Generic Image Segmentation in Fully Convolutional Networks by Superpixel Merging Map

Jin-Yu Huang and Jian-Jiun Ding

National Taiwan University
{r07942085, jjding}@ntu.edu.tw

Abstract. Recently, the Fully Convolutional Network (FCN) has been adopted in image segmentation. However, existing FCN-based segmentation algorithms were designed for semantic segmentation. Before learning-based algorithms were developed, many advanced generic segmentation algorithms are superpixel-based. However, due to the irregular shape and size of superpixels, it is hard to apply deep learning to superpixel-based image segmentation directly. In this paper, we combined the merits of the FCN and superpixels and proposed a highly accurate and extremely fast generic image segmentation algorithm. We treated image segmentation as multiple superpixel merging decision problems and determined whether the boundary between two adjacent superpixels should be kept. In other words, if the boundary of two adjacent superpixels should be deleted, then the two superpixels will be merged. The network applies the colors, the edge map, and the superpixel information to make decision about merging suprepixels. By solving all the superpixel-merging subproblems with just one forward pass, the FCN facilitates the speed of the whole segmentation process by a wide margin meanwhile gaining higher accuracy. Simulations show that the proposed algorithm has favorable runtime, meanwhile achieving highly accurate segmentation results. It outperforms state-of-the-art image segmentation methods, including feature-based and learning-based methods, in all metrics.

1 Introduction

Image segmentation is fundamental and important in many image processing applications. There are many existing image segmentation algorithms, including the region growing method [1], the mean shift method [2], the watershed [3, 4], the normalized cut [5, 6], the graph-based method [7], and the superpixel-based method [8–10].

In recent years, deep learning techniques have also been adopted in image segmentation [11–14]. With supplicated deep learning architectures, one can achieve good segmentation results with enough training time. However, these learning-based algorithms are used to produce semantic segmentation but not generic segmentation results.

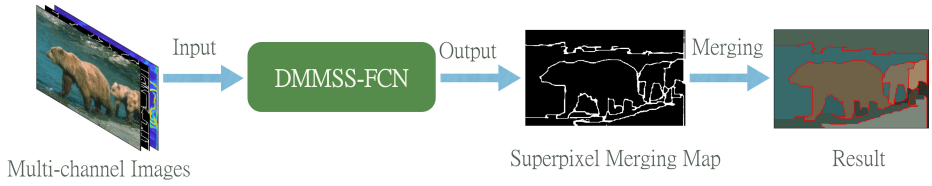

Fig. 1. Overview of the proposed DMMSS-FCN algorithm. Multiple feature maps including an original RGB image are stacked together to form the input data. Then, they are fed into the fully convolutional network. The output is the superpixel merging map that determines which boundaries should be kept. With such information, we perform superpixel merging based on the output of the model to produce the final segmentation result.

Before learning-based segmentation algorithms were developed, many advanced image segmentation algorithms are based on superpixels. However, due to the irregularity of sizes and shapes of superpixels, it is hard to apply superpixels in a learning-based generic segmentation architecture.

Therefore, in this study, we broke down the whole image segmentation problem into several superpixel merging decision problems. Furthermore, each superpixel merging process can be converted into a boundary keeping problem. That is, whether the boundaries between two adjacent superpixels should be deleted or not. We proposed a novel algorithm that leverages the Fully Convolutional Network (FCN) for learning generic image segmentation. We call it Deep Merging Models for Superpixel-Segmentation by Fully Convolutional Networks (DMMSS-FCN). With the use of FCN, all of those superpixel merging problems can be solved in just one forward pass. That is, it is extremely efficient. First, the proposed DMMSS-FCN model will predict whether the pixels along the boundary of two adjacent superpixels should be keep. Following, a majority voting technique will be applied to decide the existence of all superpixel boundaries. Therefore, the final image segmentation result will be produced with minimum effort.

We use 5-channel stacked images as the input, including an RGB image, a superpixel boundary map, an edge detection map. Since different superpixel algorithms will produce different superpixel boundary maps, by vary the parameters of different superpixel algorithms, numerous training data can be easily obtained. That is, our method does not require many human-annotated ground truth. Furthermore, our method is fully automatic, that is, user does not need to assign number of regions in prior.

We show the overview of the proposed DMMSS-FCN algorithm in Fig. 1. We will discuss the detail of DMMSS-FCN in Section 3.

2 Related Work

2.1 Superpixels

Superpixels are a group of pixels with similar colors and locations. There are many types of superpixels, including the entropy rate superpixel (ERS) [15] and the simple linear iterative clustering (SLIC) [16] superpixel. In [2], the superpixel generated by mean shift was proposed. It has good edge-preserving property and the number of superpixels have not to be specified in advance. Moreover, its boundaries highly match the borders of objects. Recently, deep learning-based superpixels like superpixel sampling network (SSN) [17], and segmentation-aware loss (SEAL) superpixels [18] were proposed. They both outperform non deep learning-based superpixels by a wide margin.

2.2 Classical Segmentation

Most classical segmentation methods utilize hand-crafted features such as color, histogram, gradient, or texture to perform segmentation. Many of them are still widely used today, such as the graph-based method [7] and the normalized cut [6]. Arbelaz et. al. [3,4] proposed a method based on the global information to perform the oriented watershed transform and generated an ultra-metric contour map for hierarchical segmentation.

Moreover, superpixel-based segmentation algorithms like the method of segmentation by aggregating superpixels (SAS) [8] perform segmentation based on merging superpixels using some local grouping cues. Kim et. al. [9] used a full range affinity model and Yang et. al. [10] proposed a spectral clustering method based on Gaussian Kernel similarity measure for image segmentation. These superpixel-based segmentation algorithms have good performance. However, due to the irregular shapes and sizes of superpixels, it is hard to embed deep learning techniques in superpixels-based algorithms.

2.3 Deep Learning in Image Segmentation

Recently, many semantic segmentation algorithms applied the deep neural network were developed. In [12, 13], the fully convolutional network (FCN) were proposed to improve the performance of image segmentation and object detection. In [14], the conditional random field (CRF) was applied in the pixel-wise segmentation method of DeepLab. In [11], Xia and Kulis proposed the W-Net based on the FCN to perform segmentation. In [19], Haeh et. al. introduced a method based on the CNN to detect split error in segmented biomedical images. In [20], Chen et. al. extracted deep features and used them for superpixel merging. In [21], Liu et. al. applied the FCN for superpixel merging.

In this study, we integrate the merits of superpixel-based methods and state-of-the-art learning-based methods and propose a high accuracy image segmentation algorithm.

3 Proposed Algorithms: DMMSS-FCN

In this work, we proposed an effective way of integrating FCN with generic image segmentation. That is, with the use of superpixel, we can encode the superpixel merging problems with keeping of superpixel boundaries into generic image segmentation. In Fig. 2(a), the existence of boundary between two superpixels implies that they are separated from each other, in other words, they are not merged. On the contrary, the disappearance of boundary means they are merged. Therefore, with the use of the FCN, we perform dense prediction on every adjacent boundaries. After one forward pass in the FCN, all the pixels along the boundaries are classified into two classes, *keeping or removing*. Then we measure the tendency of keeping a boundary by majority voting technique defined in (1) for quantization. Hence, we can easily recover the segmentation result.

$$BoundaryRate(i,j) = \frac{\#\text{ of pixels of } (keep \text{ label} \cap Bnd(i,j))}{\#\text{ of pixels of } Bnd(i,j)} \tag{1}$$

where the *keeping* label indicates that the pixel is predicted to be on the boundary of some object and should be kept. For example in Fig. 2(b), there are 15 pixels along the boundary $Bnd(i,j)$, 10 of them are predicted as the *keeping* label (orange circles) while 5 of them(black circles) are of the *removing* label, resulting in a $BoundaryRate$ of 2/3. Therefore, we can thresholding on the $BoundaryRate$ to get the segmentation result from the FCN output.

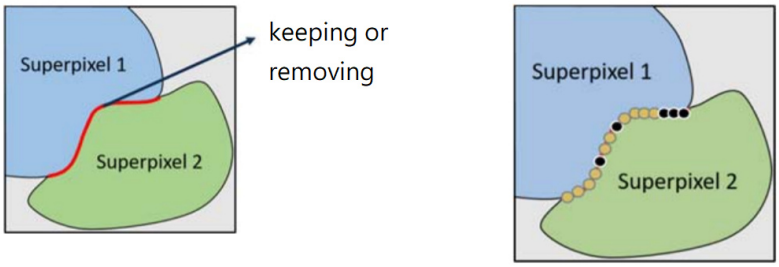

Fig. 2. (a) Converting superpixel merging problem into superpixel boundary keeping problem. (b) BoundaryRate Example.

3.1 Five-channel Input Data

We defined our input data by concatenating a RGB (3 channels) image with a superpixel boundary map (1 channel) and a edge-detection map (1 channel). The RGB image indicates the original image while the superpixel boundary map is a binary image with only the boundary between two adjacent superpixels are marked as positive (the *keeping* label). The superpixel boundary map generation is shown in Fig. 3. There are many advanced learning-based contour generation algorithms [22–24]. In this work, the RefineContourNet (RCN) [22] is adopted for edge detection. An example of the output of the RCN is shown in the right subfigure of Fig. 3.

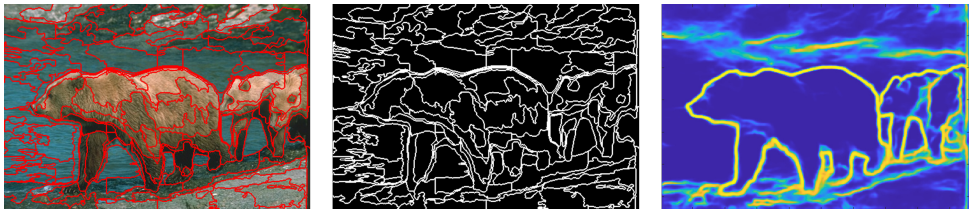

Fig. 3. Component of input data. **Left**: Superpixel result from SEAL-ERS [18]. **Middle**: Binary image with superpixel boundaries marked as 1s, others are 0s. **Right**: Edges extracted by the RCN.

We have also tried different combination of concatenating input images. Ablation studies have been carried out to analyze the impact of different input channel in Sec. 4.3.

Since we want our model to be adaptive to any input superpixel type, the superpixel boundary map plays an important role in the whole process. That is, with any superpixel result as prior input, our model can determine which boundary should be keep, with the superpixel boundary map as attention mechanism.

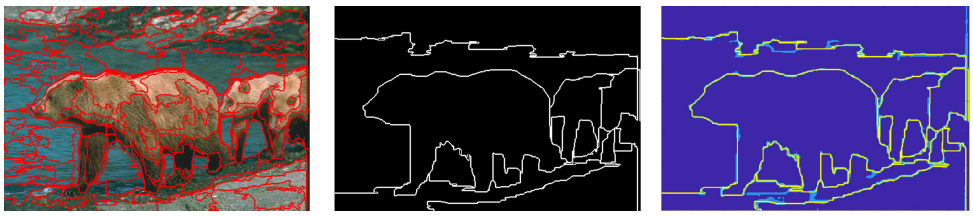

Fig. 4. Groundtruth generation. **Left**: SEAL-ERS superpixel result. **Middle**: Perfect segmentation by oracle. **Right**: Groundtruth superpixel boundary map by converting the resulting groundtruth into binary image.

3.2 GroundTruth Generation and Output

In [3], the Segmentation Covering (SC), the Probabilistic Rand Index (PRI), and the Variation of Information (VI) are proposed to be the standard evaluation metrics for generic image segmentation. Among all, the PRI as follows is often treated as the most important metric, where c_{ij} indicates the case where pixels i and j belong to the same region, p_{ij} is the probability, S is the resulting segmentation, and G_k is a set of groundtruth.

$$PRI\{S, G_k\} = \frac{1}{T} \sum_{i<j} [c_{ij} p_{ij} + (1 - c_{ij}) + (1 - p_{ij})] \tag{2}$$

That is, we use an oracleguided process to produce the highest achievable PRI score for a given oversegmentation(superpixel) result. From that, we can acquire the groundtruh by transforming the results into a binary boundary map called the superpixel merging map which is the ideal output of our model. The

groundtruth generation result is shown in Fig. 4. Sufficient training data can be obtained by varying the parameters of superpixel generation algorithms. Therefore, the model is designed to be adaptive for different superpixel type. We use BoundaryRate to recover segmentation result. In Fig. 5, we show that for different superpixel type as prior input, our models will generate corresponding superpixel merging maps.

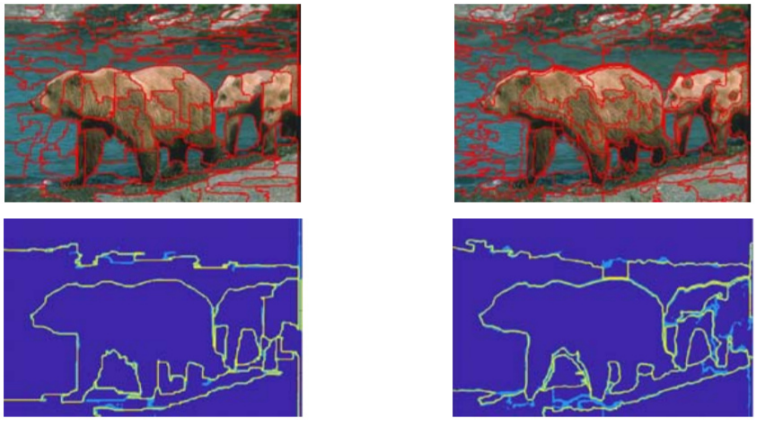

Fig. 5. Different superpixel boundary map input. **First Row**: Superpixel boundaries overlapped with the original images. **Second Row**: Superpixel merging maps overlapped with corresponding groundtruth merging maps.

3.3 Training Architecture

Since our goal is to perform pixel-wise prediction on the image, and examine the *BoundaryRate* along the boundary of two adjacent superpixels, the localization of predicted labels is crucial. Therefore, we adopted the DeepLab V3+ [25] FCN architecture proposed by Chen et. al.. The DeepLab V3+ utilized the atrous convolution in the encoder side for the better field of view and a simple but effective decoder with short cut skipped through the encoder part is added to this architecture, making it a highly accurate fully convolutional network structure while preserving good spacial information.

In this paper, we adopted the InceptionResNetV2 [26] as the hidden encoder architecture, and the output stride of encoder is set to 16. And the batch size is set to 13 with the size of input data is 321x481x5.

Since this is a binary classification problem of two labels, *keeping or removing* labels, with the *keeping* class is far less than the other one, we applied the weighted binary cross entropy loss in the following equation to calculate the loss. Furthermore, Adam optimizer is adopted to update the parameters of the networks with initial learning rate of 0.0001 and divided by 0.5 for every 10 epochs, and stopped training after 100 epochs.

$$weighted\, BCE\,(T,S) = -\sum_{i} \left[\omega_0\,(T_i log S_i) + \omega_1 log\,(1-S_i)\right] \tag{3}$$

where S is the output superpixel merging map, and $S_i \in (0, 1)$ denotes the predicted probability value at i^{th} pixel in S. T is the groundtruth merging map, and $T_i \in 0, 1$ denotes the groundtruth label in T. The weighting factors $\omega_0 = 1$ and $\omega_1 = 38$ is the ratio of two classes.

We applied the Berkeley Segmentation DataSet 500, with 200 test images, 200 training images, and 100 validation images. As we mentioned in Sec 3.2, different superpixel algorithm can produce different superpixel and groundtruth pairs. Therefore, we adopted the SEAL-ERS [18] and the SSN [17] both with the number of superpixels set to 100, and 200 to generate the training data, resulting in 800 training images, and 400 validation images in total.

3.4 Inference and Superpixel Merging

As we shown in Fig. 6, we first concatenate RGB image with superpixel boundary map and RCN edge map to form the 5-channel input data. Then we perform one forward pass through the networks to obtain superpixel merging map. Afterwards, we use the *BoundaryRate* in (1) to measure how many predicted *keeping* label are on each boundary of adjacent superpixels. Then, we perform thresholding merging procedure as we by using the formula in (1). Here we adopted an adaptive thresholding technique which first start merging by thresholding with the lowest threshold values and increase the threshold values bit-by-bit until there are no candidates for merging. After the thresholding value (0.5 in this paper) is reached, the whole merging process stops and the final segmentation result is obtained.

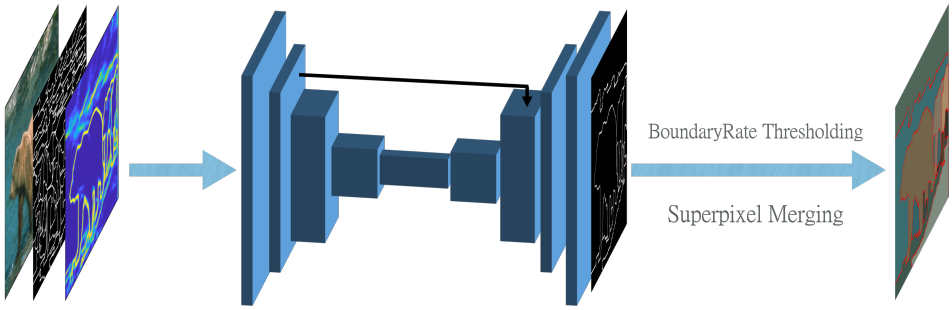

Fig. 6. Overview of DMMSS-FCN. We first form the 5-channel input data by concatenating a RGB (3 channels) image with a superpixel boundary map (1 channel) and an edge-detection map(1 channel). Then, the input is fed to perform pixel-wise prediction and output the superpixel merging map. Then, the BoundaryRate in (1) is used to threshold each superpixel boundary to perform superpixel merging and obtain the segmentation result.

4 Experiments

In this section, we carried out tons of experiments and ablation studies to justify that our proposed *DMMSS-FCN* can outperform many state-of-the-art algorithms, even compared to our own *DMMSS*, the proposed *DMMSS-FCN* can still

Table 1. Results on the BSDS500 dataset.

Method	VI	PRI	SC
Ncuts	2.23	0.78	0.45
Canny-owt-ucm	2.19	0.79	0.49
Felz-Hutt	2.21	0.80	0.52
Mean Shift	1.85	0.79	0.54
Taylor	1.78	0.81	0.56
W-Net	1.76	0.81	0.57
fPb-owt-ucm	1.70	0.82	0.58
DC-Seg-full	1.68	0.82	0.59
W-Net+ucm	1.67	0.82	0.59
gPb-owt-ucm	1.69	0.83	0.59
cPb-owt-ucm	1.65	0.83	0.59
DMMSS(SSN)	**1.46**	**0.86**	**0.63**
DMMSS-FCN(SSN)	**1.38**	**0.87**	**0.66**
Human Drawing	1.17	0.88	0.72

surpass it in all evaluation metrics while boosting the speed, making *DMMSS-FCN* a highly accurate and efficient generic image segmentation algorithm.

4.1 Segmentation Evaluation

To compare the proposed DMMSS-FCN algorithm to the existing methods, we evaluate the performance on the standard metrics of segmentation covering (SC), the probabilistic rand index (PRI), and the variation of information (VI) [3]. A higher SC and PRI and a lower VI mean better performance.

We compare the proposed DMMSS-FCN algorithm to the state-of-the-art methods, including the W-Net [11], gPb-owt-ucm [4], DC-Seg-full [27], Taylor [28], Felzenszwalb and Huttenlocher (Felz-Hutt) [7], Mean Shift [2], Canny-owtucm [4], Multiscale Normalized Cuts (NCuts) [5], fPb-owt-ucm [9], cPb-owtucm [9], and our own DMMSS. As the proposed algorithm, all the algorithms compared in Table 1 have not to assign the number of regions in prior.

4.2 Run Time Analysis

We then analyze the run time of the proposed *DMMSS-FCN* and compare it to the state-of-the-art segmentation algorithms of DC-Seg-full [27] and gPb-OWT-UCM [4]. The runtime includes the inference time of generating the superpixel boundary map and the edge map. We evaluate our algorithms on SEAL-ERS superpixels, and perform inference on the BSDS500 test set. In Table 2, the average run time of processing single image is presented. One can find out that the proposed *DMMSS-FCN* has drastically reduce the run time compare to gPb-OWT-UCM, decreasing the runtime up to 384x times less than the gPb-OWT-UCM and 227x times less than DC-Seg-full and meanwhile achieving higher accuracy. More on that, even we switch our *DMMSS-FCN* to CPU mode, our

Table 2. Run time on the BSDS500 test set.

Method	Mode	Time (s)
DMMSS-FCN(SEAL-ERS)	CPU+GPU	**0.26**
DMMSS-FCN(SEAL-ERS)	CPU	**8.2**
DC-Seg-full	CPU	59
gPb-OWT-UCM	CPU	100

proposed *DMMSSFCN* still 12x faster than the gPb-OWT-UCM and 12x faster than the DC-Segfull.

4.3 Ablation Study

We then discuss the functionality of some key component in our proposed algorithm, and how they affect the overall performance. Including the different combination of input data, different superpixel generation algorithm, and different DeepLabV3 Plus implementation detail, furthermore, different inference techniques.

Combination of Input Data Following: We show that different combination of input data could have a great impact on the performance of the model. In this section, we mainly use the SEAL-ERS 100 as the underlying superpixel representation. In Table 3, we show the difference of concatenating different feature map as input data.

Table 3. Performance of different input data combination.

Input Data	VI	PRI	SC
4-channel: RGB(3)+spixel bdry(1)	1.569	0.846	0.547
5-channel: RGB(3)+spixel bdry(1) +RCN edge(1)	**1.446**	**0.863**	**0.634**
5-channel: LAB(3)+spixel bdry(1) +RCN edge(1)	1.458	0.860	0.632
7-channel: RGB(3)+spixel bdry(1) +RCN edge(1)+AffinityXY(2)	1.472	0.861	0.629
7-channel: LAB(3)+spixel bdry(1) +RCN edge(1)+AffinityXY(2)	1.534	0.859	0.606

It is reasonable to think that concatenating another feature map as prior information might improve the performance. As we added the RCN edge-detection map, we got a huge gain in performance. Hence, we tried to concatenated more feature map or change RGB to CIE LAB color space to further improve the performance. For example, we use the Affinity map that generated from the SEAL-ERS as candidates for feature map concatenation. Nevertheless, from the experimental results in Table 3, more channels of feature maps do not imply better performance. In the end, we adopted the 5-channel with RGB image as the input of the model.

Superpixels: We measure the performance over different number of initial superpixels. In Fig. 7, the proposed *DMMSS-FCN*. Since our main superpixel

boundary maps are collected from the initial number of superpixels set to 100, and 200, the peak performance is around 200. However, our proposed *DMMSS-FCN* still maintains great performance over different numbers of initial superpixels even if some of them are not included during training.

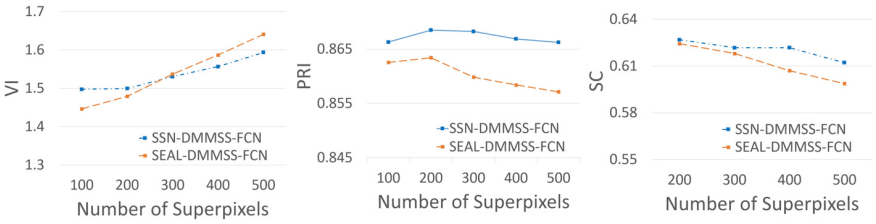

Fig. 7. Performance of different numbers of superpixels between SSN and SEALERS over *DMMSS-FCN*.

In Fig. 5, we show the performance of the proposed model on different types of superpixels, where yellow lines index that the prediction overlapped with the corresponding groundtruth superpixel boundary map, the light blue parts indicate that the model predicts the *keeping* label, and the dark blue parts represent the model predicts the *removing* label. Therefore, as we observe, different input superpixel type will lead the model to predict corresponding superpixel merging map.

DeepLabV3 Plus Variation: Chen et. al. [25] proposed a powerful encoder-decoder-based fully convolutional networks. They suggested different variations of implementation details, including changing the output stride of the model or replacing hidden encoder architecture with any existing CNN architecture such as the ResNet [29] or Xception [30]. More on that, inference techniques like forwarding not one image but four up-down and right-left flipped images at once is also reported as legit method to gain performance. Therefore, we discuss the difference of the performance among the variation of DeepLabV3 Plus applied in our work. We choose SEAL-ERS 100 as the underlying superpixel representation to carry out the following experiments.

In Table 4, we reported the difference among two recent CNN architecture, the Xception [30] and the InceptionResNetV2 [26] with two different output stride setting. In [14], they indicated that smaller output stride could greatly increase the training time and memory usage with a negligible improvement which corresponds to our case here. As for different encoder architecture, Inception-ResNetV2 has better performance over the Xception.

Furthermore, we also adopted the inference technique in [25] to improve the performance, that is, we generate extra images by flipped the original image, then feed them into the networks, then average the scores of the outputs to obtain the final superpixel merging map. In Table 5, we show that a simple but effective flipping technique could boost the performance.

Table 4. Performance of different encoder architecture and output stride.

Encoder Architecture	Output Stride	VI	PRI	SC
Xception	8	1.569	0.860	0.600
Xception	16	1.532	0.857	0.606
InceptionResNetV2	8	1.635	0.854	0.583
InceptionResNetV2	16	1.446	0.863	0.634

Table 5. Result of adopting flipping technique.

Encoder Architecture	Flip	VI	PRI	SC
InceptionResNetV2	No	1.446	0.863	0.634
InceptionResNetV2	YES	1.411	0.864	0.647

4.4 Visual Comparison

In this section, visual comparisons with other segmentation algorithms are presented. In addition to show the simulation results on the BSDS500, we also perform simulations on some real-world images to justify that under any circumstance, our proposed algorithm can be highly accurate and efficient to produce good generic image segmentation results.

BSDS500 Test Images Fig. 8 shows the comparison between the proposed methods and other methods like a deep-learning-based method, the DC-Seg-full method [27], and classical segmentation algorithm gPb-OWT-UCM [4]. As we can see, both the results of the proposed algorithms are much better than that of state-of-the-art algorithms, since ours can produce more general and compact segmentation results compared to the others.

Real-World Images: To further justify the robustness of our model, we pick some modern images that have never been in our training set. In this section, we show some segmentation results on the images taken in the night to test the capability of our proposed methods of handling dark view scenario. In Fig. 9(a) and Fig. 9(b) , one can see that, the proposed model well merges the superpixels belonging to the same object into a region but the other method fails to do this and has severe segmentation leakage. Therefore, such examples prove that our models are robust not only in images taken under great exposures, but in those low-light and blurry scenarios. Additionally, we present simulations of an aircraft engine in Fig. 9(c), which is difficult for another algorithm to perform accurate segmentation on the boundary of engine itself since the background information is similar to the engine. Nevertheless, our proposed method can generate compact segmentation without losing edge information.

5 Conclusion

In this work, a novel image segmentation algorithm that integrates the merit of the Fully Convolutional Network and superpixel-based segmentation is proposed. We call it the *DMMSS-FCN*.

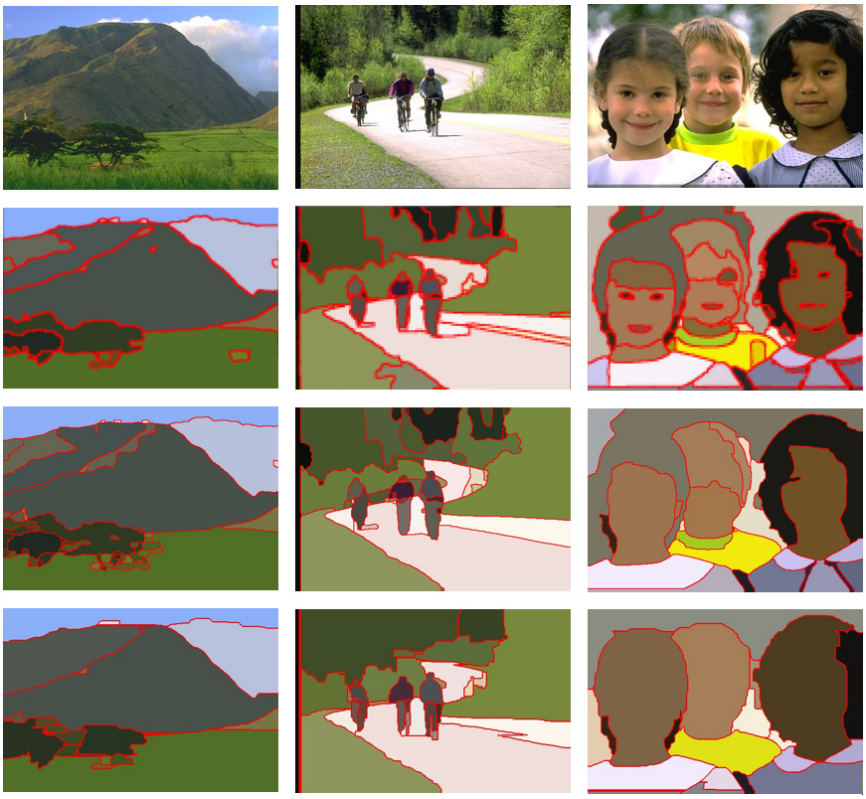

Fig. 8. Visual comparison of segmentation results. (First row): original images. Results produced by (Second row): DC-Seg-full [27]; (Third row): gPb-OWT-UCM [4]; (Fourth row): *DMMSS-FCN* (proposed).

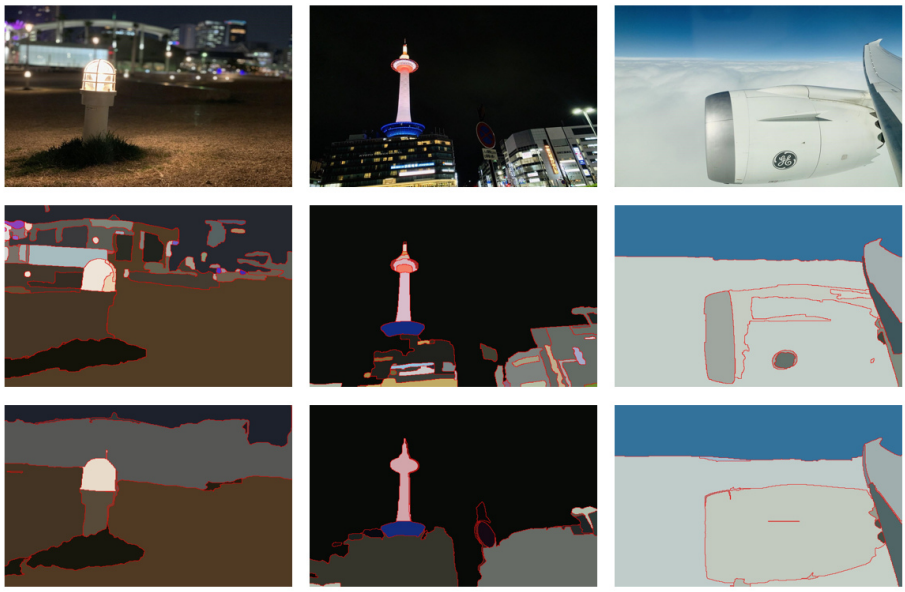

Fig. 9. Real-world image segmentation results. (First row): original images. Results produced by (Second row): gPb-OWT-UCM [4]; (Third row): *DMMSS-FCN* (proposed).

With the use of superpixel, which allows us to be able to solve the generic image segmentation by highly efficient networks. That is, we covert the merging decision into a boundary keeping problem across all superpixel pairs. Hence, with the use of Fully Convolutional Networks, we can solve all those boundary keeping problems with just one forward pass. As a result, the proposed *DMMSS-FCN* algorithm is not only vastly faster than the other state-of-the-art algorithms, but also has high score in accuracy close to human performance.

Additionally, one can obtain large quantity of training data by varying the parameters of existing superpixel algorithms to produce various superpixel boundary maps as input training data. That is, our method does not require a lot of human-annotated ground truth to be fully trained.

We proposed a simple but effective deep-learning-based generic image segmentation algorithm that leverages the FCN for learning generic image segmentation. The proposed algorithm is very effective and simulations show that it is capable of producing reliable segmentation results under many circumstances. Since our algorithm do not require any post-processing technique and further information from user, the simplicity makes it extremely efficient and reliable for any generic image segmentation task. Therefore, we hope this work could offer a great idea to be implemented in the downstream tasks, since segmentation is a quite useful technique in many computer vision applications.

References

1. Shih, F.Y., Cheng, S.: Automatic seeded region growing for color image segmentation (2005)
2. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis (2002)
3. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. (2009) 2294–2301
4. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE transactions on pattern analysis and machine intelligence **33** (2010) 898–916
5. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. **2** (2005) 1124–1131
6. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence **22** (2000) 888–905
7. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International journal of computer vision **59** (2004) 167–181
8. Li, Z., Wu, X.M., Chang, S.F.: Segmentation using superpixels: A bipartite graph partitioning approach. (2012) 789–796
9. Kim, T.H., Lee, K.M., Lee, S.U.: Learning full pairwise affinities for spectral segmentation. IEEE transactions on pattern analysis and machine intelligence **35** (2012) 1690–1703
10. Yang, Y., Wang, Y., Xue, X.: A novel spectral clustering method with superpixels for image segmentation. Optik **127** (2016) 161–167
11. Xia, X., Kulis, B.: W-net: A deep model for fully unsupervised image segmentation. arXiv preprint arXiv:1711.08506 (2017)

12. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. (2015) 1520–1528
13. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. (2015) 3431–3440
14. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence **40** (2017) 834–848
15. Liu, M.Y., Tuzel, O., Ramalingam, S., Chellappa, R.: Entropy rate superpixel segmentation. (2011) 2097–2104
16. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE transactions on pattern analysis and machine intelligence **34** (2012) 2274–2282
17. Jampani, V., Sun, D., Liu, M.Y., Yang, M.H., Kautz, J.: Superpixel sampling networks. (2018) 352–368
18. Tu, W.C., Liu, M.Y., Jampani, V., Sun, D., Chien, S.Y., Yang, M.H., Kautz, J.: Learning superpixels with segmentation-aware affinity loss. (2018) 568–576
19. Haehn, D., Kaynig, V., Tompkin, J., Lichtman, J.W., Pfister, H.: Guided proofreading of automatic segmentations for connectomics. (2018) 9319–9328
20. Cheng, M.M., Liu, Y., Hou, Q., Bian, J., Torr, P., Hu, S.M., Tu, Z.: Hfs: Hierarchical feature selection for efficient image segmentation. (2016) 867–882
21. Liu, Y., Jiang, P.T., Petrosyan, V., Li, S.J., Bian, J., 0001, L.Z., Cheng, M.M.: Del: Deep embedding learning for efficient image segmentation. **864** (2018) 870
22. Kelm, A.P., Rao, V.S., Zölzer, U.: Object contour and edge detection with refinecontournet. (2019) 246–258
23. Maninis, K.K., Pont-Tuset, J., Arbeláez, P., Van Gool, L.: Convolutional oriented boundaries: From image segmentation to high-level tasks. IEEE transactions on pattern analysis and machine intelligence **40** (2017) 819–833
24. Liu, Y., Cheng, M.M., Hu, X., Wang, K., Bai, X.: Richer convolutional features for edge detection. (2017) 3000–3009
25. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
26. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv:1602.07261 (2016)
27. Donoser, M., Schmalstieg, D.: Discrete-continuous gradient orientation estimation for faster image segmentation. (2014) 3158–3165
28. Taylor, C.J.: Towards fast and accurate segmentation. (2013) 1916–1922
29. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. (2016) 770–778
30. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. (2017) 1251–1258