introduces a high demand on memory usage and the inserted zeros of atrous convolution cause input node or information missing. These challenges have limited the practical usage of atrous convolution, particularly for medical image segmentation.
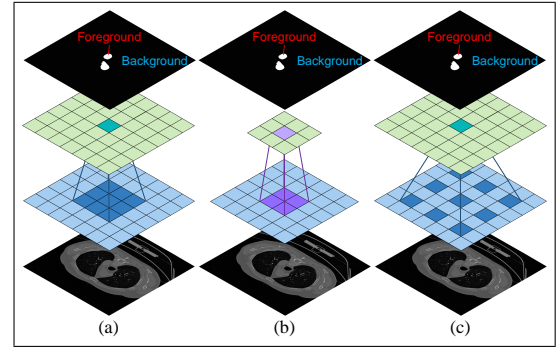

Fig. 1. Illustrations of using DCNN with different receptive fields for medical image segmentation: (a) convolutional layer with a 3×3 receptive field; (b) pooling layer with a 2×2 receptive field; (c) atrous convolutional layer (atrous rate is 2) with a 5×5 receptive field.

As mentioned above, memory shortage is the first challenge for applying atrous convolution, as high-resolution feature map propagation consumes a large amount of memory. In previous work, atrous convolution was usually applied jointly with down sampling layers as a trade-off between the accuracy and memory. For example, in Deeplab [9], atrous convolutional layers were applied on a feature map down-sampled at $\frac{1}{8}$ spatial size of the input image. In multi-scale context aggregation [13], a feature map with 64×64

*Xiao-Yun Zhou and Jian-Qing Zheng contribute equally to this paper

[1]The Hamlyn Centre for Robotic Surgery, Imperial College London, UK. xiaoyun.zhou14@imperial.ac.uk

[2]Institute of Medical Robotics, Shanghai Jiao Tong University, China

dimension was firstly down-sampled from the input image, then a context module with seven atrous convolutional layers was applied. Similar joint usage of atrous convolutional and down sampling layers can also be found in [14].

In practice, setting the atrous rates is another challenge when applying atrous convolution. This is because the output node only links to input nodes which align with non-zero filter taps, as shown in Fig. 1c. The input nodes which align with zero filter taps are not considered. There are thus far no standard ways of setting the atrous rates. For example, an atrous rate setting of (1, 1, 2, 4, 8, 16, 1), representing the atrous rates of seven layers respectively, was allocated for achieving a receptive field of 67×67 in [13] following the strides of max-pooling layers in FCNN. *Wang et al.* found that an atrous rate setting of (2, 4, 8) would cause gridding effects (regular input nodes are missed) and proposed a hybrid atrous rate setting, i.e., (1, 2, 5, 9) to guarantee a coverage of all input nodes [14]. An atrous rate setting of (6, 12, 18) was used for the block and an atrous rate setting of (1, 2, 4) was set inside each block based on empirical knowledge [15].

In this paper, we propose a full resolution DCNN where the spatial dimension of intermediate feature maps remains the same as that of the input image. This is different from the work of [16], for which the spatial dimension of intermediate feature maps at the residual stream is still smaller than that of the input image. For proposing a full resolution DCNN, the proposed network needs to: 1) maximize the receptive field with as few atrous convolutional layers as possible to save memory usage; 2) fully cover the receptive field without missing any input node. In Sec. II, we first prove a method that sets the atrous rate as $(k)^{n-1}$ at the nth atrous convolutional layer, where k is the kernel size and n is the sequence number of atrous convolutional layer, can achieve the largest and fully-covered receptive field with a minimum number of atrous convolutional layers. Due to the truncation effect and for containing more trainable parameters, a full resolution DCNN - Atrous Convolutional Neural Network (ACNN) is proposed by using multiple cascaded atrous II-blocks, residual learning and Instance Normalization (IN). Cardiovascular Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) image segmentation of the RV, Left Ventricle (LV) and aorta are used to validate the proposed ACNN with data collection shown in Sec. II-D and with results shown in Sec. III. U-Net [17] and Deeplab [18] are used as the comparison methods for performance assessment. It has been shown that the proposed ACNN can achieve higher segmentation Intersection over Union (IoU) compared to other techniques with much less trainable parameters and model sizes, indicating the benefit of full resolution feature maps in DCNN. Discussions and conclusions are stated in Sec. IV and Sec. V respectively.

II. METHODOLOGY

A. Atrous Rate Setting

In this section, we focus on optimizing the atrous rate setting which could achieve the largest and fully-covered

receptive field with a minimum number of atrous convolutional layers. Before presenting the detailed mathematical derivation, three 1D receptive field examples with three different atrous rate settings are intuitively shown in Fig. 2. In this three-layer network, with an atrous rate setting of (1, 2, 4), a receptive field of 15 is achieved, while with an atrous rate setting of (1, 2, 9), a receptive field of 25 is achieved with a coverage ratio (the ratio of linked input nodes over all input nodes in the receptive field) of 0.84. With the proposed atrous rate setting of (1, 3, 9), the largest receptive field of 27 is achieved with a full coverage - coverage ratio is 1.0. Detailed mathematical proofs are presented below. For simplification, batch size is fixed at 1 here.
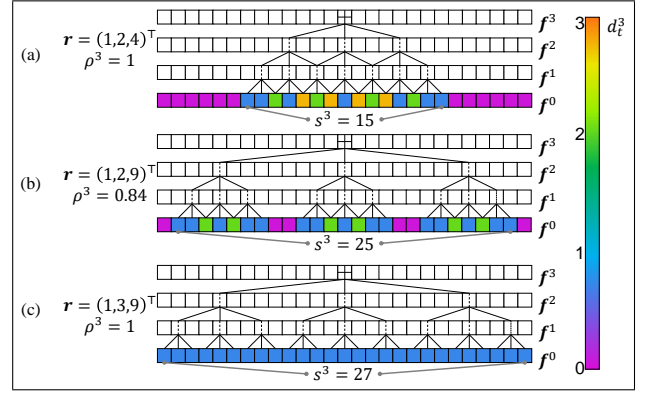

Fig. 2. Three 1D receptive field examples with different atrous rate settings for a three-layer network: (a) an atrous rate setting of (1, 2, 4), (b) an atrous rate setting of (1, 2, 9), (c) an atrous rate setting of (1, 3, 9). The colour represents the link number from the bottom/input node to the top central/output node. ρ^3 is the coverage ratio defined by (7), r is the atrous rate array, s^3 is the receptive field size, $f^{(1 \sim 3)}$ is the 1D feature map, f^0 is the 1D input image, d_t^3 is the receptive field of f_0^3, these notations are explained and used in Sec. II-A.

Denote f and F as 1D and 2D image/feature map. With an input feature map. F^{n-1} of size $H \times W \times c_{n-1}$, an output feature map F^n of size $H \times W \times c_n$ is calculated by the nth atrous convolutional layer with an atrous rate r_n, where $F^0 \in \mathbb{R}^{H \times W \times c_0}$, $F^n \in \mathbb{R}^{H \times W \times c_n}$, $n \in [1, N] \cap \mathbb{N}$, and $r = (r_1 \cdots r_N)^\top \in \mathbb{N}^N$, where $N \in \mathbb{Z}_+$ is the total number of atrous convolutional layers. Here $H \in \mathbb{N}$ is the feature height and $W \in \mathbb{N}$ is the feature width, though these two values are usually equal for medical images. The channel number of feature maps is denoted as $\mathbf{c} = (c_0 \cdots c_N)^\top \in \mathbb{N}^{N+1}$, and F^0 is the input image. By ignoring the non-linear modules, i.e., relu, and the biases, an equivalent 2D atrous convolution could achieve a backward propagation from F^n to F^{n-1}, which can be decomposed into two 1D atrous convolutions [19], with kernel v^n indexed by $t \in \mathbb{Z}$:

$$v_t^n(k, r_n) = \sum_{u=-(k-1)/2}^{(k-1)/2} w_u^n \cdot \mathbf{1}(t - u r_n) \quad (1)$$

Here, k is an odd number which represents the kernel size, i.e., 3, 5, or 7. t is the pixel index. w_u^n, each element of weight matrix $w^n \in \mathbb{R}^k$, is a trainable variable. $\mathbf{1}(t) : \mathbb{Z} \to \{0, 1\}$

is an indicator function defined as:

$$\mathbf{1}(t) := \begin{cases} 1 & t = 0 \\ 0 & t \neq 0 \end{cases} \qquad (2)$$

Denote vectors $\boldsymbol{f}^0, \boldsymbol{f}^n$ as the 1D input image and the nth 1D feature map, both indexed by t. $\boldsymbol{f}^0$ can be calculated from $\boldsymbol{f}^n$ by:

$$\boldsymbol{f}^0 = \boldsymbol{v}^1 * \cdots * \boldsymbol{v}^n * \boldsymbol{f}^n \qquad (3)$$

Define $\boldsymbol{d}^n(k, \boldsymbol{r}) := \boldsymbol{f}^0(\boldsymbol{f}^n = \mathbf{1}(t))$, in which $\boldsymbol{f}^n = \mathbf{1}(t)$ indicates that only the central pixel of $\boldsymbol{f}^n$ is with a non-zero value (=1). It is calculated as:

$$\boldsymbol{d}^n(k, \boldsymbol{r}) := \boldsymbol{v}^1 * \cdots * \boldsymbol{v}^n * \mathbf{1}(t) \qquad (4)$$

By setting $w^n = (1)_k, \forall n$, vectors consisting of 1, then $d_t^n \in \mathbb{N}$, the element indexed by $t \in \mathbb{Z}$, is the link number from f_0^n to the input image's pixel or node. Thus, $\boldsymbol{d}^n$ represents the receptive field of f_0^n, where its receptive field coverage could be represented by the non-zero element number in vector $\boldsymbol{d}^n$:

$$\|\boldsymbol{d}^n\|_0 := \sum_t \left(1 - \mathbf{1}(d_t^n)\right) \qquad (5)$$

and its receptive field size $s^n \in \mathbb{N}$ is calculated as:

$$s^n(k, \boldsymbol{r}) = 1 + (k - 1)\sum_{m=1}^{n} r_m \qquad (6)$$

The receptive field coverage ratio of f_0^n, denoted by $\rho^n \in \mathbb{R}_+$, is then defined as:

$$\rho^n(k, \boldsymbol{r}) := \frac{\|\boldsymbol{d}^n\|_0}{s^n} \qquad (7)$$

In order to ensure a fully-covered receptive field, our target is to maximize the receptive field size with a constraint of receptive field coverage ratio:

$$\max_{\boldsymbol{r} \in \mathbb{N}^{\text{N}}} \left\{ s^{\text{N}} : \rho^{\text{N}} = 1 \right\} \qquad (8)$$

By substituting (6) and (7) into (8), the optimization problem can be converted as:

$$\max_{\boldsymbol{r} \in \mathbb{N}^{\text{N}}} \left\{ \left\|\boldsymbol{d}^{\text{N}}\right\|_0 : \left\|\boldsymbol{d}^{\text{N}}\right\|_0 = 1 + (k-1)\sum_{n=1}^{\text{N}} r_n \right\} \qquad (9)$$

The total link number from f_0^n to $\boldsymbol{f}^0$ is represented by:

$$\|\boldsymbol{d}^n\|_1 = \sum_t d_t^n = (k)^n \qquad (10)$$

where $(k)^{\text{n}}$ represents an exponent calculation. It is the upper bound of $\|\boldsymbol{d}\|_0$ because:

$$\|\boldsymbol{d}\|_0 \leq \|\boldsymbol{d}\|_1, \forall d_t \in \mathbb{N}, \forall t \in \mathbb{Z} \qquad (11)$$

where

$$\|\boldsymbol{d}\|_0 = \|\boldsymbol{d}\|_1 \Leftrightarrow d_t \in \{0, 1\}, \forall t \in \mathbb{Z} \qquad (12)$$

We assume that the (12) holds. By substituting this into the constraint of (9):
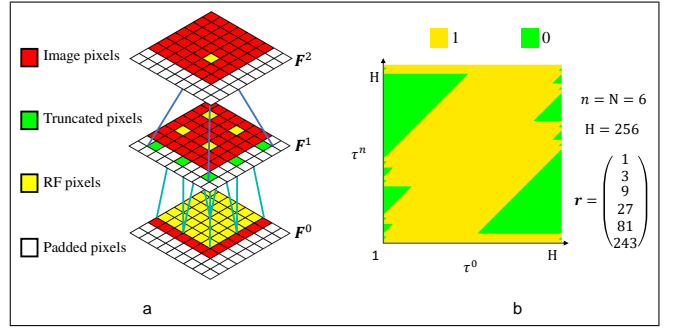
$$1 + (k-1)\sum_{n=1}^{\text{N}} r_n = (k)^{\text{N}} \qquad (13)$$

Fig. 3. Illustration of truncation effect: a. back-propagation from $\boldsymbol{F}^2$ to $\boldsymbol{F}^0$, when $\boldsymbol{F}^1$ is calculated from $\boldsymbol{F}^2$, only yellow pixels at $\boldsymbol{F}^1$ are left, green pixels at $\boldsymbol{F}^1$ are truncated, resulting the red pixels at $\boldsymbol{F}^0$ are no longer covered by the receptive field of the yellow pixel in $\boldsymbol{F}^2$, b. the 1D coverage state of the output node (τ^n) on the input node (τ^0), yellow pixels are covered while green pixels are missed, the total layer number is 6, the 1D image length is 256.

This is a sum of geometric progression; one solution can be obtained as:

$$\boldsymbol{r}' = \begin{pmatrix} 1 & \cdots & (k)^{n-1} & \cdots & (k)^{\text{N}-1} \end{pmatrix}^{\top} \qquad (14)$$

It satisfies a uniformly covered receptive field: $d_t^{\text{N}}(k, \boldsymbol{r}') = \begin{cases} 0 & t \notin \mathbb{S} \\ 1 & t \in \mathbb{S} \end{cases}$, where $\mathbb{S} := [-\frac{s^{\text{N}}-1}{2}, \frac{s^{\text{N}}-1}{2}] \cap \mathbb{Z}$ in 1D and the same in 2D, which satisfies the equivalent condition in (12) and thus is a solution to (9). Therefore, the atrous rate setting of $(k)^{n-1}$ at the nth atrous convolutional layer could lead to the largest and fully-covered receptive field under the condition that the same number of atrous convolutional layers is used.

B. Truncation effect

While the mathematical theory can be solved, the practical implementation of this solution must consider that the feature map has edges. Because the calculation results outside the feature map are not stored, some paths are lost, causing the truncation effect. An intuitive illustration is shown in Fig. 3a. The receptive field of the yellow pixel in $\boldsymbol{F}^2$ should be the red and yellow pixels in $\boldsymbol{F}^0$. However, the red pixels in $\boldsymbol{F}^0$ are not covered due to the truncation of green pixels in $\boldsymbol{F}^1$. Mathematical derivations are stated below.

Denote the two boundary pixels as b and a, where $b \leq 0, a \geq 0$, $a - b = \text{H} - 1$. Thus (4) could be rewritten as (15) where $\varepsilon(t) := \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases}$. Substitute (1) into (15), the general term formula for the 1D path number is in (16). We simulate (16) in $MATLAB^{\copyright}$, an 1D image with length of 256 is put through 6 atrous convolutional layers with an atrous rate of $3^0, 3^1, \cdots, 3^5$ for each, the coverage state of the 256 pixels on the 1D output image is shown in Fig. 3b. We can see that many pixels are missed (shown in green).

a) Link with previous work: traditional DCNNs composed of convolutional layers and down-sampling layers are with Gaussian covered receptive field. The path number for nodes at $\boldsymbol{F}^0$ contribute to $F_{0,0}^{\text{N}}$ shrinks quickly from the central area to the outer area, which is called Gaussian damage [19]. The weights for the outer area nodes grow

$$\boldsymbol{d}^n(b,k,\boldsymbol{r}) := \underbrace{\left(\cdots\left[(\boldsymbol{o}*\boldsymbol{v}^n)\cdot\varepsilon(t-b)\cdot\varepsilon(a-t)\right]\cdots*\boldsymbol{v}^1\right)\cdot\varepsilon(t-b)\cdot\varepsilon(a-t)}_{n\text{-fold atrous convolution with truncation}} \tag{15}$$

$$d_t^n = \sum_{u_n=\max(-(k-1)/2,\lceil b/r_n\rceil)}^{\min((k-1)/2,\lfloor a/r_n\rfloor)}\left\{\cdots\sum_{u_1=\max\left(-(k-1)/2,\left\lceil(b-\sum_{m=2}^n u_m r_m)/r_1\right\rceil\right)}^{\min\left((k-1)/2,\left\lfloor(a-\sum_{m=2}^n u_m r_m)/r_1\right\rfloor\right)}\left\{\mathbf{1}\left(t-\sum_{m=1}^n u_m r_m\right)\right\}\cdots\right\} \tag{16}$$
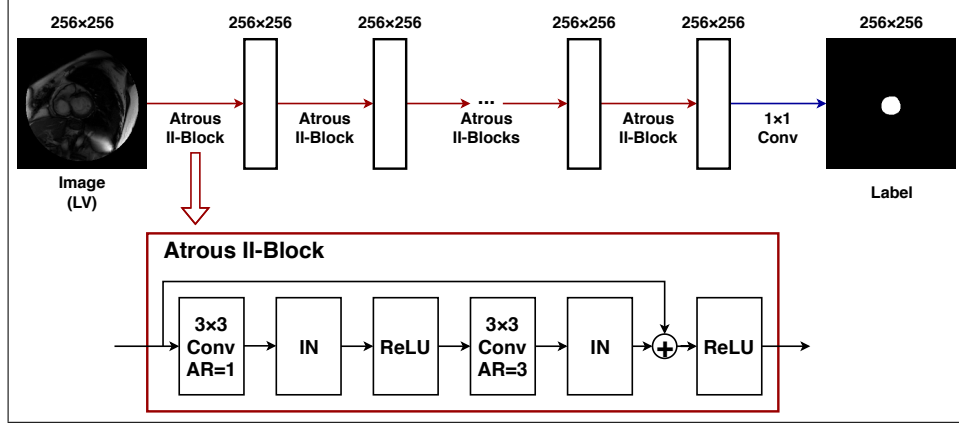

Fig. 4. The network architecture of the proposed ACNN. The number of residual II-blocks is determined by $\frac{(\text{RF}-1)}{8}$, RF is the targeted receptive field. AR - atrous rate, 3×3 Conv - atrous convolution with kernel size of 3, 1×1 Conv1 - atrous convolution with kernel size of 1.

during the training, indicating that outer area nodes are also important. A weight initialization with higher weights at the outer area and lower weights at the central area was tried to compensate this Gaussian damage, however, the improvement is limited and unstable [19]. We propose uniformly covered receptive field which could be a solution for Gaussian damage, but with a different purpose - high-resolution feature map propagation.

C. Atrous Convolutional Neural Network

With the proof in Sec. II-A, a receptive field of $(k)^N$ could be achieved by a block of N atrous convolutional layers. Each node in the receptive field is linked evenly. In this paper, the kernel size of atrous convolutional layers is 3, following the settings used in [20]. A block of N atrous convolutional layers has a receptive field of $(3)^N$. We call this block as atrous block and the one specific with N atrous convolutional layers as N-block, here N is expressed in the roman numeral.

Although a large N indicates larger receptive field, it results in severe truncation effect and less trainable parameters as well. As a trade off, the proposed ACNN is designed into multiple cascaded atrous II-blocks to increase the receptive field linearly by $(3)^2$. For achieving a receptive field of RF, $\frac{\text{RF}-1}{(3)^2}$ blocks are needed. For solving the gradient vanishing/exploding problems and facilitating back propagation, residual learning [21] is added while IN is used as the normalization method, following the review in [22] where IN showed better performance than other normalization method. The final proposed ACNN architecture is shown in Fig. 4. The number of residual II-blocks - $\frac{(\text{RF}-1)}{8}$ is determined by the targeted receptive field.

D. Experimental Setup and Validation

Three cardiovascular MRI and CT datasets for RV, LV and aorta segmentation were used for validation.

a) Right Ventricle (RV): 37 patients, with different levels of Hypertrophic Cardiomyopathy (HCM) were scanned with a 1.5T MRI scanner (Sonata, Siemens, Erlangen, Germany) [1], involving 6082 images with 10mm slice gap, $1.5 \sim 2$mm pixel spacing, $19 \sim 25$ times frames, and 256×256 image size. Analyze (AnalyzeDirect, Inc, Overland Park, KS, USA) was used to label the ground truth. Rotation from $-30°$ to $30°$ with $5°$ as the interval was used to augment the data. Two groups with 18 and 19 patients were split randomly for cross validations.

b) Left Ventricle (LV): 45 patients, from the Sunny-Brook MRI data set [23] were used, it has 805 images with 256×256 image size. Rotation from $-60°$ to $60°$ with $4°$ as the interval was used to augment the data. Two groups, with 22 and 23 patients respectively, were split randomly for cross validations.

c) Aorta: 20 patients, from the VISCERAL data set [24], were used, 4631 CT images with 512×512 image size. Rotation from $-40°$ to $40°$ with $10°$ as the interval was used to augment the data. Two groups with 10 patients for each were split randomly for cross validations.

Image intensities were normalized to $0.0 \sim 1.0$. Evaluation images were not split. For cross validations, one group was used in the training stage while the other group was used in the testing stage. The kernel size of the last

Data	Fold 1 Cross Validation of LV			
Block Number	64	96	128	192
Mean IoU	0.715	0.716	0.726	0.736
Parameter Number	198,928	449,456	599,984	901,040
Training Time	22.1s	33.5s	42.4s	66.8s

atrous convolutional layer is 1 while the kernel size of all the other atrous convolutional layers is 3. The momentum was set as 0.9. Multiple epoch settings, i.e., 1, 2, or 3 and multiple learning rate schedules, i.e., dividing the learning rate by 5 or 10 at the second or third epoch, indicating an optimal learning schedule that: one epoch was trained and the learning rate was divided by 5 and 25 at 2000 and 4000 iterations respectively. Four initial learning rates: 0.1, 0.05, 0.01, 0.005 were trained for each experiment and the highest accuracy was recorded as the final accuracy to avoid non-optimal hyper-parameter settings. For all experiments, Stochastic Gradient Descent (SGD) was utilized as the optimizer.

Pixel-level softmax was applied to transfer the network outputs into probabilities. Cross-entropy was used as the loss function while IoU was used to evaluate the segmentation accuracy. The worker used was *Titan Xp* (12G memory) with the CPU of Intel Xeon(R) CPU E5-1650 v4 @ 3.60GHz 12. The method was implemented with Tensorflow. The process status of the CPU and GPU both influence the training speed. Training all models under exactly the same computer process status is not possible. For a fair speed comparison, the time recorded in this paper is for 100 iterations under the computer process status where all other processes are ended.

III. RESULTS

The receptive field is important for ACNN performance. In Sec. III-A, ACNN with 64, 96, 128 and 192 atrous II-blocks are explored. ACNN-II with a receptive field of double input image size is compared to U-Net [17] and Deeplabv3+ [18] in Sec. III-B. Three segmentation and training curve examples are shown in Sec. III-C and Sec. III-D respectively.

A. ACNN depth

ACNN with 64, 96, 128 and 192 atrous II-blocks were trained to segment the Fold 2 cross validation of LV. The segmentation mean IoUs, number of parameters and training time for 100 iterations are shown in Tab. III-A. We can see that the segmentation performance increases along the block number of atrous II-blocks, and both the parameter number and training time also increase.

B. Comparison to other methods

To trade off between the performance and training time, ACNN with a receptive field of double the input image size (512 for the RV and LV while 1024 for the aorta), named ACNN-II, is used to compare with U-Net and Deeplabv3+.

The mean IoU, parameter number and model size of the three networks are shown Tab. III-B, with validations on the LV, RV and aorta. We can see that the proposed ACNN-II outperforms both the U-Net and Deeplabv3+ in five of the six cross validations. For Fold 2 LV cross validation, ACNN-II achieves very similar mean IoU to the highest one. In addition, the proposed ACNN-II is with much less trainable parameters and model sizes. The performance of Deeplabv3+ is much worse than it was claimed in [18]. This may due to two reasons: 1) it was trained from scratch rather than using pre-trained parameters trained on ImageNet; 2) Deeplabv3+ contains many parameters which causes severe over-fitting in medical image segmentation where the dataset is small.

C. Segmentation examples

Two segmentation examples are selected from each dataset to show the segmentation details in Fig. 6. We can see that ACNN-II achieves noticeably better visual segmentation results than U-Net and DeeplabV3+ with less false positives.

D. Loss curves

One training loss is selected from each dataset to show the loss convergence in Fig. 5. It can be observed that ACNN convergences better and achieves lower loss than U-Net and Deeplabv3+. For the LV and aortic data, the convergence speed is also faster.

IV. DISCUSSION

An atrous rate setting of $(k)^{n-1}$ at the nth atrous convolutional layer, where k is the kernel size, is proposed. It can achieve the largest and fully-covered receptive field with a minimum number of atrous convolutional layers. Comparison experiments with traditional atrous rate settings, i.e., (1, 2, 4, 8, ...), (1, 2, 5, 9, ...) are not conducted due to: 1) smaller receptive field resulted by traditional atrous rate settings would not definitely indicate lower segmentation accuracy, as a large receptive field may be redundant when the target is small; 2) in addition to the receptive field, complex factors, i.e., the link number of each input node and the trainable parameter number influence the segmentation accuracy too. These complex reasons behind a good segmentation result make it difficult to judge the atrous rate setting from the segmentation accuracy. Hence, in this paper, only detailed mathematical proof and derivations are given.

The proposed ACNN achieves a higher segmentation accuracy than U-Net and Deeplabv3+, but uses much less trainable parameters. We think this achievement comes from the efficient information contained in full resolution feature maps. This advantage is very useful when applying the trained model to mobile devices, as less memory is required. The training time and GPU memory consumption is large in ACNN due to the full resolution feature map calculation. Target specific segmentation DCNNs are not compared in this paper, i.e., Omega-Net proposed for cardiac MRI segmentation [25] and Equally-weighted Focal U-Net proposed for marker segmentation [3], as additional algorithms related to the target character is usually applied in these methods and

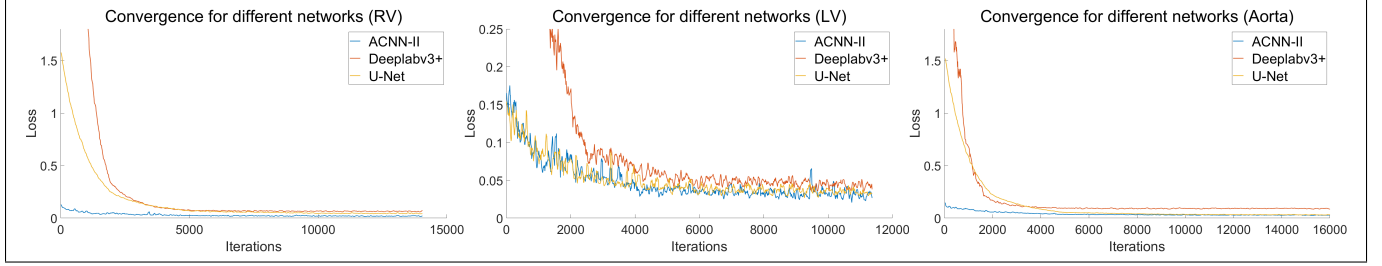| Data		LV		RV		Aorta	
Cross Validation		Fold 1	Fold 2	Fold 1	Fold 2	Fold 1	Fold 2
U-Net	Mean IoU	0.674	**0.722**	0.691	0.733	0.628	0.661
	Parameter Number	7,782,336					
	Model Size	62.3 MB					
Deeplabv3+	Mean IoU	0.523	0.509	0.429	0.489	0.513	0.639
	Parameter Number	36,741,392					
	Model Size	293.9 MB					
ACNN-II	Mean IoU	**0.738**	0.715	**0.697**	**0.735**	**0.655**	**0.684**
	Parameter Number	198,928				599,984	
	Model Size	2.4 MB				4.8 MB	

Fig. 5.   The loss curves of training ACNN-II, U-Net and Deeplabv3+ for the RV, LV and aortic segmentation.
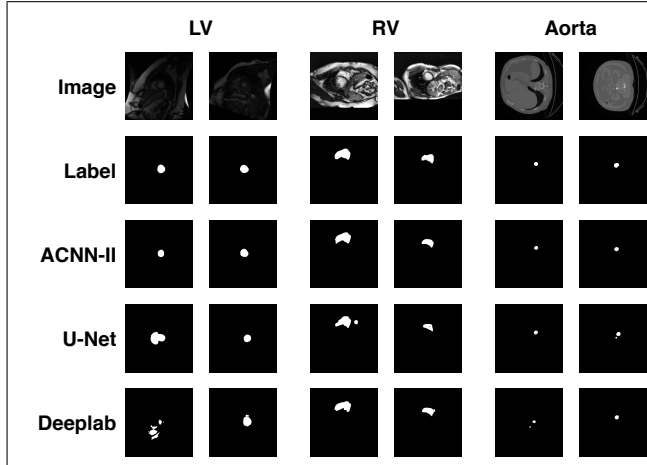

Fig. 6.   Two examples of the image, ground truth, and segmentation results of ACNN-II, U-Net and Deeplabv3+.

hence these methods usually are not generalizable to other datasets.

Due to the fact that the training time increases along the block number of ACNN, up to a block number of 192 is tested due to the time and resource limitation in Sec. III-A and ACNN-II is tested in other sections. In the future, ACNNs with larger block numbers may be further tested. In this paper, ACNN is trained from scratch on three small medical datasets. In the future, the ability of ACNN on large-scale dataset, i.e. PASCAL may be tested, and also its ability for transfer learning.

For a fair comparison, four initial learning rates are explored for each experiment to avoid setting the learning rate less optimally. This process may indicate an optimized

accuracy. However, it would not cause unfairness, as it is the same for all experiments. The shown training time is only for 100 iterations and under a clear computer process status. This time could be longer when the computer and GPU are filled with other processes. It can also be different if the implementation is programmed differently. Training configurations, i.e. the momentum and optimizer are selected based on experience. Different results may exist if different training configurations are utilized.

Based on the author's knowledge, all codes were optimized. Further optimization may exist and may influence the recorded memory usage and training time. The applications of the proposed ACNN are not limited to medical image segmentation, but also could be expanded to other pixel-level tasks, which needs further explorations.

V. CONCLUSION

A new full resolution DCNN - ACNN is proposed for medical image segmentation with the use of cascaded atrous II-blocks, residual learning and IN. A new atrous rate setting is proposed to achieve the largest and fully-covered receptive field with a minimum number of atrous convolutional layers. With much less trainable parameters than that used in the Deeplabv3+ and U-Net, improved accuracy is achieved by even ACNN-II and further improved accuracy can be achieved by deeper ACNN. The derived atrous rate setting contributes to other researches as well. Codes are available at Xiao-Yun Zhou's github.

VI. ACKNOWLEDGEMENT

REFERENCES

[1] X.-Y. Zhou, G.-Z. Yang, and S.-L. Lee, "A real-time and registration-free framework for dynamic shape instantiation," *Medical image analysis*, vol. 44, pp. 86–97, 2018.

[2] X.-Y. Zhou, Z.-Y. Wang, P. Li, J.-Q. Zheng, and G.-Z. Yang, "One-stage shape instantiation from a single 2D image to 3D point cloud," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 30–38.

[3] X.-Y. Zhou, C. Riga, S.-L. Lee, and G.-Z. Yang, "Towards automatic 3D shape instantiation for deployed stent grafts: 2D multiple-class and class-imbalance marker segmentation with equally-weighted focal U-Net," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1261–1267.

[4] X. Zhou, G. Yang, C. Riga, and S. Lee, "Stent graft shape instantiation for fenestrated endovascular aortic repair," in *Proceedings of the The Hamlyn Symposium on Medical Robotics*. The Hamlyn Symposium on Medical Robotics, 2016.

[5] J.-Q. Zheng, X.-Y. Zhou, C. Riga, and G.-Z. Yang, "Real-time 3D shape instantiation for partially deployed stent segments from a single 2D fluoroscopic image in fenestrated endovascular aortic repair," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3703–3710, 2019.

[6] X.-Y. Zhou, J. Lin, C. Riga, G.-Z. Yang, and S.-L. Lee, "Real-time 3-D shape instantiation from single fluoroscopy projection for fenestrated stent graft deployment," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1314–1321, 2018.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2980–2988.

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[12] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 12, pp. 2481–2495, 2017.

[13] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR*, 2016.

[14] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 1451–1460.

[15] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[16] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 3309–3318.

[17] 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, pp. 234–241.

[18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[19] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 4898–4906.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] X.-Y. Zhou and G.-Z. Yang, "Normalization in training U-Net for 2D biomedical semantic segmentation," *IEEE Robotics and Automation Letters*, 2019.

[23] P. Radau, Y. Lu, K. Connelly, G. Paul, A. Dick, and G. Wright, "Evaluation framework for algorithms segmenting short axis cardiac mri," *The MIDAS Journal-Cardiac MR Left Ventricle Segmentation Challenge*, vol. 49, 2009.

[24] O. Jimenez-del Toro, H. Müller, M. Krenn, K. Gruenberg, A. A. Taha, M. Winterstein, I. Eggel, A. Foncubierta-Rodríguez, O. Goksel, A. Jakab *et al.*, "Cloud-based evaluation of anatomical structure segmentation and landmark detection algorithms: Visceral anatomy benchmarks," *IEEE transactions on medical imaging*, vol. 35, no. 11, pp. 2459–2475, 2016.

[25] D. M. Vigneault, W. Xie, C. Y. Ho, D. A. Bluemke, and J. A. Noble, "ω-net (omega-net): Fully automatic, multi-view cardiac mr detection, orientation, and segmentation with deep neural networks," *Medical image analysis*, vol. 48, pp. 95–106, 2018.