 doi:10.3934/ipi.2020049

AUTOMATIC EXTRACTION OF CELL NUCLEI USING DILATED CONVOLUTIONAL NETWORK

RAJENDRA K C KHATRI

Department of Mathematical Sciences, The University of Texas at Dallas
Richardson, TX 75080, USA

BRENDAN J CASERIA

Department of Mathematical Sciences, The University of Texas at Dallas
Richardson TX 75080, USA

YIFEI LOU

Department of Mathematical Sciences, The University of Texas at Dallas
Richardson TX 75080, USA

GUANGHUA XIAO

Quantitative Biomedical Research Center, Department of Population and Data Sciences
University of Texas Southwestern Medical Center
Dallas TX 75390, USA

YAN CAO*

Department of Mathematical Sciences, The University of Texas at Dallas
Richardson TX 75080, USA

ABSTRACT. Pathological examination has been done manually by visual inspection of hematoxylin and eosin (H&E)-stained images. However, this process is labor intensive, prone to large variations, and lacking reproducibility in the diagnosis of a tumor. We aim to develop an automatic workflow to extract different cell nuclei found in cancerous tumors portrayed in digital renderings of the H&E-stained images. For a given image, we propose a semantic pixel-wise segmentation technique using dilated convolutions. The architecture of our dilated convolutional network (DCN) is based on SegNet, a deep convolutional encoder-decoder architecture. For the encoder, all the max pooling layers in the SegNet are removed and the convolutional layers are replaced by dilated convolution layers with increased dilation factors to preserve image resolution. For the decoder, all max unpooling layers are removed and the convolutional layers are replaced by dilated convolution layers with decreased dilation factors to remove gridding artifacts. We show that dilated convolutions are superior in extracting information from textured images. We test our DCN network on both synthetic data sets and a public available data set of H&E-stained images and achieve better results than the state of the art.

2020 *Mathematics Subject Classification.* Primary: 62H30, 62H35; Secondary: 68T10.

Key words and phrases. convolution, dilated convolution,semantic segmentation, deep learning, histopathology image.

This work was supported in part by the National Science Foundations Enriched Doctoral Training Program, DMS grant #1514808.

* Corresponding author: Yan Cao.

1. **Introduction.** When diagnosing and treating patients of cancer, the cellular distribution of tumors is of great interest. Such an analysis can be accomplished using hematoxylin and eosin (H&E)-stained images of cellular tissue extracted via biopsy. Currently, pathological examination has been performed manually by visual inspection of these images. However, this process is labor intensive, prone to large variations, and lacking reproducibility in the diagnosis of a tumor. As a result, it is desired to design automated processes to study different aspects of the cellular distribution. In particular, image segmentation (classifying each pixel as cell nuclei or background) is a crucial first step in enabling the visual study of the disease.

We aim to develop an automatic workflow to detect cell nuclei in cancerous tumors portrayed in digital renderings of the H&E-stained images. Given a H&E-stained image, our goal is to create a binary label describing which pixels are part of cell nuclei (denoted "object" pixels) and which are not (denoted "background" pixels). Convolutional Neural Network (CNN) [10, 8, 18] has been recently demonstrated to be particularly effective in image recognition and classification. Advanced CNN methods such as the Fully Convolutional Neural Network (FCN) [17], U-Net [16] and Segnet [1], which are trained from end-to-end and pixel-to-pixel, have been very successful in semantic segmentation. At the same time, there are also developments in efficient implementations of traditional variational image segmentation models [11, 5].

Both global image context and spatial accuracy are very important in cell nuclei segmentation. In CNNs, multi-scale contextual information is acquired by successive pooling and subsampling layers which reduce the image resolution until a global prediction is obtained [8, 18]. To recover the lost resolution, repeated deconvolutions or unpooling are usually performed to obtain the full-resolution dense prediction [13, 17, 16, 1]. Dilated Convolutions [19, 2] is a promising technique to obtain large context information without the loss of resolution. We design a dilated convolutional neural network for cell nuclei segmentation based on the architecture of SegNet.

Our contributions are three-fold. First, we study the pros and cons of dilated convolutions, and suggest how to use them in applications to improve the performance. We also propose a method to implement dilated convolutions with dilation factor 2^n through efficient conventional convolutions. Second, we design a dilated convolutional neural network for cell nuclei segmentation, which achieves better results than the state of the art. Lastly, the proposed workflow is efficient enough to run on a personal laptop. This work is the first step towards using mathematical models to generate diagnostic inferences and providing clinically actionable knowledge to physicians and patients.

The rest of the paper is organized as follows. We discuss the related work in Section 2. We then detail the proposed approach in Section 3. Practical issues such as color normalization, data augmentation, network training, evaluation metrics are discussed in Section 4. The experiments on synthetic data and real H&E-stained images are conducted in Section 5. Finally, conclusions are given in Section 6.

2. **Related works.** Traditional convolutional networks have fully connected layers which make hard to manage different input sizes. Fully convolutional networks proposed by Long et al. [17] replaced the fully connected layers with convolutional ones to output spatial maps instead of classification scores. Then those maps are upsampled using deconvolutions to produce dense predictions. FCN shows how to train CNNs end-to-end to make dense predictions for semantic segmentation.

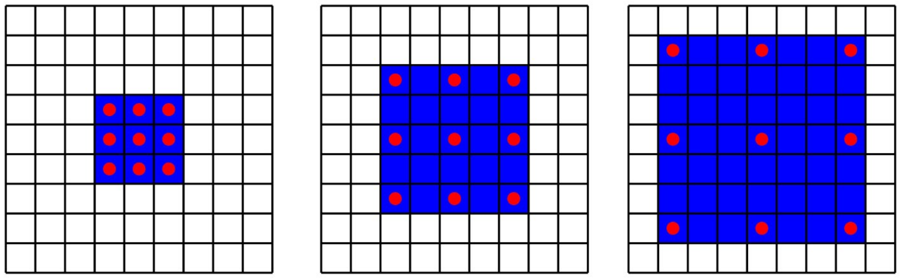

FIGURE 1. 3×3 convolution kernels with different dilation factors 1, 2 and 3 respectively. Red dots indicate nonzero values.

However, the architecture of FCN has a large number of trainable parameters in the encoder network, which makes it hard to train and memory intensive when it comes to testing time.

To overcome the problems caused by deconvolution, Noh et al. [13] proposed a semantic segmentation algorithm (DeconvNet) by learning a deep deconvolution network. Independently, Badrinarayanan et al. [1] proposed SegNet, a deep convolutional encoder-decoder architecture for semantic pixel-wise segmentation. The SegNet consists of an encoder network and a corresponding decoder network, followed by a final pixelwise classification layer. In the encoder network, a mix of convolutional and pooling layers is used to combine and summarize the data obtained from the RGB coordinates. In the decoder network, a mix of upsampling and convolutions is performed to map the low resolution encoder feature maps to the same resolution feature maps as the input for pixel-wise classification. The max pooling indices at the corresponding encoder layer are recalled to perform the non-linear upsampling, which improves boundary delineation and reduces the number of parameters. The last decoder is connected to a softmax classifier to obtain a pixelwise classification map. SegNet is efficient both in terms of memory and computational time during inference.

3. **Proposed method.**

One of the critical components of our design is the dilated convolutional layers [19]. A 2-dimensional dilated convolution can be defined as following:

$$y(m,n) = \sum_{i=1}^{M} \sum_{j=1}^{N} x(m + r \cdot i, n + r \cdot j) w(i,j) \tag{1}$$

where $y(m,n)$ is the output of dilated convolution, $x(m,n)$ is the input, $w(i,j)$ is the filter with size $M \times N$, and r is the dilation factor. If $r = 1$, a dilated convolution turns into a conventional convolution. Figure 1 show 3×3 convolution kernels with different dilation factors.

In dilated convolutions, the alignment of the kernel weights is expanded by the dilation factor. By increasing this factor, the weights are placed more sparse around a pixel, which enlarges the kernel size. Hence, by monotonously increasing the dilation factors through layers, the receptive field (the region in the input space which contributes to feature extraction) can be effectively expanded while preserving the resolution. This means that global information can be exchanged between layers without the loss of resolution. The use of dilated convolutions can cause gridding artifacts [3]. This is because when the dilation factor is greater than 1, the neighboring pixels in the output has non-overlapping receptive fields. The gridding artifacts can be removed using dilated convolution layers with decreased dilation factors [3].

For a given image, we propose a semantic pixel-wise segmentation technique using dilated convolutions. The architecture of our network is based on SegNet, a deep convolutional encoder-decoder architecture. For the encoder, all the max pooling layers in the SegNet are removed and the convolutional layers are replaced by dilated convolution layers with *increased* dilation factors to preserve image resolution. For the decoder, all max unpooling layers are removed and the convolutional layers are replaced by dilated convolution layers with *decreased* dilation factors to remove gridding artifacts. We choose all dilation factors as 2^n to balance the computation load and the size of the receptive fields. The whole process is like enlarging the view gradually first and then gradually focusing on to the point of interest. We remove all max pooling layers because the performance of the networks with and without max pooling layers are similar, yet there is more computational work with max pooling layers. In histopathology images, the percentage of regions occupied by cell nuclei is much less than the background regions. For the training data set we use, the percentage of cell nuclei regions is 23%. To balance the classes, inverse frequency weighting is used in the classification layer [4]. Table 1 shows the detailed network architecture of a SegNet and our dilated convolutional network (DCN).

Our DCN has 34 layers, each convolution layer (except the first and last one) has $3 \times 3 \times 64 \times 64$ weights and 64 bias values. Each batch normalization layer has 128 learnable parameters. Hence the total number of learnable parameters is 335,554. The corresponding SegNet (SegNet with encoding depth 3) has 45 layers, 373,638 learnable parameters. We consider U-Net with 64 output channels for first encoder. U-Net has an architecture that the number of output channels for each convolutional layer in the encoder is doubled going forward and and the number of output channels for each convolutional layer in the decoder is halved. U-Net with encoding depth 3 (U-Net3) has 46 layers, 7,697,410 learnable parameters, U-Net with encoding depth 4 (U-Net4) has 58 layers, 31,031,810 learnable parameters. Compared to U-Net, our DCN which has much less learnable parameters (335,554 vs 7,697,410 for U-Net3) is less likely to overfit the data. Another advantage of our DCN is the efficiency during inference. Our DCN is trained with image patches of fixed size, however, the trained DCN can be applied to images of any sizes, which makes the testing easy and efficient. U-Net has extra links between further-away layers, hence the testing image must have the same size as the training images. To preserve the resolution, our DCN keeps the original size of the input image at each layer, hence the memory usage and computational load are higher than SegNet.

The run time of dilated convolution layers in Matlab deep learning tool box are much longer than the run time of the conventional convolution layers. To overcome this drawback, we design a method to convert dilated convolutions to conventional convolutions. This method works for all dilated convolutions with a dilation factor of 2^n. Assume matrix A is the input matrix, we can reorder the entries of A to get a matrix B, so that in B, all odd rows of A are before the even rows of A, and all odd columns of A are before the even columns of A. We call this procedure "matrix splitting" and the reverse operation "matrix merging". After matrix splitting, a dilated convolution on A with a dilation factor 2 is equivalent to a conventional convolution on B. Figure 2 shows matrix A and its reordering B. It also shows

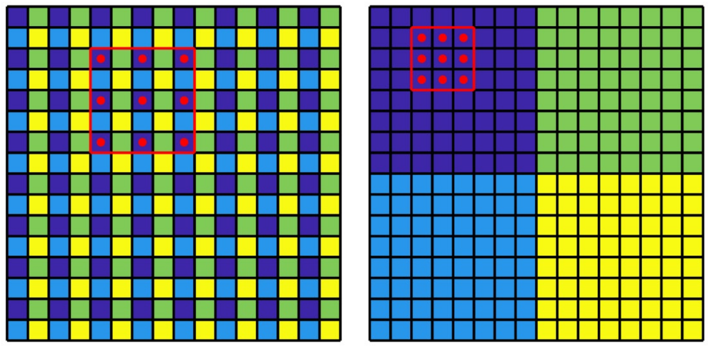

Figure 2. A matrix A and its reordering B. It also shows a dilated convolution on A and its corresponding convolution on B.

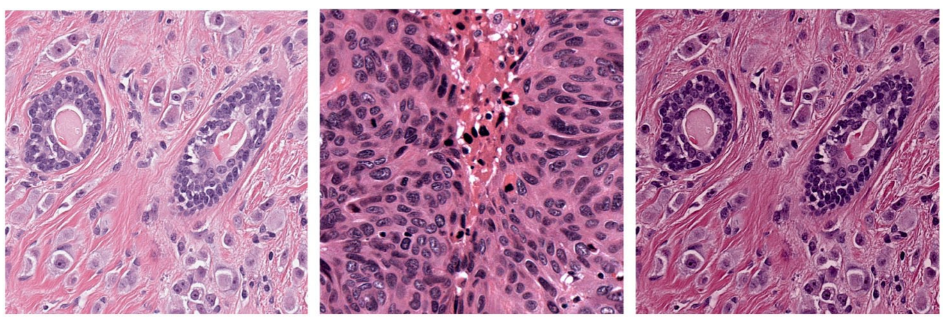

Figure 3. Source image (left), Target image (middle) and Normalized image by Reinhard method (Right).

a dilated convolution on A and its corresponding convolution on B. The third column of Table 1 shows how to use this observation to implement our DCN to train it efficiently. However, the testing time is longer, half of the test time is for matrix splitting and merging. This shows there are rooms of improvements for implementing dilated convolutions in neural network models.

4. **Learning DCN.**

4.1. **Reinhard color normalization.** One of the challenges of nuclei segmentation is the color variation in histopathology images due to the difference in H&E reagents, staining process, and sensor response. Reinhard color normalization [15] is a method of color normalization where the mean and standard deviation of each channel of the image are matched to that of the target by means of a set of linear transforms in Lab colorspace. All images are color normalized by Reinhard method using the Stain Normalisation Toolbox for Matlab [6] before training and testing. Figure 3 shows an example of Reinhard color normalization.

4.2. **Data augmentation.** Data augmentation is a regularization technique which can be used to create artificial training data. It is one of the many popular methods to prevent overfitting of the model and generally we get better performance after data augmentation. During training, reflections about x-axis are randomly performed on training data. In addition, random translations along x-axis and y-axis (within ± 10 pixels) are also performed. Augmentation done this way will not increase the total number of training images at each iteration, which makes the training process more efficient.

4.3. **DCN training.** We implement our method using Matlab deep learning toolbox [12]. Original tissue images and corresponding labeled ground-truth segmentation masks are partitioned into $128 \times 128 \times 3$ non-overlapping image patches. Each patch is normalized to have zero mean. The extracted patches are partitioned randomly into two sets, 80% of the patches are for training and 20% are for validation. To train the network, we try both the stochastic gradient descent with momentum (SGDM) optimizer [14] and the Adam optimization algorithm [7]. The momentum value used for SGDM is 0.9. A fixed learning rate of 0.001 is used. L_2 regularization with a factor of 0.0005 is used for the weights to the loss function to reduce overfitting. The maximum number of epochs for training is set to be 30, and a mini-batch with 4 observations at each iteration was used. The training data and validation data are shuffled before each training epoch. During training, the network is tested against the validation data every epoch. Our method is efficient enough to run on a personal laptop.

4.4. **Evaluation metrics.** We used five metrics to quantitatively measure the segmentation results.

For qualitative analysis, the measures are defined as follows:

- GlobalAccuracy: the percentage of correctly classified pixels among all image pixels, regardless of whether the pixel is a ground truth nuclei pixel or a background pixel.
- MeanAccuracy: the percentage of correctly identified pixels for each class, averaged over the two classes in the image, nuclei or background.
- MeanIoU: Intersection over union (IoU), also known as the Jaccard similarity coefficient. For each class, IoUscore $= \frac{TP}{TP+FP+FN}$, where TP is the number of true positives, i.e. the number of pixels that are ground truth pixels among the detected, FP is the number of false positives, i.e. the number of pixels that are not ground truth pixels among the detected, FN is number of false negatives, i.e. the number of ground truth pixels that have not been detected. MeanIoU is the average IoU score of all classes in the image.
- WeightedIoU: Average IoU of all classes in the image, weighted by the number of pixels in each class.
- MeanBFScore: For each class, BF (Boundary F1) contour matching score is defined by $BF = \frac{2TP}{2TP+FP+FN}$. MeanBFScore is the average BF score of each class in the image.

5. **Experiments.**

5.1. **Test on synthetic data sets.** We generate two data sets of random triangles. Each contains 200 training images and 100 test images of size 64×64. Each triangle in the first data set has an uniform foreground and an uniform background. Each triangle in the second data set has a textured foreground and a textured background. Figure 4 shows some sample training images in the first data set. Figure 5 shows some sample training images in the second data set.

Our DCN is created according to the architecture showed in Table 1, with the input size as 64×64. All networks are trained using the SGDM optimizer with

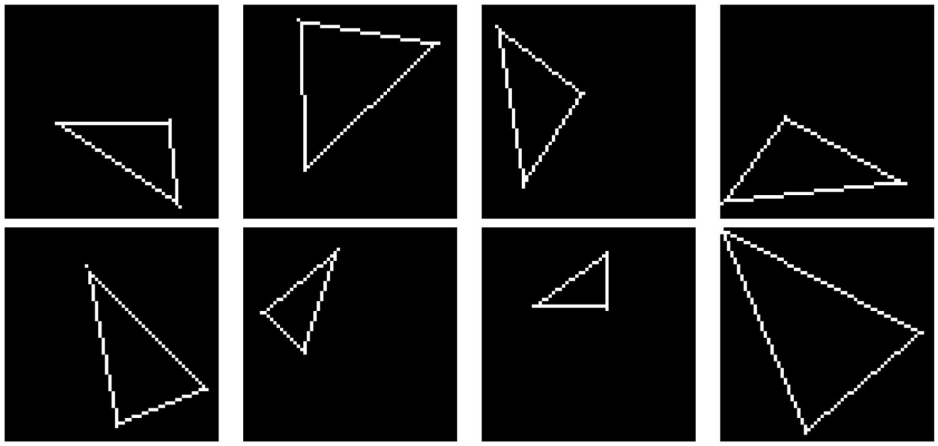

Figure 4. Sample training images in the triangle data set with a uniform foreground and a uniform background.

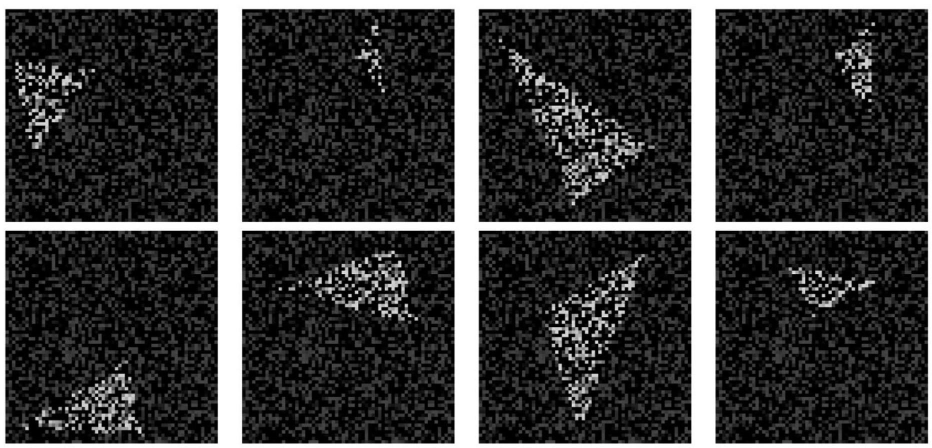

Figure 5. Sample training images in the triangle data set with a textured foreground and a textured background.

the same parameters as described in Section 4.3, except that U-Net3 and U-Net4 are trained with a smaller learning rate of 0.0001. Figure 6 and Figure 7 show the segmentation results on the triangle data sets. Table 2 shows the quantitative metrics of the segmentation results. Here we compare the performance of SegNet, U-Net3, U-Net4 and our DCN. For the triangle data set with an uniform foreground and an uniform background, our method and U-Net4 have similar performance. Both methods achieve better results than SegNet and U-Net3. For the triangle data set with a textured foreground and a textured background, our method is much better than SegNet and U-Net. The main problem of U-Net in this situation is overfitting. The accuracy of training set is very high, however, the performance on the test set is much worse. This is partially because U-Net has a lot more learnable parameters than SegNet and our DCN.

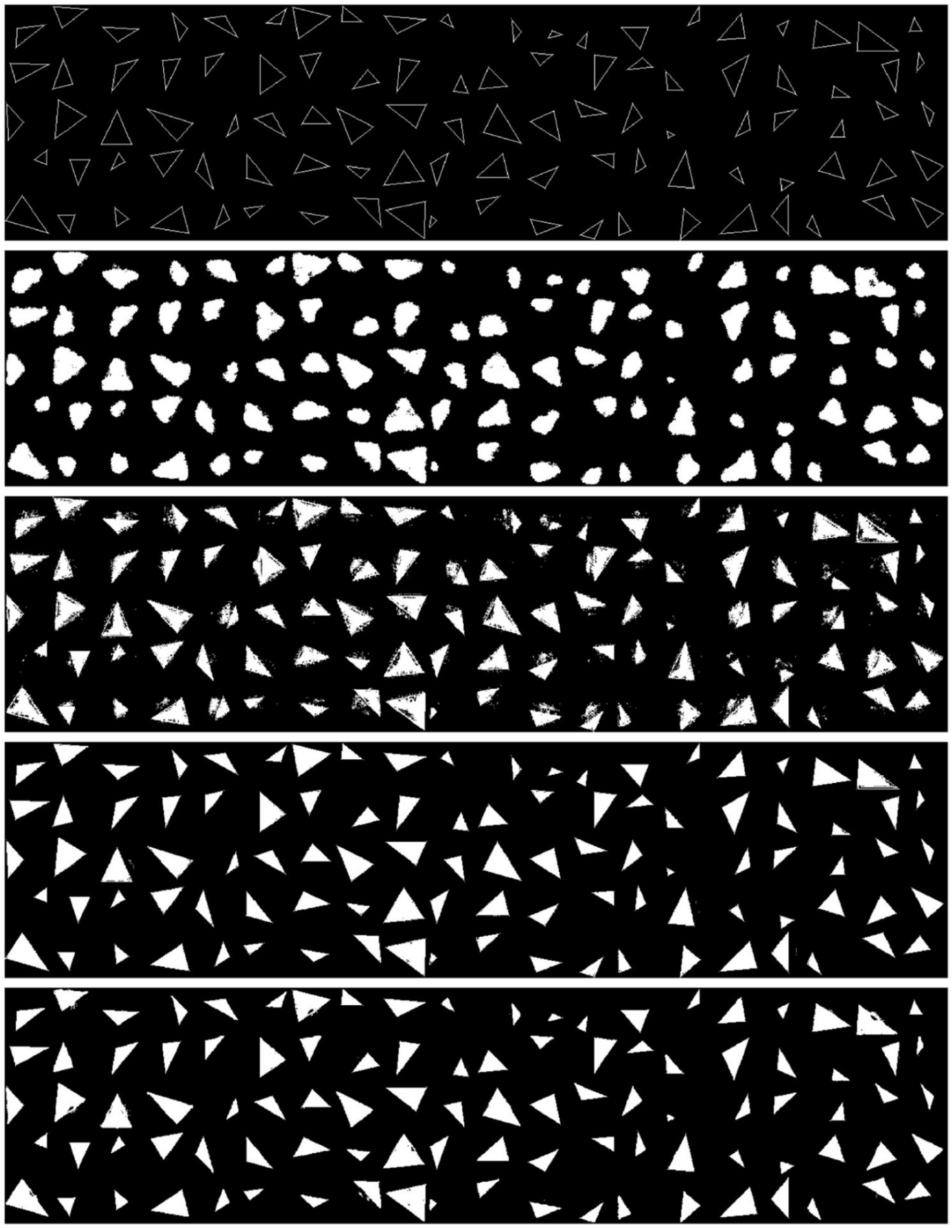

Figure 6. Test images (1st row) with corresponding segmentations using SegNet (2nd row), U-Net3 (3rd row), U-Net4 (4th row) and our dilated convolutional network (5th row).

5.2. **Test on real H&E-stained pathology images.**

5.2.1. *Dataset.* We also test the proposed method on a publicly available dataset of real H&E-stained pathology images [9]. The dataset consists of pathologically labeled images with annotated boundaries of each cell. The data set was generated

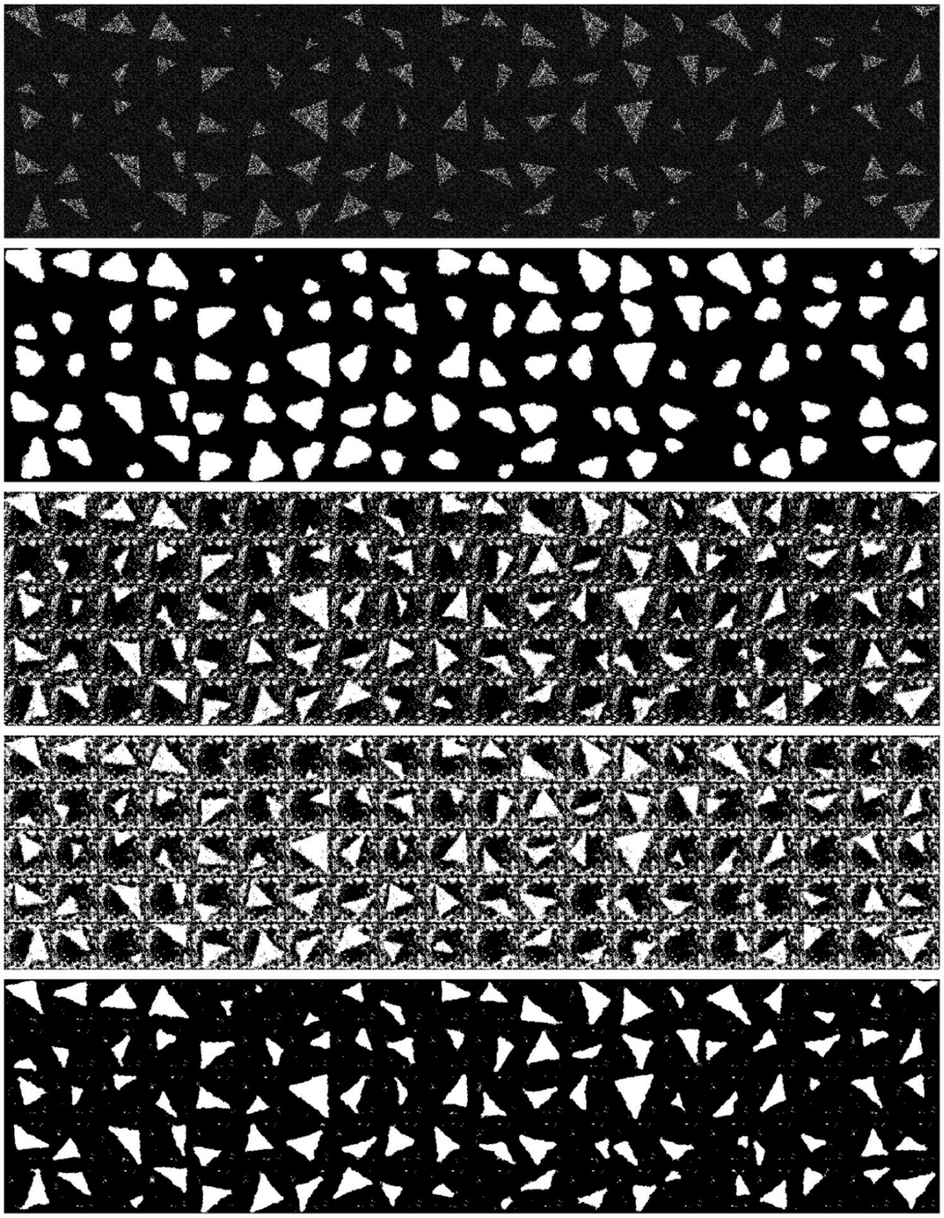

Figure 7. Test images (1st row) with corresponding segmentations using SegNet (2nd row), U-Net3 (3rd row), U-Net4 (4th row) and our dilated convolutional network (5th row).

using the H&E-stained tissue sample images captured at 40x magnification from The Cancer Genomic Atlas (TCGA), one image per patient. This data set contains 6 images of the tissue samples from each of the following types of cancer patients: lung, breast, kidney, prostate. We tested our methods on those images. Training data consists of 16 images (4 lung, 4 prostate, 4 breast & 4 kidney cancer) and test

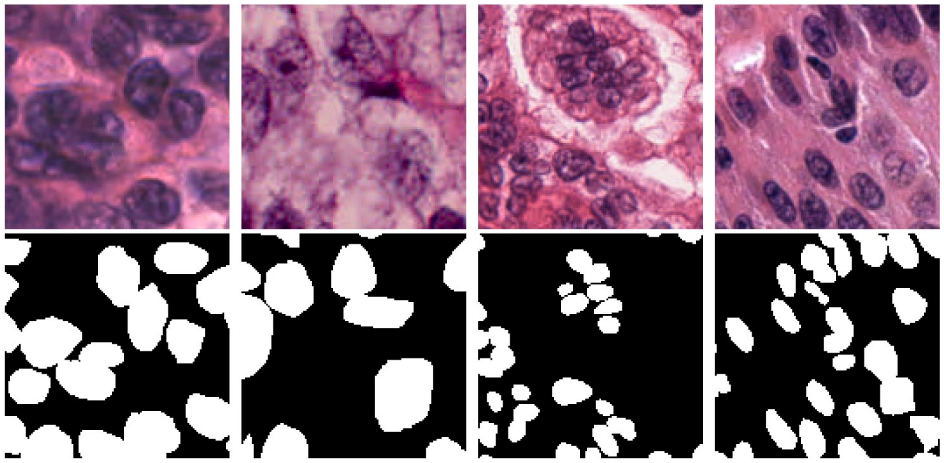

FIGURE 8. Sample normalized image patches and corresponding manual segmentations from the dataset.

data consists of 8 images, 2 from each categories. For the training purpose we have split images into patches of size $128 \times 128 \times 3$. Figure 8 shows several normalized patches and corresponding manual segmentations from the data set.

5.2.2. *DCN testing.* All networks are trained using the Adam optimization algorithm with the same parameters as described in Section 4.3. Figure 9 show the segmentation results on several patches along with the ground truth. Compared to SegNet and U-Net, our method provides better segmentation around the cell boundaries. Table 3 shows the quantitative metrics of the segmentation results. Our method achieves better results than SegNet and U-Net in general. Table 4 shows the training and testing time of SegNet, U-Net3, and our DCN using the H&E-stained image data set. The tests are performed on a Windows 10 personal laptop with 2.8 GHz Intel(R) Core (TM) i7, 16GB RAM, and one GeForce GTX 1050 graph card. Since U-Net cannot handle input images with different sizes, the testing time of U-Net includes time to divide test images into patches and to stitch image patches to the original sizes.

6. **Conclusions.** We developed a cell nuclei segmentation method using dilated convolutional neural network. We also propose a method to implement it efficiently. Experiments show that the proposed method achieves a higher level of accuracy than the state-of-the-art. In particular, the proposed method works better in detecting cell boundaries, shows superior performance on textured images. Future work involves further using the segmentation results to analyze the morphological properties of the tumor cells and predict further progress of the cancer.

REFERENCES

[1] V. Badrinarayanan, A. Kendall and R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.*, **39** (2017), 2481–2495.

[2] L. 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 Trans. Pattern Anal. Mach. Intell.*, **40** (2018), 834–848.

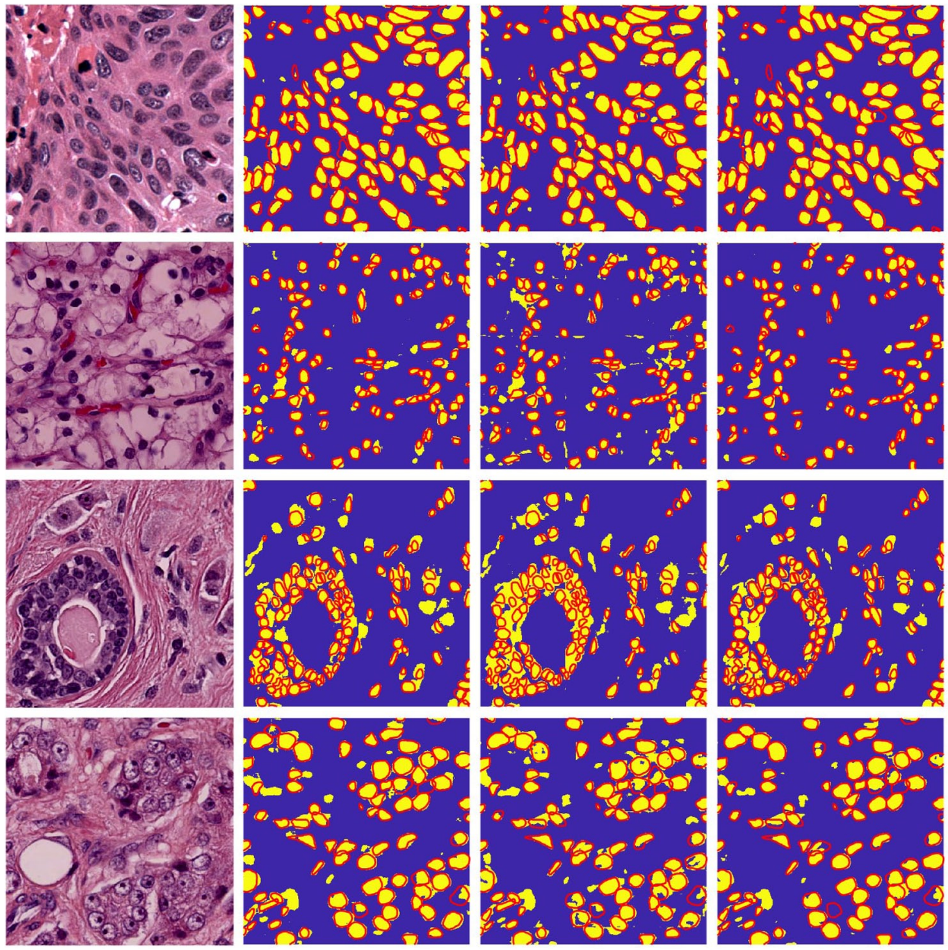

FIGURE 9. Test images (1st column) with corresponding segmentations using SegNet (2nd column), U-Net3 (3rd column) and our dilated convolutional network (4th column). Ground truth contours are plotted in red.

[3] R. Hamaguchi, A. Fujita, K. Nemoto, T. Imaizumi and S. Hikosaka, Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery, *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, URL `http://arxiv.org/abs/1709.00179`.

[4] N. Japkowicz and S. Stephen, The class imbalance problem: A systematic study, *Intelligent Data Analysis*, **6** (2002), 429–449.

[5] M. Jung and M. Kang, Efficient nonsmooth nonconvex optimization for image restoration and segmentation, *Journal of Scientific Computing*, **62** (2015), 336–370.

[6] A. Khan, N. Rajpoot, D. Treanor and D. Magee, A non-linear mapping approach to stain normalisation in digital histopathology images using image-specific colour deconvolution, *IEEE Trans. Biomedical Engineering*, **61** (2014), 1729–1738.

[7] D. P. Kingma and J. L. Ba, Adam: A method for stochastic optimization, in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015, URL `http://arxiv.org/abs/1412.6980`.

[8] A. Krizhevsky, I. Sutskever and G. E. Hinton, Imagenet classification with deep convolutional neural networks, in *Communications of the ACM*, 2017, 1–9, URL `http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`.

[9] N. Kumar, R. Verma, S. Sharma, S. Bhargava, A. Vahadane and A. Sethi, A dataset and technique for generalized nuclear segmentation for computational pathology, *IEEE Trans. Med. Imag.*, **36** (2017), 1550–1560.

[10] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition, in *Proceedings of the IEEE*, **86** (1998), 2278–2324.

[11] C. Liu, M. Ng and T. Zeng, Weighted variational model for selective image segmentation with application to medical images, *Pattern Recognition*, **76** (2018), 367–379.

[12] MATLAB, *version 9.5 (R2018b)*, The MathWorks Inc., Natick, Massachusetts, 2018.

[13] H. Noh, S. Hong and B. Han, Learning deconvolution network for semantic segmentation, in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, IEEE Computer Society, Washington, DC, USA, 2015, 1520–1528.

[14] N. Qian, On the momentum term in gradient descent learning algorithms, *Neural Networks: The Official Journal of the International Neural Network Society*, **12** (1999), 145–151.

[15] E. Reinhard, M. Adhikhmin, B. Gooch and P. Shirley, Color transfer between images, *IEEE Comput. Graph. Appl.*, **21** (2001), 34–41.

[16] O. Ronneberger, P. Fischer and T. Brox, U-net: Convolutional networks for biomedical image segmentation, in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, vol. 9351 of LNCS, Springer, 2015, 234–241, URL `http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a`, (available on `arXiv:1505.04597 [cs.CV]`).

[17] E. Shelhamer, J. Long and T. Darrell, Fully convolutional networks for semantic segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.*, **39** (2017), 640–651.

[18] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015, URL `http://arxiv.org/abs/1409.1556`.

[19] F. Yu and V. Koltun, Multi-scale context aggregation by dilated convolutions, *CoRR*, abs/1511.07122.

Received December 2019; revised April 2020.

E-mail address: rajendraksee@utdallas.edu
E-mail address: Brendan.Caseria@utdallas.edu
E-mail address: Yifei.Lou@utdallas.edu
E-mail address: Guanghua.Xiao@UTSouthwestern.edu
E-mail address: yan.cao@utdallas.edu

	SegNet	Our DCN	Our DCN (for efficient training)
	128x128x3 Input (or 64x64 Input)	128x128x3 Input (or 64x64 Input)	128x128x3 Input (or 64x64) Input
Encoder	64 3x3 Conv Normalization & RELU ReLU 64 3x3 Conv Normalization & RELU Max Pooling 64 3x3 Conv Normalization & RELU 64 3x3 Conv Normalization & RELU Max Pooling 64 3x3 Conv Batch Normalization ReLU 64 3x3 Conv Batch Normalization ReLU Max Pooling	64 3x3 Conv, D=1 Normalization & RELU ReLU 64 3x3 Conv, D=1 Normalization & RELU 64 3x3 Conv, D=2 Normalization & RELU 64 3x3 Conv, D=2 Normalization & RELU 64 3x3 Conv, D=4 Batch Normalization ReLU 64 3x3 Conv, D=4 Batch Normalization ReLU	64 3x3 Conv Normalization & RELU ReLU 64 3x3 Conv Normalization & RELU Matrix Splitting 64 3x3 Conv Normalization & RELU 64 3x3 Conv Normalization & RELU Matrix Splitting 64 3x3 Conv Batch Normalization ReLU 64 3x3 Conv Batch Normalization ReLU
Decoder	Max Unpooling 64 3x3 Conv Normalization & RELU 64 3x3 Conv Normalization & RELU Max Unpooling 64 3x3 Conv Normalization & RELU 64 3x3 Conv Normalization & RELU Max Unpooling 64 3x3 Conv Normalization & RELU 2 3x3 Conv Normalization & RELU	64 3x3 Conv, D=4 Normalization & RELU 64 3x3 Conv, D=2 Normalization & RELU 64 3x3 Conv, D=1 Normalization & RELU 64 3x3 Conv, D=1 Normalization & RELU 2 1x1 Conv	64 3x3 Conv Normalization & RELU Matrix Merging 64 3x3 Conv Normalization & RELU Matrix Merging 64 3x3 Conv Normalization & RELU 64 3x3 Conv Normalization & RELU 2 1x1 Conv
	Softmax Pixel Classification	Softmax Pixel Classification	Softmax Pixel Classification

Table 1. Comparison of the network architectures of the SegNet and our Dilated Convolutional network, where "Conv" means "Convolutions" and D is the dilation factor. Third column shows how to use matrix splitting and merging procedures to implement dilated convolutions through efficient conventional convolutions.

	Triangle Dataset	Global Accuracy	Mean Accuracy	Mean IoU	Weighted IoU	Mean BFScore
SegNet	Uniform	0.9325	0.9508	0.7829	0.8882	0.4172
U-Net3	Uniform	0.9694	0.9531	0.8784	0.9438	0.6572
U-Net4	Uniform	**0.9974**	**0.9953**	**0.9884**	**0.9949**	**0.9488**
Our DCN	Uniform	0.9952	0.9941	0.9786	0.9906	0.8946
SegNet	Textured	0.8764	0.9280	0.6818	0.8139	0.3605
U-Net3	Textured	0.8119	0.8614	0.5855	0.7359	0.2157
U-Net4	Textured	0.7250	0.8148	0.4945	0.6391	0.2000
Our DCN	Textured	**0.9658**	**0.9741**	**0.8728**	**0.9386**	**0.4638**

Table 2. Quantitative metrics of the segmentation results on triangle data sets. Best values are displayed in bold.

	Image Set	Global Accuracy	Mean Accuracy	Mean IoU	Weighted IoU	Mean BFScore
SegNet	Lung	0.8819	0.8975	0.7324	0.8074	0.9204
U-Net3	Lung	0.8917	0.9013	0.7475	0.8190	0.9266
U-Net4	Lung	0.8929	0.8966	0.7484	0.8193	0.9343
Our DCN	Lung	**0.9045**	**0.9033**	**0.7690**	**0.8355**	**0.9448**
SegNet	Breast	0.8691	0.8990	0.7002	0.7917	0.8900
U-Net3	Breast	0.8829	0.9051	0.7183	0.8124	0.8743
U-Net4	Breast	0.8775	0.8977	0.7086	0.8057	0.8700
Our DCN	Breast	**0.9047**	**0.9123**	**0.7538**	**0.8415**	**0.9210**
SegNet	Kidney	0.9122	0.9249	0.7290	0.8634	0.9425
U-Net3	Kidney	0.9133	**0.9281**	0.7259	0.8639	0.9218
U-Net4	Kidney	0.8993	0.9145	0.7013	0.8462	0.9306
Our DCN	Kidney	**0.9329**	0.9277	**0.7725**	**0.8911**	**0.9634**
SegNet	Prostate	0.8956	0.9142	0.7533	0.8271	0.9105
U-Net3	Prostate	0.8949	0.9041	0.7496	0.8255	0.9047
U-Net4	Prostate	0.8961	0.9032	0.7510	0.8271	0.9090
Our DCN	Prostate	**0.9211**	**0.9163**	**0.7962**	**0.8632**	**0.9336**
SegNet	Overall	0.8897	0.9000	0.7383	0.8184	0.9159
U-Net3	Overall	0.8957	0.8976	0.7467	0.8264	0.9069
U-Net4	Overall	0.8914	0.8905	0.7380	0.8201	0.9110
Our DCN	Overall	**0.9158**	**0.9039**	**0.7815**	**0.8548**	**0.9407**

Table 3. Quantitative metrics of the segmentation results on the H&E-stained image data set. Best values are displayed in bold.

	SegNet	U-Net3	Our DCN	Our DCN (efficient)
Training Time	15 min 14 sec	18 min 36 sec	26 min 47 sec	19 min 45 sec
Testing Time	7.6 sec	37.7 sec	10.1 sec	19.9 sec

Table 4. Comparison of the training and testing time of SegNet, U-Net3 and our DCN.