https://doi.org/10.2352/ISSN.2470-1173.2018.10.IMAWM-455

Person Segmentation Using Convolutional Neural Networks With Dilated Convolutions

David Joon Ho ; School of Electrical and Computer Engineering; Purdue University; West Lafayette, Indiana, USA*
Qian Lin; HP Labs, HP Inc; Palo Alto, California, USA

Abstract

Semantic segmentation, classifying each pixel in an image to a set of various objects, is an important and necessary problem to understand images. In recent years, convolutional neural networks trained with public datasets enable to segment objects and understand images. However, it is still challenging to segment objects with high accuracy on a simple and small network. In this work, we describe convolutional neural networks with dilated convolutions to segment person accurately especially near boundary using data augmentation technique. Additionally, we develop a smaller network which can run each frame in webcam video faster without degrading segmentation performance. Our method both numerically and visually outperforms other segmentation techniques.

Introduction

Semantic segmentation, a pixel-wise classification in an image to a set of multiple objects, is a prerequisite step for understanding and analyzing images or videos. Especially, person segmentation can be used in many applications such as video surveillance or activity analysis. Additionally, person segmentation can be used in video conferencing because bandwidth can be saved by only sending pixels belonging to person. However, person segmentation is a challenging task due to various poses and activities.

Several person segmentation techniques have been developed in the past years. An interactive foreground segmentation method by iterative graph cuts, called GrabCut [1], is presented. Although it may produce an accurate person segmentation mask, a user is required to spend some time/effort for interaction. Due to a large number of images, an automatic segmentation method is desired. To automatically segment crowded people [2] introduced a model-based segmentation method where a human shape model is used to segment individual person. Similarly, person segmentation by learning a set of posture clusters and a codebook of local shape distributions in various posture is described in [3]. Alternatively, a level-set technique is used to segment person due to its efficiency and flexibility [4]. Due to the complexity in images, these segmentation methods tend to have poor segmentation performance.

In the recent years, deep learning is widely used to understand images [5]. Training and running convolutional neural networks (CNNs) become possible due to advances in graphics processing units (GPUs) and availability of multiple datasets. CNNs are generally designed with a series of convolutional layers followed by non-linear activation functions. In the first few layers, simple features such as edges or corners in various orientations can be captured. In deeper layers, more sophisticated features can be captured. Deep learning is actively used in image analysis such as image classification [6, 7] and object detection [8, 9, 10].

Deep learning brings a big impact to image segmentation as well. Fully convolutional network (FCN) is described in [11] for semantic segmentation. Instead of classifying each pixel in an image using a patch around it [12], FCN can efficiently segment an arbitrary-sized image. However, FCN could not have an accurate semantic segmentation because details can be lost during a deconvolution layer with a large stride. To produce better segmentation results, an encoder-decoder architecture with a series of deconvolutional layers are introduced such as SegNet [13], DeconvNet [14], and U-Net [15]. Here, an encoder is composed of convolutional layers with pooling layers and a decoder is composed of deconvolutional layers with unpooling layers. Pooling layers generally decrease image resolution so an encoder-decoder architecture still has a limitation for fine segmentation.

An alternative approach uses dilated convolutions, or atrous convolutions, originally introduced to compute wavelet transform more efficiently [16]. Employing convolutional layers with dilated convolutions instead of pooling layers can generate segmentation without losing resolution. A semantic segmentation technique using dilated convolutions, called DilatedNet, is described in [17]. Using dilated convolutions can increase receptive fields exponentially so DilatedNet can provide more fine segmentation. However, dilated convolutions generally require expensive computation.

More recently, other segmentation methods have been developed for accurate results. DeepLab [18] uses a combination of a convolutional neural network with dilated convolutions and fully connected conditional random fields (CRFs) [19] to have accurate segmentation. [20] uses generative adversarial networks (GANs) [21] as a post processing to improve segmentation results. Here, GAN is used so that segmentation results cannot be distinguishable to groundtruth images. Alternatively, pyramid scene parsing network (PSPNet) using pyramid pooling module to cooperate both local and global information in an image is described in [22]. Similarly, a multi-path refinement network (RefineNet) [23] is designed with short-range and long-range residual connections to generate high-resolution feature maps. Alternatively, DeepMask [24] and SharpMask [25] can segment and additionally generate object proposals.

In this paper, we present a compact and accurate method for segmenting person in images or webcam video. The main contributions of our work are: (1) using data augmentation techniques using various public datasets and an HP private dataset (2) fine-tuning DilatedNet [17] with augmented training images for fine segmentation performance (3) training a compact DilatedNet with

*This work was performed while David Joon Ho was an intern at HP.

augmented training images for faster segmentation performance to be more practical in webcam video applications. Our methods are evaluated by measuring the size of models and execution time.

Proposed Method

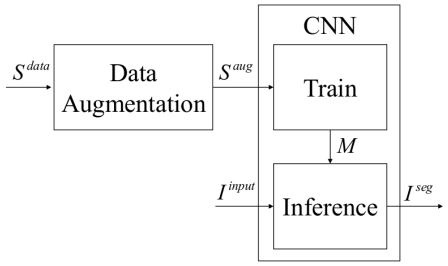

Figure 1. *Block diagram of the proposed method for person segmentation*

Figure 1 is a block diagram of our proposed method for person segmentation. A set of training images, S^{data}, is acquired from various public datasets such as the Microsoft COCO (MSCOCO) dataset [26], the ADE20K dataset [27, 28], and the Cityscapes dataset [29]. We multiply training images using data augmentation methods to improve segmentation performance. We denote the augmented set of training images as S^{aug}. Then S^{aug} is used to train a convolutional neural network model, M. Here, we simplify DilatedNet [17] for faster performance. Once the training process is completed, a segmented image, I^{seg}, is generated from an input image, I^{input}.

Data Augmentation

Data augmentation is a well-known method to avoid overfitting during a training process by increasing the number of training images with simple transformations [6]. In this work, we developed our data augmentation methods to have more accurate segmentation.

The first data augmentation method generates synthetic images by placing person masks on an image. In this data augmentation method, we used an HP private dataset containing person portrait with accurate groundtruth as person masks. We placed a random number of person from n_{min} to n_{max} with a random scaling factor from s_{min} to s_{max} at a random position on background images from the MSCOCO dataset [26]. Here, a background image is defined as an image without person presented. We selected $n_{min} = 3$, $n_{max} = 10$, $s_{min} = 3$, and $s_{max} = 30$. In this paper, augmented images generated from the first data augmentation method are called "augmented images using the HP private dataset as foreground and the MSCOCO dataset as background". Using the first data augmentation method, we generated 56,479 images.

The second data augmentation method uses simple transformations to produce training images. The size of an original image from the Cityscapes dataset [29] is 2048×1024 which is larger than other training images we have. To reduce the size of images, we performed the following data augmentation method: (1) we cropped the upper left corner and the upper right corner of an original image to create two 1024×512 images (2) we downsampled the original image with a factor of 2 to create one 1024×512 image. Note that we did not crop the lower left corner and the lower right corner of an original image because these regions rarely contain person. Figure 2 shows an example of second data augmentation method using the Cityscapes dataset. In this paper, augmented images generated from the second data augmentation method are called "augmented images using the Cityscapes dataset". Using the second data augmentation method, we generated 8,823 images.

(a)

(b) (c) (d)

Figure 2. *An example of the second data augmentation method using the Cityscapes dataset [29] (a) an original image with size of* 2048×1024*, (b) a cropped image from the upper left corner of the original image with size of* 1024×512*, (c) a cropped image from the upper right corner of the original image with size of* 1024×512*, (d) a downsampled image by a factor of 2 from the original image with size of* 1024×512

Dilated Convolutions

Before describing our convolutional neural network, let us define dilated convolutions. First of all, a discrete convolution, $*$, of a signal, f, and a filter, h, is:

$$(f * h)(x) = \sum_{x_1 + x_2 = x} f(x_1)h(x_2) \tag{1}$$

In this work, an l-dilated convolution, $*_l$, is defined as:

$$(f *_l h)(x) = \sum_{x_1 + lx_2 = x} f(x_1)h(x_2) \tag{2}$$

where l is defined as a dilation factor. Note that a discrete convolution in Equation 1 is a 1-dilated convolution.

Using dilated convolutions in convolutional layers can generate accurate segmentation due to a large receptive field. Let f_0 be an original 2D image and h_i be a 3×3 filter on the i-th convolutional layer where $1 \leq i \leq N$. If there is a combination of N convolutional layers with discrete convolutions, $f_i = f_{i-1} * h_i$, then the receptive field in the N-th layer is $(2N+3) \times (2N+3)$. But if there is a combination of N convolutional layers with dilated convolutions, $f_i = f_{i-1} *_{2^i} h_i$, then the receptive field in the N-th layer is $(2^{N+2}-1) \times (2^{N+2}-1)$. Here, the receptive field is linearly increased if discrete convolutions are used in convolutional layers but exponentially increased if dilated convolutions are used. Figure 3 shows receptive fields when discrete convolutions and dilated convolutions are used for $N = 1, 2, 3$.

Convolutional Neural Network with Dilated Convolutions

In this work, we employ DilatedNet [17] to segment person on images because DilatedNet produces segmentation results with

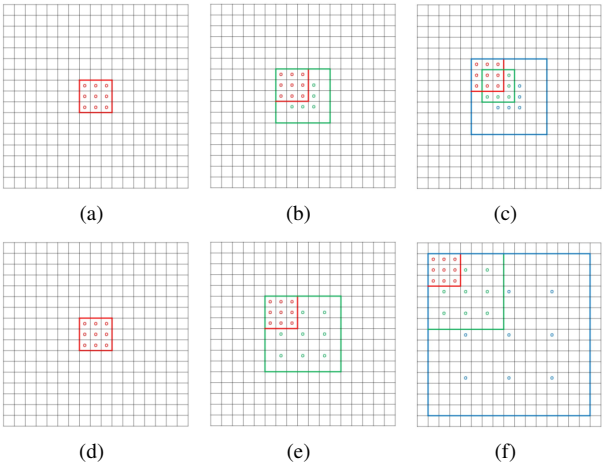

Figure 3. *Receptive fields using 2D discrete convolutions and 2D dilated convolutions (a) a receptive field is* 3×3 *using discrete convolutions when* $N = 1$ *(b) a receptive field is* 5×5 *using discrete convolutions when* $N = 2$ *(c) a receptive field is* 7×7 *using discrete convolutions when* $N = 3$ *(d) a receptive field is* 3×3 *using dilated convolutions when* $N = 1$ *(e) a receptive field is* 7×7 *using dilated convolutions when* $N = 2$ *(f) a receptive field is* 15×15 *using dilated convolutions when* $N = 3$

high accuracy. DilatedNet is composed of a front-end module and a context module which are described below.

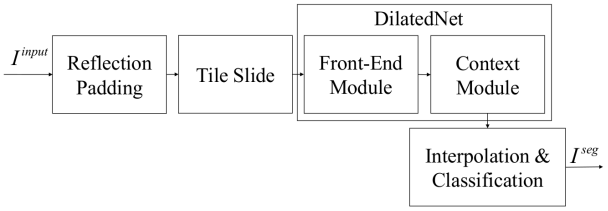

Figure 4. *Block diagram of our convolutional neural network*

Figure 4 is a block diagram of our convolutional neural network. First of all, an input image, I^{input}, is reflection-padded by m, where m is a thickness of a margin lost during convolutional layers and pooling layers in the front-end module. Then a tile of size $n \times n$ is slided on the padded input image by $s = n - 2m$ to both horizontal and vertical directions to generate a cropped image which becomes an input to the front-end module. Here, if a cropped image on the tile is smaller than $n \times n$, the cropped image is reflection-padded to be $n \times n$. In this implementation, $m = 186$, $n = 900$, and $s = 528$.

The original DilatedNet [17] architectures of the front-end module and the context module are described in Table 1 and Table 2, respectively. Here, Conv$k-c-l$ is a convolutional layer with a filter size of $k \times k$, c channels, and a dilation factor of l, ReLU is rectified linear unit activation function, Maxpool$p$ is max-pooling layer with a filter size of $p \times p$ and a stride of p. C is the number of channels in the output image. Here, C can be decided by the number of objects classified by the CNN. The front-end module is motivated from a VGG-16 network [7] with some changes by using dilated convolutions and removing some pooling layers. In the front-end module, no padding is done during convolutional layers. The size of the input image to the front-end module is $n \times n$, a margin lost during the front-end module is m, and a total downsampling rate is $d = 8$, so the size of the output feature map of the front-end module will be $q \times q$ where $q = \frac{n-2m}{d} = 66$. The feature map then becomes an input to the context module which enhances segmentation performance by a large receptive field with a few coefficients. In the context module, padding is done in all convolutional layers to keep feature maps the same size. Note that the receptive field of the context module is 67×67 which is larger than the size of the feature map, $q \times q = 66 \times 66$.

The next step interpolates the feature map from the context module by a factor of d to reconstruct the feature map to be the same size of the input image. Here, bilinear interpolation is used in each channel. The final step is a pixel-wise classification to generate a final segmented image. Figure 5 shows an example of our person segmentation method steps.

Table 1. Architecture of an original front-end module

Layer 1	Conv3-64-1 + ReLU
Layer 2	Conv3-64-1 + ReLU
Layer 3	Maxpool2
Layer 4	Conv3-128-1 + ReLU
Layer 5	Conv3-128-1 + ReLU
Layer 6	Maxpool2
Layer 7	Conv3-256-1 + ReLU
Layer 8	Conv3-256-1 + ReLU
Layer 9	Conv3-256-1 + ReLU
Layer 10	Maxpool2
Layer 11	Conv3-512-1 + ReLU
Layer 12	Conv3-512-1 + ReLU
Layer 13	Conv3-512-1 + ReLU
Layer 14	Conv3-512-2 + ReLU
Layer 15	Conv3-512-2 + ReLU
Layer 16	Conv3-512-2 + ReLU
Layer 17	Conv7-4096-4 + ReLU
Layer 18	Conv1-4096-1 + ReLU
Layer 19	Conv3-C-1

Table 2. Architecture of a context module

Layer 1	Conv3-C-1 + ReLU	3×3 receptive field
Layer 2	Conv3-C-1 + ReLU	5×5 receptive field
Layer 3	Conv3-C-2 + ReLU	9×9 receptive field
Layer 4	Conv3-C-4 + ReLU	17×17 receptive field
Layer 5	Conv3-C-8 + ReLU	33×33 receptive field
Layer 6	Conv3-C-16 + ReLU	65×65 receptive field
Layer 7	Conv3-C-1 + ReLU	67×67 receptive field
Layer 8	Conv1-C-1 + ReLU	67×67 receptive field

The original DilatedNet architecture was initially designed for semantic segmentation on 21 classes ($C = 21$) [17]. Person segmentation requires only 2 classes (person and background). In order to reduce the size of a model and decrease execution time, we designed a compact DilatedNet by reducing the number of channels, c, in convolutional layers in a front-end module. Table 3 shows the architecture of our front-end module.

In this work, we designed two CNNs: Fine-tuned DilatedNet and Compact DilatedNet. Here, "Fine-tuned DilatedNet"

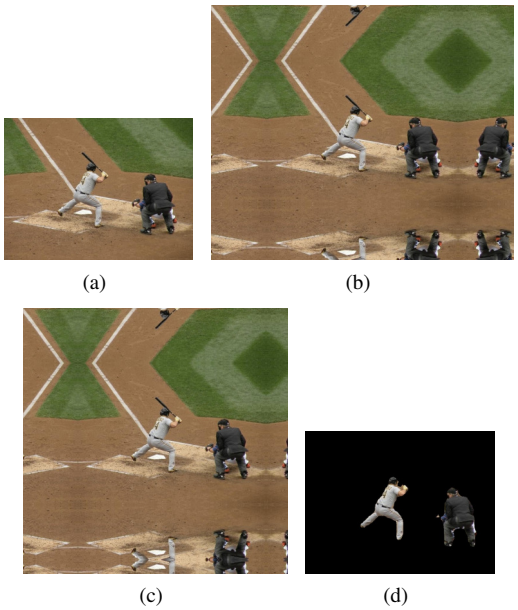

Figure 5. An example of our person segmentation method steps using dilatedNet [17] (a) an input image (b) a reflection-padded image (c) a cropped image on a tile (d) a segmented image after interpolation

Table 3. Architecture of our front-end module

Layer 1	Conv3-64-1 + ReLU
Layer 2	Conv3-64-1 + ReLU
Layer 3	Maxpool2
Layer 4	Conv3-64-1 + ReLU
Layer 5	Conv3-64-1 + ReLU
Layer 6	Maxpool2
Layer 7	Conv3-64-1 + ReLU
Layer 8	Conv3-64-1 + ReLU
Layer 9	Conv3-64-1 + ReLU
Layer 10	Maxpool2
Layer 11	Conv3-64-1 + ReLU
Layer 12	Conv3-64-1 + ReLU
Layer 13	Conv3-64-1 + ReLU
Layer 14	Conv3-64-2 + ReLU
Layer 15	Conv3-64-2 + ReLU
Layer 16	Conv3-64-2 + ReLU
Layer 17	Conv7-4096-4 + ReLU
Layer 18	Conv1-4096-1 + ReLU
Layer 19	Conv3-*C*-1

refers to a model using the original DilatedNet architecture [17] fine-tuned with our training images and "Compact DilatedNet" refers to a model with reduced number of channels. Fine-tuned DilatedNet and Compact DilatedNet are implemented in Caffe [30]. To train Fine-tuned DilatedNet, we fine-tuned the pretrained model [17] using 127,888 training images from 65,808 MSCOCO images [26], 5,601 ADE20K images [27, 28], and 56,479 augmented images using the HP private dataset as foreground and the MSCOCO dataset as background generated during our data augmentation process. We trained 100K iterations using stochastic gradient descent (SGD) with a learning rate of 10^{-5} and a momentum of 0.9. To train Compact DilatedNet, we first initialized all weights according to Xavier Initialization [31]. We trained Compact DilatedNet using 136,711 training images from 65,808 MSCOCO images [26], 5,601 ADE20K images [27, 28], 56,479 augmented images using the HP private dataset as foreground and the MSCOCO dataset as background, and 8,823 augmented images using the Cityscapes dataset generated during our data augmentation process. We first trained 100K iterations with a learning rate of 10^{-3}, and then trained 100K iterations with a learning rate of 10^{-4}, and then trained 100K iterations with a learning rate of 10^{-5}, with a momentum of 0.9, using SGD.

Experimental Results

Our methods were tested on 1,000 images from the MSCOCO dataset [26] and 591 images from the Gallagher dataset [32].

First of all, we compared segmentation performances of Fine-tuned DilatedNet without and with 56,479 augmented images using the HP private dataset as foreground and the MSCOCO dataset as background during training process, shown in Figure 6. We visually confirmed that a CNN with our data augmentation process can segment closer to person boundaries.

Also, we compared the segmentation performance of Fine-tuned DilatedNet and Compact DilatedNet, shown in Figure 7. We visually confirmed that the segmentation results from Compact DilatedNet is similar to the segmentation results from Fine-tuned DilatedNet.

Next, we compared the segmentation results to other known segmentation results such as PSPNet [22] and SharpMask [25]. Both PSPNet and SharpMask are trained to segment person, and we consider other objects as background. Here, we used pre-trained networks for PSPNet and SharpMask. Figure 8 shows the segmentation results on various methods. We observed that PSPNet segmented person in an image but other objects overlapping with person were also segmented. Also, SharpMask failed to segment a portion of person and the segmentation boundary was not accurate. Fine-tuned DilatedNet can accurately segment person and remove background objects. Compact DilatedNet can also successfully segment person from background.

Table 4 compares the size of models and execution time (loading time and running time) of our methods and other segmentation methods. Here, a loading time is the time for loading the model and a running time is the time for running one image. Fine-tuned DilatedNet and Compact DilatedNet segments an image faster than PSPNet and SharpMask. Although Fine-tuned DilatedNet has a great performance, the size of the model is quite big. We are able to develop a smaller model which can have a similar performance to Fine-tuned DilatedNet.

Table 4. A comparison between our methods and other segmentation methods

Segmentation Method	Model Size	Loading Time	Running Time
PSPNet [22]	197 MB	7 sec	
SharpMask [25]	212 MB	10 sec	
Fine-tuned DilatedNet	537 MB	3 sec	0.46 sec
Compact DilatedNet	**120 MB**	**1 sec**	**0.23 sec**

Figure 6. *A comparison between segmentation results without and with augmented images using the HP private dataset as foreground and the MSCOCO dataset as background during fine-tuning a pretrained DilatedNet [17] (a) an original image (b) a segmented image without augmented images (c) a segmented image with augmented images (d) an original image (e) a segmented image without augmented images (f) a segmented image with augmented images (g) an original image (h) a segmented image without augmented images (i) a segmented image with augmented images*

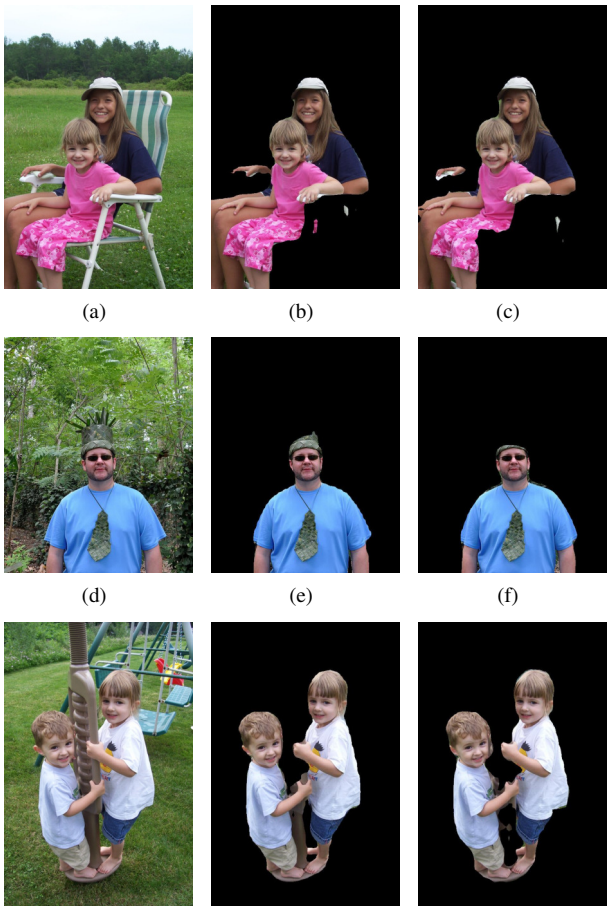

Figure 7. *A comparison between segmentation results using Fine-tuned DilatedNet and Compact DilatedNet (a) an original image (b) a segmented image using Fine-tuned DilatedNet (c) a segmented image using Compact DilatedNet (d) an original image (e) a segmented image using Fine-tuned DilatedNet (f) a segmented image using Compact DilatedNet (g) an original image (h) a segmented image using Fine-tuned DilatedNet (i) a segmented image using Compact DilatedNet*

Lastly, we tested Fine-tuned DilatedNet and Compact DilatedNet on webcam video. Here, in each frame in a webcam video, person is segmented using a CNN and background is replaced by a user-specified image. The webcam video using Compact DilatedNet has approximately 4 fps whereas the webcam video using Fine-tuned DilatedNet has approximately 2 fps due to the size of models. Figure 9 shows an example of the webcam frames using Fine-tuned DilatedNet and Compact DilatedNet which visually do not have a big difference.

Conclusions

In this paper, we presented a person segmentation method using convolutional neural networks with dilated convolutions. We designed two data augmentation methods to improve segmentation performance. We included augmented images to our training set to fine-tune the original DilatedNet [17]. Our Fine-tuned DilatedNet provides accurate segmentation especially near person boundaries. The original DilatedNet architecture requires a heavy computation so the usage in webcam application may not be practical. By reducing the channel number we were able to develop and train our Compact DilatedNet to increase the speed without sacrificing segmentation performance. We expect that our proposed method can save bandwidth during video conferencing by only transmitting pixel information on a person mask and setting background as a user-specified image. In the future, we plan to reduce the size of the model more by using network pruning techniques such as deep compression [33].

References

[1] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut": Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, August 2004.

[2] T. Zhao and R. Nevatia, "Bayesian human segmentation in crowded situations," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2003, Madison, WI.

[3] M.D. Rodriguez and M. Shah, "Detecting and segmenting humans in crowded scenes," *Proceedings of the ACM International Conference on Multimedia*, pp. 353–356, September 2007, Augsburg, Germany.

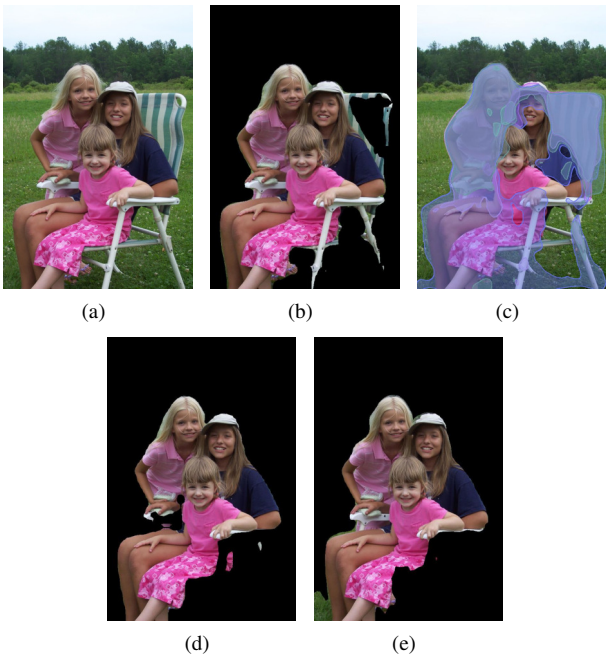

Figure 8. *A comparison between our methods and other segmentation methods (a) an original image (b) a segmented image using PSPNet [22] (c) a segmented image using SharpMask [25] (d) a segmented image using Fine-tuned DilatedNet (e) a segmented image using Compact DilatedNet*

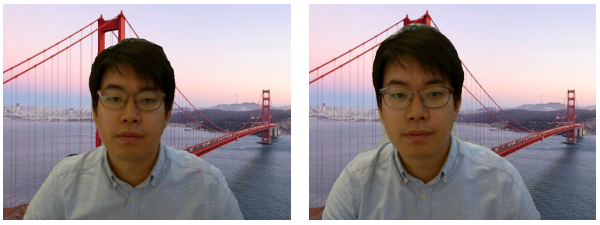

Figure 9. *An example of webcam frames (a) a webcam frame using Fine-tuned DilatedNet (b) a webcam frame using Compact DilatedNet*

[4] E. Horbert, K. Rematas, and B. Leibe, "Level-set person segmentation and tracking with multi-region appearance models and top-down shape information," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1871–1878, November 2011, Barcelona, Spain.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Proceedings of the Neural Information Processing Systems*, pp. 1097–1105, December 2012, Lake Tahoe, NV.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, April 2015.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, June 2014, Columbus, OH.

[9] R. Girshick, "Fast R-CNN," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, December 2015, Santiago, Chile.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Proceedings of the Neural Information Processing Systems*, pp. 91–99, December 2015, Montreal, Canada.

[11] 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, June 2015, Boston, MA.

[12] D. Ciresan, A. Giusti, L.M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," *Proceedings of the Neural Information Processing Systems*, pp. 2843–2851, December 2012, Lake Tahoe, NV.

[13] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, October 2016.

[14] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1520–1528, December 2015, Santiago, Chile.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pp. 231–241, October 2015, Munich, Germany.

[16] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, "A real-time algorithm for signal analysis with the help of the wavelet transform," *Wavelets: Time-Frequency Methods and Phase Space Proceedings of the International Conference*, pp. 286–297, December 1987, Marseille, France.

[17] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, April 2016.

[18] 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," *arXiv preprint arXiv:1606.00915*, May 2017.

[19] P. Krahenbuhl and V. Koltun, "Efficient inference in fully connected CRFs with gaussian edge potentials," *Proceedings of the Neural Information Processing Systems*, pp. 109–117, December 2011, Granada, Spain.

[20] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," *arXiv preprint arXiv:1611.08408*, November 2016.

[21] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Proceedings of the Neural Information Processing Systems*, pp. 2672–2680, December 2014, Montreal, Canada.

[22] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6230–6239, July 2017, Honolulu, HI.

[23] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5168–5177, July 2017, Honolulu, HI.

[24] P.O. Pinheiro, R. Collobert, and P. Dollar, "Learning to segment object candidates," *Proceedings of the Neural Information Processing Systems*, December 2015, Montreal, Canada.

[25] P.O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollar, "Learning to refine object segments," *Proceedings of the European Conference*

on Computer Vision, pp. 75–91, October 2016, Amsterdam, The Netherlands.

[26] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C.L. Zitnick, and P. Dollar, "Microsoft COCO: Common objects in context," *Proceedings of the European Conference on Computer Vision*, pp. 740–755, September 2014, Zurich, Switzerland.

[27] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through ADE20K dataset," *arXiv preprint arXiv:1608.05442*, August 2016.

[28] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5122–5130, July 2017, Honolulu, HI.

[29] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, June 2016, Las Vegas, NV.

[30] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, June 2014.

[31] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 249–256, May 2010, Sardinia, Italy.

[32] A.C. Gallagher and T. Chen, "Clothing cosegmentation for recognizing people," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, Anchorage, AK.

[33] S. Han, H. Mao, and W.J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, February 2016.

Author Biography

David Joon Ho is currently a Ph.D student of Electrical and Computer Engineering at Purdue University. He received his BS and MS in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign. His research interests include deep learning, image processing, and computer vision.

Dr. Qian Lin is a distinguished technologist working on computer vision and deep learning research in HP Labs. Dr. Lin joined the Hewlett-Packard Company in 1992. She received her BS from Xi'an Jiaotong University in China, her MSEE from Purdue University, and her Ph.D. in Electrical Engineering from Stanford University. Dr. Lin is inventor/co-inventor for 44 issued patents. She was awarded Fellowship by the Society of Imaging Science and Technology (IS&T) in 2012, and Outstanding Electrical Engineer by the School of Electrical and Computer Engineering of Purdue University in 2013.