Morphable Convolutional Neural Network for Biomedical Image Segmentation

Huaipan Jiang*, Anup Sarma*, Mengran Fan*, Jihyun Ryoo*, Meenakshi Arunachalam†
Sharada Naveen†, Mahmut T. Kandemir*
*The Pennsylvania State University, University Park, PA-16802.
Email: {hzj5142, avs6194, mxf97, jihyun, mtk2}@psu.edu
†Intel, Email: {meena.arunachalam, sharada.naveen}@intel.com

***Abstract*—We propose a morphable convolution framework, which can be applied to irregularly shaped region of input feature map. This framework reduces the computational footprint of a regular CNN operation in the context of biomedical semantic image segmentation. The traditional CNN based approach has high accuracy, but suffers from high training and inference computation costs, compared to a conventional edge detection based approach. In this work, we combine the concept of morphable convolution with the edge detection algorithms resulting in a hierarchical framework, which first detects the edges and then generate a layer-wise annotation map. The annotation map guides the convolution operation to be run only on a small, useful fraction of pixels in the feature map. We evaluate our framework on three cell tracking datasets and the experimental results indicate that our framework saves ∼30% and ∼10% execution time on CPU and GPU, respectively, without loss of accuracy, compared to the baseline conventional CNN approaches.**

***Index Terms*—Image Segmentation, Approximate Computing**

I. Introduction

Semantic image segmentation is an important and challenging step in many biomedical image analysis works. Examples of such tasks include neural structure construction [1], cell tracking [2], [3], retina layer segmentation [4], and MRI image analysis [5]. Many recent studies first segment the EM (Electron Microscopy) images, which classify all the pixels into different categories. For example, to solve the problem of cell tracking, the first step is to segregate each image into two groups: *foreground (cells)* and *background*. The foreground here refers to the cells on the image, while the background refers to the part of the image which contains trivial information.

The traditional algorithms for image segmentation include edge detection [6], clustering [7], and region growing [8]. However, these traditional algorithms lack high accuracy for very complex images, which is a critical metric in biomedical domain. Motivated by this, several convolutional neural network(CNN) based approaches [9]–[13] have been proposed to improve the accuracy of segmentation. These neural networks typically first extract the features of the image, then combine the features with the location information, and finally output the predicted category for each pixel.

While neural networks with large number of layers and weights significantly increase the accuracy of the segmentation, the execution latency costs and the energy consumed during the training and inference become the main bottleneck [14]. For example, during PDT (photo dynamic therapy) treatment, MRI images are segmented in real time, while treating the patient. Any improvement in MRI Image analytics directly help the treatment accuracy in real time. To build more efficient segmentation approaches, recent studies have proposed attention models [15]–[18]. The distinguishing characteristic of these models is that they identify "regions of interest" (RoI) *before* invoking segmentation. This two-step pipeline reduces total amount of computation by eliminating the redundant computation on the background parts of the images. In the other words, under an attention model, segmentation only works on the RoI.

Intuitively, to solve the image segmentation problem, we only need to detect the "borders" of the objects and then fill the area inside. In addition, we observe that for identifying candidate regions, CNN based networks also spend considerable computation effort. Whereas, in biomedical dataset context, identifying candidate regions is a relatively simpler task. Consequently, edge detection is a straightforward solution to image segmentation. However, one of the main limitations of the existing edge detection algorithms is the "false positives" they generate [19], that is, some lines are incorrectly detected as borders. Therefore, we only use edge detection result towards candidate region generation (irregular in shape) to be later processed through morphable convolution framework. Our proposed framework thus effectively combines the speed of traditional edge detection algorithms with the accuracy of convolutional neural networks.

In summary, in this work we make the following major contributions:

1. We propose the concept of *Morphable Convolution*, which is capable of performing convolution on selective regions of *irregular* shapes. The neural network only classify the pixels which are selected (annotated) on the input images. Our framework does not require to re-train the weights for a given dataset. Instead, we can reuse the weights from the existing conventional CNN models.

2. We propose a hierarchical framework consisting of an edge detector module and morphable convolutional neural network applied towards semantic segmentation. In this context, we

This work is supported in part by NSF grants 1629915, 1629129, 1763681 and 2008398 as well as a grant from Intel.

978-3-9819263-5-4/DATE21/©2021 EDAA

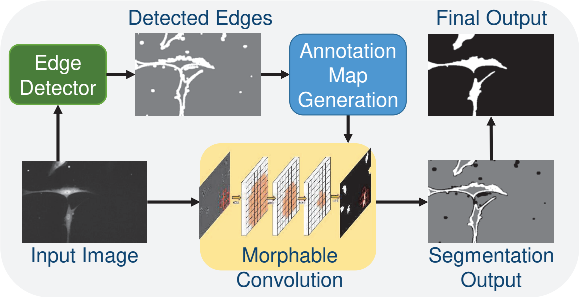


Fig. 1: Overview of the framework

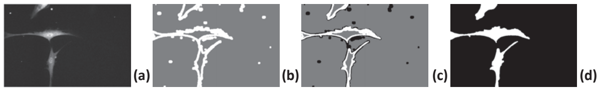


Fig. 2: The workflow of applying the Morphable U-net on Fluo-C2DL-MSC dataset. We first take the input image (a) and detect the edges of the objects (b) on the images. Those pixels which is detected as edges will be considered as the annotation. We then use Morphable convolution neural network to only classify the pixels which annotated as 1 on the image (c). The white part in (c) are classified as "foreground", the black part are classified as "background", whereas, we did not perform any operation on the grey part since those area are omitted. Finally, we fill up the inside area of each objects to generate the segmentation mask for the image (d).

introduce the notion of an ”annotation map”, computed backwards for every layer, of a given segmentation CNN network. It guides the network to classify only the region of pixels that are selected (annotated) on the input feature map.

3. We evaluate our work on recent biomedical datasets, and compare it against other competing approaches to demonstrate its efficiency and efficacy. Our experiments indicate that our hierarchical image segmentation framework achieve $\sim$30% and $\sim$10% execution time savings on CPUs and GPUs, respectively, without drop in accuracy.

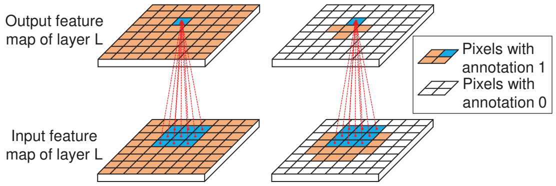
II. Background and Related Work

A. Biomedical Semantic Image Segmentation:

Semantic image segmentation is an important step in many biomedical image processing projects. Here, all the pixels on a given image are classified into different categories. Different from instance segmentation, semantic segmentation only classify the category of each object, while in instance segmentation, each objects in the same category are distinguished. For example, different cells are classified into different classes in instance segmentation, whereas, semantic segmentation only classify pixels into two classes (cells and background).

B. Edge Detection based Image Segmentation:

Edge Detection is a technique used in image segmentation. Traditional approaches first apply filter kernels [20] to remove the noise on the image and then employ gradient kernels [21] to compute the derivatives of each pixel along each direction. The pixels with higher gradient are considered as the pixels on the border. Another class of algorithms detect the edge by selecting the local maximum and minimum values of the pixels, which is also referred as ridge detection [22]. Recent studies [23]–[25] also proposed neural networks to locate edges of the object in an image. For all aforementioned approaches, the borders are connected after they are detected, and finally the interior area is filled to complete the segmentation process.



(a) Normal Convolution (b) Morphable Convolution

Fig. 3: Feature map and annotation for a convolution layer. Only calculating colored pixels.

C. CNN based Image Segmentation:

Convolutional Neural Networks have been successfully employed in the context of biomedical image segmentation problems. Ciresan et al. [11] trained a classification network with a sliding-window on the image, where each pixels is represented by its neighboring pixels. FCN [12] proposed a fully connected prediction module to integrate the location and classification features. Based on FCN, U-net [13] refined the architecture of FCN and proposed a "U-shaped" architecture. The low resolution feature map is concatenated with the high resolution features from the corresponding max-pooling layers to avoidd the resolution loss. Apart from the aforementioned networks, SegNet [9], also employs a u-shaped architecture with a index-based deconvolution. StarDist [26] predict the shape of each cell object by predicting a "star-convex polygon" for every pixel. This approach improve the accuracy of generating the segmentation results for the crowded cells. However, none of these approaches targets execution time, which is a critical factor in the inference of biomedical image segmentation. Cellpose [27] employed a U-net like CNN with human labeled cell shape as annotation to classify the pixels inside the cells. Although it reduce the execution time when compare to process on the entire image, it requires expensive human labor.

III. Morphable Convolution Framework

During the inference phase of segmentation, our framework first identifies the RoIs and then computes the segmentation mask for the candidate regions, which organized in a *hierarchical manner*. However, our RoIs are *not* necessarily of rectangle-shaped. As shown in Figure 1, the input images are first fed to an *edge detector module*. After the edges in an image have been generated, they are sent to the *annotation generator module*, which generates the layer-wise annotation maps for the segmentation network. The annotation map is a matrix, of the same height and width, as the output feature map, but its entries are either ”0” or ”1”. The annotation map guides the segmentation network on the problem of which pixels need to be classified, instead of indiscriminately classifying all the pixels. After all the pixels near the border have been classified,

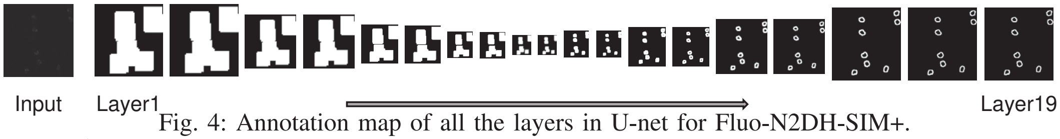


Fig. 4: Annotation map of all the layers in U-net for Fluo-N2DH-SIM+.

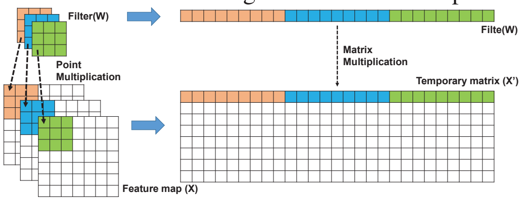


Fig. 5: Converting the feature map into a temporary matrix.

We fill up the inside area and generate the segmentation results. Figure 2 illustrated an example of using Morphable U-net for segmentation on an image from Fluo-C2DL-MSC dataset.

A. Annotation Map Generation

Each convolution layer in our morphable CNN framework receives an input tensor $I \in \mathcal{R}^{C\times H\times W}$ and an annotation map $A \in \{0,\ 1\}^{H'\times W'}$ as the input. The annotation map guides the framework to calculate results on specific regions of the output tensor (to make it simple, we assume that our input is 2D images for the remaining of our paper). The annotation map is a matrix consisting of 0s and 1s. As shown in Figure 3, the conventional convolution layer calculates the results for all the pixels on the output feature map, while our morphable convolution only computes the results for the pixels with a corresponding 1 on the annotation map.

As one can expect, the annotation map for each layer is different. For example, in Figure 3(b), layer L is a convolution layer with a filter size of 3. In order to calculate the result of the blue pixel p on the output feature map, we need to first get the value of 9 pixels on the input feature map of layer L (colored as blue). Hence, on the annotation map of layer L-1, all the corresponding pixels of those 9 pixels should be 1.

The annotation map of the last layer is exactly the same as the annotation generated by the edge detector. We then generate the annotation map for each layer – from the last one to the first one. Taking U-net as an example. Figure 4 illustrates the annotation map of all convolution layers in U-net with Fluo-N2DH-SIM+ dataset. One can clearly observe the variations in the annotation map across the convolution layers.

B. Morphable Convolution Implementation

We implemented morphable convolutional neural network framework in CAFFE [28] and modify the *im2col* (which *converts convolution to matrix multiplication*) in convolution layer implementation. As shown in Figure 5, the input feature map is duplicated and copied to a temporary matrix where each row corresponds to a pixel on the output feature map. After performing the matrix multiplication, we obtain the output feature map. More specifically, let us denote the input feature map as $X \in \mathcal{R}^{C\times H\times W}$, the convolution filter as $W \in \mathcal{R}^{K\times K\times C\times C'}$, and the output feature map as $Y \in \mathcal{R}^{C'\times H'\times W'}$. We first convert X into a temporary matrix $X' \in \mathcal{R}^{H'\times W'\times K\times K\times C}$. For each pixel $p \in \{H' \times W'\}$ in the output feature map Y, we convert $K \times K$ values from X to X' and then compute Y by $Y = W \times X'^{T}$.

In our framework, we only copy the dependent data for pixels that have 1 on the annotation map. In the result, $X' \in \mathcal{R}^{N\times K\times K\times C}$, where N is the number of 1s in the annotation map. It can be observed that, this framework is much more efficient compared to the conventional convolution framework since $N \leqslant H' \times W'$.

IV. Experimental Results

We evaluate five segmentation approaches in this section, which include two edge detection and three CNN approaches. All five approaches are listed in the first column of Table I. We evaluate our framework and run all our experiments on a cell segmentation dataset of the microscopic images [2], [3], which contain three different types of cells (**PhC-C2DH-U373 (D-I)**, **Fluo-C2DL-MSC (D-II)** and **Fluo-N2DH-SIM+ (D-III)**).

A. Segmentation with Edge Detection

In this section, we study two different edge detection based segmentation algorithms, one is a classical edge detection algorithm, the other is a simple neural network that can detect the edges on the image. Here, we compare the segmentation resultss of both algorithms by detecting the edges first and then filling the inside area. The first two rows in Table I give the IoU (Intersection over Union) results of the segmentation mask with the ground truth for those two edge detection based approaches. It can be observed that these two edge detection-based segmentation approaches perform similar; whereas, they both are much worse than CNN-based approach (original U-net) in terms of the accuracy. However, although these two edge detection based segmentation algorithms have same level of accuracy performance, the edge detected by the CNN model is more precise and clean than the edges detected by Canny operator. In the other word, CNN edge detector based morphable CNNs can potentially achieve more speedup than the morphable CNNs guided by Canny detector since it proposed less "false edges".

B. Comparison against Conventional CNNs

To show the benefits of employing our morphable convolution framework on CNNs, we compare Canny-based and CNN-based morphable U-net against the original U-net. The former uses Canny edge detector to obtain the annotation while the later approach use CNN edge detector to generate the annotation. In both approaches, the annotation is later send to the morphable convolutional framework to guide the convolution operators. Note that we use the *same trained weights* for *all* versions of the U-net framework (Original, Canny-based and CNN-based). In this experimental analysis, we use normalized execution time as our latency metric, and IoU (Intersection over Union) as our accuracy metric. We show latency results in

TABLE I: IoU results.

	D-I	D-II	D-III
Canny Edge Detector segmentation	0.538	0.699	0.639
CNN Edge Detector segmentation	0.495	0.686	0.690
Original U-net	**0.933**	0.737	0.691
Morphable U-net (Canny)	0.931	0.743	0.715
Morphable U-net (CNN)	0.931	**0.747**	**0.717**

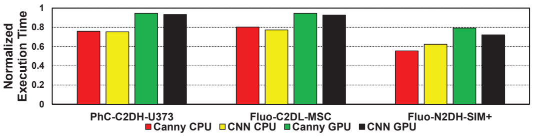


Fig. 6: Performance of Morphable U-net against original U-net over three datasets on both CPU and GPU. Each experiment results have been normalized to the correspond baseline U-net.

Figure 6 and accuracy results in Table I. For all three datasets, two morphable U-nets achieve same level of accuracy with the original U-net. All CNN based segmentation approaches are outperform than the edge detection based softwares in terms of the accuracy. It is important to point out that the accuracy result for morphable convolution might has small difference from the original U-net, since some pixels on the border area have different predictions result by edge detector and U-net. Another observation is that the morphable convolution based U-net achieves higher speedups on all the three datasets for both the CPU and GPU implementations. It can also be observed that the CPU implementation achieves more relative speedup, since the serial annotation generation process (running on CPU) contributes more to the total execution time of GPU implementation. Here, the latency results of both morphable CNN implementations already include the edge detection and annotation generation time.

V. Conclusion

In this work, we propose and experimentally evaluate a morphable convolution framework, which can be used to construct a hierarchical framework for biomedical semantic image segmentation. This framework enables convolution layers to operate on irregularly shaped regions of input feature map using an annotation map. We evaluate our framework on three cell tracking datasets. The collected results indicate that, our approach achieves ∼30% and ∼10% execution time saving on average for CPU and GPU based platforms respectively, against baseline CNN model. We believe that our framework is quite general and orthogonal to other performance-enhancing optimizations on image segmentation.

References

[1] A. Cardona *et al.*, "An integrated micro- and macroarchitectural analysis of the drosophila brain by computer-assisted serial section electron microscopy," *PLOS Biology*, vol. 8, pp. 1–17, 10 2010.

[2] M. Maška, V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Ederra, A. Urbiola, T. España, S. Venkatesan, D. M. Balak, *et al.*, "A benchmark for comparison of cell tracking algorithms," *Bioinformatics*, vol. 30, no. 11, pp. 1609–1617, 2014.

[3] V. Ulman, M. Maška, K. E. Magnusson, O. Ronneberger, C. Haubold, N. Harder, P. Matula, P. Matula, D. Svoboda, M. Radojevic, *et al.*, "An objective comparison of cell-tracking algorithms," *Nature methods*, vol. 14, no. 12, p. 1141, 2017.

[4] P. P. Srinivasan, S. J. Heflin, J. A. Izatt, V. Y. Arshavsky, and S. Farsiu, "Automatic segmentation of up to ten layer boundaries in sd-oct images of the mouse retina with and without missing layers due to pathology," *Biomed. Opt. Express*, vol. 5, pp. 348–365, Feb 2014.

[5] G. Litjens *et al.*, "Evaluation of prostate segmentation algorithms for mri: The promise12 challenge," *Medical Image Analysis*, vol. 18, no. 2, pp. 359 – 373, 2014.

[6] T. Lindeberg and M.-X. Li, "Segmentation and classification of edges using minimum description length approximation and complementary junction cues," *Computer Vision and Image Understanding*, 1997.

[7] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, (Berkeley, Calif.), pp. 281–297, University of California Press, 1967.

[8] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1452–1458, Nov 2004.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," 2015.

[10] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*, pp. 424–432, Springer, 2016.

[11] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in Neural Information Processing Systems 25*, pp. 2843–2851, Curran Associates, Inc., 2012.

[12] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[13] 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*, pp. 234–241, Springer, 2015.

[14] A. Sarma, H. Jiang, A. Pattnaik, J. Kotra, M. T. Kandemir, and C. R. Das, "Cash: compiler assisted hardware design for improving dram energy efficiency in cnn inference," in *Proceedings of the International Symposium on Memory Systems*, pp. 396–407, 2019.

[15] W. Huang, D. He, X. Yang, Z. Zhou, D. Kifer, and C. L. Giles, "Detecting arbitrary oriented text in the wild with a visual attention model," in *Proceedings of the 2016 ACM on Multimedia Conference*, MM '16, (New York, NY, USA), pp. 551–555, ACM, 2016.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[17] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017.

[18] H. Jiang, A. Sarma, J. Ryoo, J. B. Kotra, M. Arunachalam, C. R. Das, and M. T. Kandemir, "A learning-guided hierarchical approach for biomedical image segmentation," in *2018 31st IEEE International System-on-Chip Conference (SOCC)*, pp. 227–232, IEEE, 2018.

[19] G. Shrivakshan and C. Chandrasekar, "A comparison of various edge detection techniques used in image processing," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 5, p. 269, 2012.

[20] K. Ito, "Gaussian filter for nonlinear filtering problems," in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, vol. 2, pp. 1218–1223, IEEE, 2000.

[21] J. Canny, "A computational approach to edge detection," in *Readings in computer vision*, pp. 184–203, Elsevier, 1987.

[22] K. Magnusson, "Cell tracking for automated analysis of timelapse microscopy," 2011.

[23] R. Wang, "Edge detection using convolutional neural network," in *International Symposium on Neural Networks*, pp. 12–20, Springer, 2016.

[24] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[25] Z. Yu, C. Feng, M.-Y. Liu, and S. Ramalingam, "Casenet: Deep category-aware semantic edge detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[26] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers, "Cell detection with star-convex polygons," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 265–273, Springer, 2018.

[27] C. Stringer, M. Michaelos, and M. Pachitariu, "Cellpose: a generalist algorithm for cellular segmentation," *bioRxiv*, 2020.

[28] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675–678, 2014.