

# Semantic Binary Segmentation using Convolutional Networks without Decoders

Shubhra Aich, William van der Kamp, Ian Stavness  
Department of Computer Science  
University of Saskatchewan, Canada

{s.aich, william.vanderkamp, ian.stavness}@usask.ca

## Abstract

*In this paper, we propose an efficient architecture for semantic image segmentation using the depth-to-space (D2S) operation. Our D2S model is comprised of a standard CNN encoder followed by a depth-to-space reordering of the final convolutional feature maps. Our approach eliminates the decoder portion of traditional encoder-decoder segmentation models and reduces the amount of computation almost by half. As a participant of the DeepGlobe Road Extraction competition, we evaluate our models on the corresponding road segmentation dataset. Our highly efficient D2S models exhibit comparable performance to standard segmentation models with much lower computational cost.*

## 1. Introduction

Semantic segmentation refers to classifying the pixels of images or videos according to specific categories of objects or background regions known as stuff [4]. Like many other areas of computer vision, research on semantic segmentation has received a tremendous performance boost with the emergence of deep learning in recent years. All recent semantic segmentation models follow a general encoder-decoder type of architecture where the encoder front-end of the network extracts the features necessary for a particular task, and the decoder back-end of the network approximates the segmentation map from these salient features.

FCN [23] is the earliest example of an encoder-decoder style semantic segmentation network. This architecture is built by converting the fully connected (FC) layers at the backend of traditional image classification architectures like AlexNet [15] or VGG [25] into fully convolutional layers with  $1 \times 1$  convolution followed by upsampled or fractional convolution or deconvolution to generate the pixel-level segmentation map. Skip connections from the higher resolution layers at the convolutional front-end are added for better information gain or performance in both FCN and U-Net [21]. Next comes the SegNet [3] or deconvolutional network [18] architectures, where for upsampling,

max-pooling indices are used with the stack of simple convolutional layers. In SegNet, FC layers or equivalent convolution layers are omitted in order to reduce both the memory requirements and computational complexity of the network. Our network design has some similarities to both FCN and SegNet. First, like SegNet, we do not use any FC layers or their equivalent. Moreover, our network uses  $1 \times 1$  convolution like FCN, but with a much smaller size. Unlike FCN or SegNet, we do not use any deconvolution operation, rather a rearrangement of the feature grid is done by a depth-to-space operation with negligible computational cost.

Recent state-of-the-art segmentation approaches follow the traditional approach of requiring some sort of decoder back-end. The DeepLab models [5, 7] use atrous convolutions [28] in the backend of the CNN models to generate comparatively higher resolution coarse score maps ( $1/8^{th}$  of the original image) instead of using max-pooling. They also use spatial pyramid pooling [10] with the atrous convolution of variable rates for better multi-scale prediction as well as fully connected CRF [14] to finetune the bilinearly upsampled score maps. Overall this results in a complex, multi-stage pipeline where CNN and CRF are trained separately, though the latest version of the model [6] omits CRF post-processing. RefineNet [16] uses a multi-path refinement architecture as its decoder. Each refinement block fuses high-level, and low-level feature maps using residual convolution layers and bilinear upsampling for shape adjustment. Also, the authors use the residual sequence of pooling for efficient fusion of multi-scale, pooled prediction. The pyramid scene parsing network (PSPNet) [29] uses pyramid pooling on the feature map of the ResNet equipped with dilated or atrous convolution for global context aggregation. The authors also added a branch in the middle of the ResNet to propagate auxiliary loss for faster convergence. The large kernel paper [19] uses larger convolution kernels to empirically cover larger receptive field [30]. Large symmetric kernels are broken down into a few asymmetric kernels to reduce the computational complexity. Finally, recent literature focused on road mapping from aerial images combine the power of deep learning for pixel-

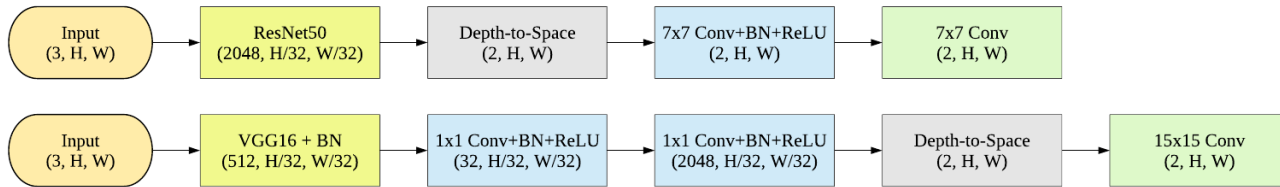


Figure 1. D2S models with ResNet50 (top) and VGG16-BN (bottom) backbones. Because of the differences in the shape of the final output layer, the placement of the rearrangement or depth-to-space block is different for these models. The last one or two convolution operations incur a negligible computational cost due to the small number of channels (2).

level segmentation with graph-based optimization for the extraction of road topology [17, 8]. All of these related works on semantic segmentation share the common feature of including a decoder sub-network composed of different variations of convolutional and/or upsampling blocks.

In this paper, we challenge the basic assumption that a decoder sub-network is needed to approximate a segmentation map from encoder-generated feature maps. We hypothesize that, at least for relatively easy segmentation tasks, such as binary segmentation, the computationally-complex decoder procedure can be replaced by a simple depth-to-space rearrangement of the output of the final convolution layer, without loss of segmentation accuracy. We call this type of encoder with depth-to-space (D2S) spatial reordering the D2S network. From an efficiency perspective, our D2S architecture needs only half of the computation to learn the same task.

The idea of depth-to-space reordering that we use in our paper to replace long-range decoders is identical to the sub-pixel convolution for image super-resolution [24]. Depth-to-space operations have also been used before for benchmarking different decoding approaches [27], but in a different way. In that work, multiple instances of the depth-to-space reordering operation are used for  $2 \times 2$  upsampling in between the convolution layers in the decoder, whereas in our D2S model we use a single depth-to-space block as a replacement for a large stack of convolution layers.

We incorporate our D2S idea as a participant in the *DeepGlobe Road Extraction* challenge. For the competition, our focus is to use a novel and efficient approach instead of an ensemble of sophisticated models. We evaluate our approach on the road extraction dataset. Our D2S model based on the ResNet50 encoder achieves 60.60% mean intersection over union (IoU) whereas the top entry has IoU of 65.60% on the validation set (at the time of writing). This small difference with the best entry validates our hypothesis that for at least easy segmentation problems, encoder-only architectures without any decoder might be a reasonable and efficient model of choice.

## 2. Method

With most current segmentation networks, such as FCN, DeepLab, or RefineNet, the predicted score map has a lower resolution than the input image. For SegNet, the shape of the input image and output segmentation map are the same with the same amount of computation in both encoder and decoder. For that reason, SegNet incurs twice the computational cost of the encoder alone.

In comparison, our architectural design allows pixel-wise prediction naturally. We train the network in such an arrangement where the neighborhood pixel contributions are stored along the depth dimension and then just re-ordered. One apparent drawback might be that the model will have artifacts in the final prediction map because the contextual mapping task in the neighborhood of the prediction map is interrupted. However, we did not see any such problems in practice, likely because the network learns to overcome the spatial disruption while training end-to-end.

### 2.1. Architecture

We employ Resnet50 [11] and VGG16 [25] with batch normalization [12] as the backbone or encoder of our network with minor differences due to the difference in the dimension of the output of the final convolution layer in these models. The complete architectures for both versions are depicted in Figure 1.

We use a similar D2S reordering as proposed previously for the image super-resolution problem [24]. The domain of image super-resolution is a mapping task from the image space to itself with greater detail in the output space. Also, there is no encoder-decoder type of architectures; rather the raw image is taken as a dense feature map and a simple stack of convolution layers are used to produce the corresponding high-resolution version. Thus, the D2S transformation for super-resolution is an arguably more natural operation compared to our case, where we use this block right at the end of the encoder sub-network. In that sense, although we use a similar encoder type of network, it works as a decoder directly from the image space. This decoding task refers to mapping the RGB image pixels into the binary pixel space considering a semi-global context. Theoretically, the

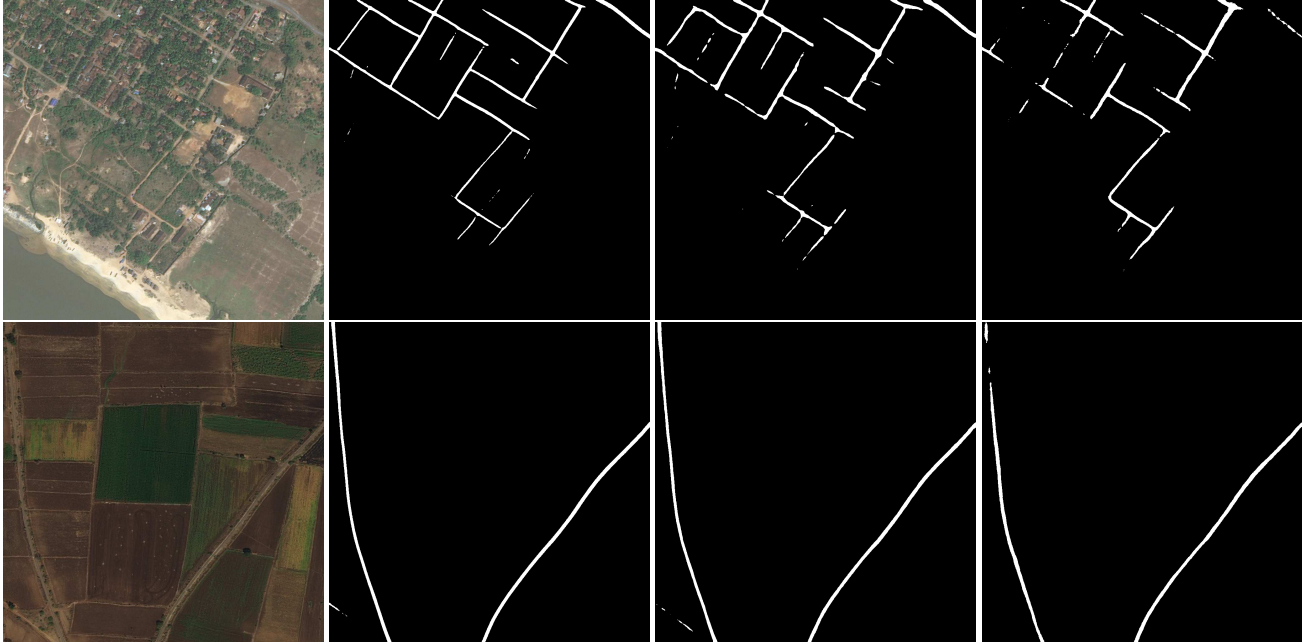


Figure 2. (Left to Right) Sample image; Segmentation maps generated by ResNet50-D2S, VGG16-BN-D2S, and Segnet models, respectively.

Table 1. Results of our D2S models compared to SegNet as a baseline on the validation set.

Model	Pixel IoU
ResNet50-D2S	0.6060
VGG16-BN-D2S	0.5897
SegNet [3]	0.5612

amount of context covered depends on the depth of the network. Moreover, we incorporate two-dimensional dropout [26] after each max-pooling for the VGG model and after each block except the last one for the ResNet model for improved performance. For the VGG based model, we use  $1 \times 1$  convolution to obtain the depth necessary for pixel-level mapping (Figure 1, bottom).

### 3. Experiments

In this section, we provide a brief description of the dataset, the implementation and training details of our models, and results achieved on the validation set in the leaderboard compared to SegNet as a baseline.

#### 3.1. Dataset

At the time of writing, the DeepGlobe Road Extraction dataset is only open for the participants of the “DeepGlobe Road Extraction” challenge [9]. The dataset consists of 6226 and 1243 training and validation images, respectively, each of resolution  $1024 \times 1024$ . This dataset is a binary image segmentation problem, where the road pixels are marked as foreground and the rest of the objects and

stuff are background. One of the challenges of this dataset is that it is highly imbalanced in terms of the number of pixels per class, i.e. roads are thin lines within the images and therefore occupy few pixels as compared to the background.

#### 3.2. Training and Implementation

We train all of our models with ImageNet [22] pre-trained encoders. In the beginning, we started training with  $224 \times 224$  patches extracted from around the true positives in the ground truth due to the scarcity of the foreground (road pixels) in the images. However, empirically we found that having the full context of the image, i.e., training with the whole image at once helped to improve the model’s performance.

We use PyTorch [20] as the deep learning framework. All the models are finally trained with full resolution images, and their color jittered versions with the batch size varying in the range of [3, 8]. The models are trained on NVIDIA TITAN Xp GPUs and an NVIDIA Quadro P6000 workstation. We use the Adam optimizer [13] with an initial learning rate of 0.0001 which is later reduced based on the training statistics. Codes and pre-trained models are publicly available.<sup>1</sup>

#### 3.3. Results

Table 1 lists the pixel-level intersection over union (IoU) [9] for three different models on the validation set in the

<sup>1</sup><https://github.com/littleaich/deepglobe2018>

competition leaderboard. We provide the performance metric for a standard SegNet architecture to benchmark our D2S models for a couple of reasons. First, the front-end of our VGG-D2S model is a replica of the SegNet encoder, which is the set of convolution layers of the VGG16 model with batch normalization. Therefore, it is more straightforward to compare the symmetric decoder of SegNet against our spatial rearrangement strategy. Second, SegNet has been reliably employed for binary image segmentation problems with substantial accuracy in recent works [2, 1].

From Table 1, we find the D2S models to have comparable performance to the SegNet architecture. Figure 2 shows two sample images and their corresponding segmentation maps generated by the three models. From this figure, it is also evident the qualitative performance of the models are quite similar.

Moreover, at the time of writing, the top entry in the leaderboard had IoU of 0.6560 which is  $\sim 5\%$  better than our best model. We anticipate that like other featured competitions, the top entries in this competition comprise an ensemble of different approaches, whereas our result is generated using only the D2S models described in this paper. Therefore, we conclude that for segmentation problems containing only a few classes, heavy-decoder models like SegNet can be reliably replaced by our efficient D2S architecture without significant loss in performance.

## 4. Conclusion

In this paper, we propose an efficient image segmentation network, called D2S, that uses only a convolutional encoder along with spatial reordering of the final feature maps. Empirically, we show that for relatively easier image segmentation problems, such as binary segmentation, the D2S models give comparable performance to the standard models. Although we only evaluate our model on a simpler problem, this kind of depth-to-space architecture may also be useful in more complex tasks, which we plan to explore in future research.

## Acknowledgment

This research was undertaken thanks in part to funding from the Canada First Research Excellence Fund, the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the Microsoft AI for Earth and NVIDIA GPU programs.

## References

[1] S. Aich, I. Obsyannikov, I. Ahmed, I. Stavness, A. Josuttis, K. Strueby, H. S. Duddu, C. Pozniak, and S. Shirtliffe. Deepwheat: Estimating phenotypic traits from images of crops

using deep learning. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[2] S. Aich and I. Stavness. Leaf counting with deep convolutional and deconvolutional networks. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2080–2089, Oct 2017.

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, Dec 2017.

[4] H. Caesar, J. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. In *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018.

[5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062, 2014.

[6] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[7] 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*, 40(4):834–848, April 2018.

[8] D. Costea, A. Marcu, M. Leordeanu, and E. Slusanschi. Creating roadmaps in aerial images with generative adversarial networks and smoothing-based optimization. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2100–2109, Oct 2017.

[9] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar. DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images. *ArXiv e-prints*, May 2018.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, Sept 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[14] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 109–117. Curran Associates, Inc., 2011.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

- [16] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5168–5177, July 2017.
- [17] G. Mittyus, W. Luo, and R. Urtasun. Deeproadmapper: Extracting road topology from aerial images. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3458–3466, Oct 2017.
- [18] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1520–1528, Dec 2015.
- [19] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun. Large kernel matters – improve semantic segmentation by global convolutional network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1743–1751, July 2017.
- [20] PyTorch. <http://pytorch.org/>.
- [21] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.
- [22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [23] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, April 2017.
- [24] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, June 2016.
- [25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [26] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 648–656, June 2015.
- [27] Z. Wojna, V. Ferrari, S. Guadarrama, N. Silberman, L. Chieh Chen, A. Fathi, and J. Uijlings. The devil is in the decoders. In *BMVC*, 2017.
- [28] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *CoRR*, abs/1511.07122, 2015.
- [29] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, July 2017.
- [30] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *CoRR*, abs/1412.6856, 2014.