Learning Based Superpixel Merging Model for Image Segmentation

Jin-Yu Huang
Graduate Inst. Communication Engineering
National Taiwan University
Taipei, Taiwan
r07942085@ntu.edu.tw

Jian-Jiun Ding
Graduate Inst. Communication Engineering
National Taiwan University
Taipei, Taiwan
jjding@ntu.edu.tw

Pei-Chi Huang
Graduate Inst. Communication Engineering
National Taiwan University
Taipei, Taiwan
p08942a07@ntu.edu.tw

***Abstract*—Most conventional segmentation methods are superpixel-based. Recently, the convolutional network (CNN) has been adopted in image segmentation. However, most existing CNN-based segmentation algorithms are pixel-wise. Due to the irregular shape and the non-fixed size of superpixels, it is hard to apply superpixels into the CNN architecture directly. In this work, several ideas are proposed to solve this problem. Instead of applying the whole image as the input directly, we apply a square patch that contains only two superpixels as the input of the CNN. Also, instead of generating the segmentation result directly, the output of the CNN is whether the two superpixels should be merged. The proposed algorithm integrates the merits of conventional superpixel-based methods, feature-based methods, and CNN-based methods. Simulations show that the proposed algorithm can achieve very high accurate segmentation results and outperform state-of-the-art methods in all metrics.**

Keywords—Superpixel merging, image segmentation, deep learning, computer vision

I. Introduction

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

In recent years, deep learning techniques have been adopted in image segmentation [9-12]. With sophisticated deep learning architectures, one can achieve good segmentation results with enough training time. However, these learning-based algorithms are pixel-wise methods. 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 segmentation architecture.

In this paper, a novel superpixel-based image segmentation algorithm based on deep neural networks is proposed. Classical superpixel-based algorithms [6-8] utilized several rules to determine whether two superpixels should be merged. In this paper, instead of applying these grouping rules, deep neural networks are applied to decide whether two superpixels should be merged. Different from other learning-based algorithms, instead of applying the whole image as the input, we apply the patch containing only two superpixels as the input of the deep neural network. Moreover, instead of outputting the segmentation result directly, the output of the network in the proposed algorithm is a label to indicate whether two superpixels should be merged.

Fig. 1 shows the overview of the proposed method. Initially, an image is divided into superpixels. Then, learning-based merging models are applied to combine superpixels and obtain the segmentation result. Its detail will be illustrated in Section II.

The source code of the proposed algorithm can be downloaded from [13].

II. Proposed Algorithm

In this section, we illustrate the architecture of the proposed algorithm in detail. It consists of three parts: (i) two-superpixel patch generation; (ii) the training architecture; (iii) superpixel pairing and the merging procedure.

Different from other learning-based methods, which take the whole image as the input and output the segmentation result directly, in the proposed algorithm, the input and the output of the deep neural network are

- Input: A patch containing only two adjacent superpixels, as the two-label patch in Fig. 2.
- Output: A label to indicate whether the two superpixels should be merged.

That is, we convert an image segmentation problem into a classification problem with only two labels: whether two superpixels 'should' or 'should not' be merged. Moreover, since every image contains many superpixel pairs, huge amount of training data can be acquired.

A. Two-Superpixel Patch Generation

To ensure the robustness of the deep learning model, we apply the following two rules to extract two-superpixel patches:

1. The patch should contain only two adjacent superpixels.
2. If the two superpixels do not fill the whole patch, as in the bottom right part of Fig. 1, the image inpainting technique is applied to pad the blank region.

Fig. 1 shows the flowchart of the two-superpixel patch extraction process and some examples of the extracted two-superpixel patches are shown in Fig. 2.

First, we apply superpixel generation algorithms to acquire the initial over-segmented image. In the training set, to increase the diversity of two-superpixel patches, three different methods

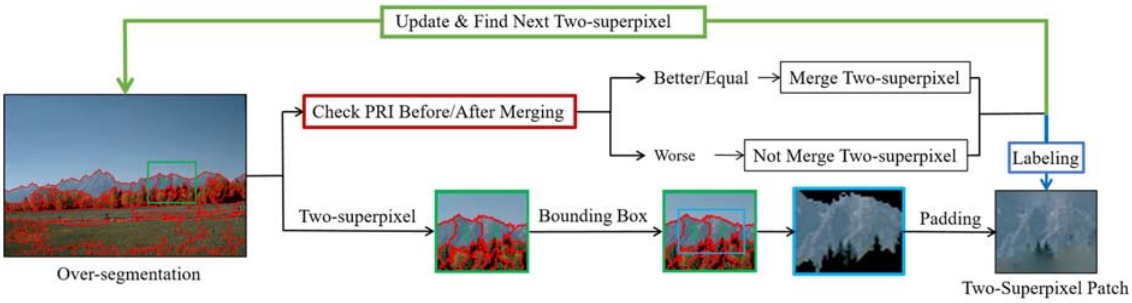

Fig. 1. Flowchart of generating two-superpixel patches.

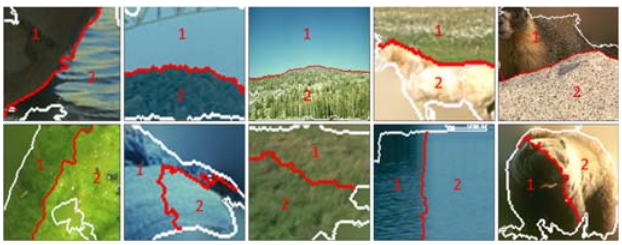

Fig. 2. **Examples of extracted two-superpixel patches.** The subfigures in the upper row are the two-superpixel patches labeled "not to be merged" and those in the lower row are the two-superpixel patches labeled "to be merged". The red curves are the boundaries between the two superpixels and the white lines are the borders of superpixels.

for superpixel generation are applied, including the mean-shift algorithm [2] and deep-learning-based superpixel algorithms like the SEAL [14] and the SSN [15]. Furthermore, by varying the numbers of superpixel in the SEAL and the SSN, one can obtain multi-scaled superpixels.

Note that each two-superpixel patch is treated as a training data. Since there are many superpixel pairs within an image, huge amount of training data can be acquired even if there are limited number of training images.

Then, two-superpixel patches are trimmed to follow the two rules defined at the beginning of this subsection. After applying a bounding box to capture two adjacent superpixels, the anchor point, the width, and the length of the bounding box are recorded. As mentioned in the first rule, the bounding box should cover only two superpixels. Therefore, we choose the middle point on the boundary between the two superpixels as the anchor point.

Then, we find the centroids of two superpixels and calculate the Euclidean distances between the two centroids and the anchor point. The smaller distance is denoted by d. Since better performance can be achieved if the areas of the two superpixels are roughly equal, the width and the length of the bounding box are both set to $2d$. This can ensure that the areas of two superpixels within the bounding box are almost the same.

After generating the bounding box, it is inevitable that there are some pixels within the bounding box that do not belong to the two superpixels. To perform **blank space padding**, the naive solutions like padding with zeros or padding with the mean value will cause discontinuities and artifact edges, which violate the second rule. In this paper, we apply the technique called inpainting [18] to fill the blank regions by solving the following Laplace equation with two independent variables:

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = 0 \tag{1}$$

Given a region R, the Dirichlet problem tries to find the solution where the harmonic function φ equals to a function on the boundary of R. Thus, φ is dominated by the boundaries. In digital images, we can approximate the 2nd order partial differentiations in (1) by the following central difference operations:

$$\frac{\partial^2 u}{\partial x_{i,j}^2} \approx \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(\Delta x)^2}, \quad \frac{\partial^2 u}{\partial y_{i,j}^2} \approx \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{(\Delta y)^2} \tag{2}$$

If Δx and Δy are set to 1, the discretized form of the Laplace equation can be rearranged as follows:

$$-4u_{i,j} + u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} = 0\,. \tag{3}$$

For pixel (i, j) in an image of each channel, $u_{i,j}$ represents the color intensity of the pixel. The color intensities of the pixels surrounding the blank space are treated as the Dirichlet boundary condition. After performing inpainting by (1)-(3) together with the Dirichlet boundary condition, the blank regions are filled with the values come from the original superpixels. Furthermore, this will not generate artifact edges around the borders of the blank spaces.

With all the procedures described above, 710,000 two-superpixel patches can be extracted from 300 training and validation images in the BSDS500 dataset. Examples of the two-superpixel patches after padding are shown in Fig. 2.

B. Training Strategy

To achieve an even better segmentation result, we developed a two-stage superpixel merging process. We train two deep models with different amount of data. One applies balanced labeled data, that is, the numbers of training two-superpixel patches in two classes are roughly the same. The other one uses unbalanced labeled data and the two-superpixel patches labeled by "to be merged" is few times more than those labeled by "not to be merged". As a result, the first model performs merging cautiously to avoid over-merging. Then, the second model is adopted to obtain the final segmentation results.

C. Superpixel Pairing

To ensure that the segmentation result is edge-preserving, we apply some criteria to sift the superpixel pair that are impossible to be merged. These criterions are described as follows.

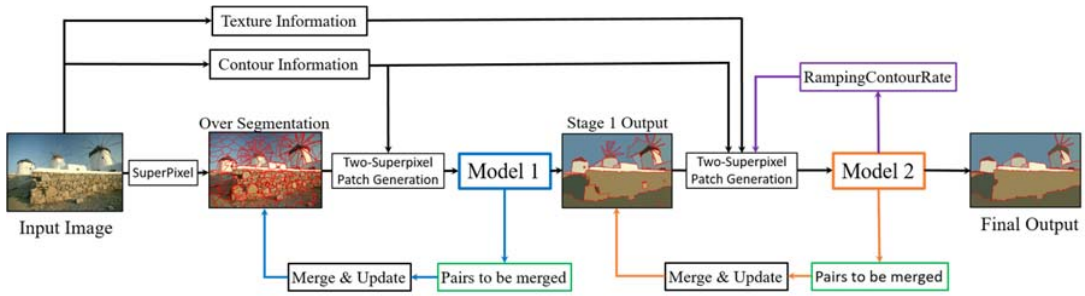

Fig. 3. Flowchart of the learning-based merging procedure.

The *ContourRate* is to indicate the percentage of pixels on boundary of two adjacent superpixels that have high responses for edge detection. If the *ContourRate* is high, it means that the boundary of the two adjacent superpixels may be the edge of some object and one should avoid merging the two superpixels. We apply the RefineContourNet (RCN) algorithm proposed by Kelm *et al.* [16] to generate the edge map of the input image and threshold it to get a binary contour map. Then, the *ContourRate* is defined as:

$$ContourRate(i,j) = \frac{\#\ of\ \ pixels\ of\ (long\ contours \cap Bnd(i,j))}{\#\ of\ \ pixels\ of\ \ Bnd(i,j)} \quad (4)$$

where long contours are the contour on the edge map with larger length and $Bnd(i, j)$ is the boundary between two adjacent superpixels i and j.

Moreover, in image segmentation, texture is one of the commonly used feature to compare the similarity between two regions. Smaller texture difference means that two regions are similar. Therefore, the Log-Gabor filter [17] is used to extract texture features:

$$G(f_x, f_y) = \exp\left(-\frac{\left(\log\left((f_x \cos\phi + f_y \sin\phi)/f_0\right)\right)^2}{2\left(\log(\sigma/f_0)\right)^2} \right). \quad (5)$$

Usually, two scales and four orientations are used to extract total of 8 texture images by changing the values of σ and φ. The difference of the texture is determined from:

$$dTex(i,j) = \sqrt{\sum_{k=1}^{8} (T_k(i) - T_k(j))^2} \quad (6)$$

where $T_k(i)$ and $T_k(j)$ are the means of the kth textures of adjacent superpixels i and j, respectively.

To avoid merging two superpixels that are connected by only a few pixels, we introduced the *ContactRate*:

$$ContactRate(i,j) = \frac{\#\ of\ \ pixels\ of\ \ Bnd(i,j)}{\min(BL(i), BL(j))} \quad (7)$$

where $BL(i)$ means the perimeter of superpixel i.

Moreover, traditional features like the area of regions and the color difference are also adopted. In the first stage, the criteria of the *ContourRate*, the *ContactRate*, and the area are applied with fixed thresholds, whereas in the second stage the *ContourRate*, *dTex* in (6), the *ContactRate*, the *area*, and the *color difference* are applied with adaptive thresholding.

After training the superpixel merging model, image segmentation is performed. As described in Section II-B, we apply a two-stage merging procedure. The initial superpixels are fed into Model 1. It aims to merge the adjacent superpixel pairs selected by the criteria in this section. Then, the output is fed into Model 2. It performs superpixel merging using adaptive criteria. The overview of merging procedure is shown in Fig. 3.

We call the proposed image segmentation algorithm the *deep merging model for superpixel-based segmentation* (DMMSS).

III. Experiments

We evaluate the proposed DMMSS algorithm on the popular Berkeley Segmentation Dataset 500, which consists of 500 color images. We split it into 200 *test images*, 200 *training images*, and 100 *validation images*. We extracted two-superpixel pairs on the 300 training and validation images and used them for training the network.

In the proposed architecture, the deep learning model of the ResNet101 [18] is adopted with the last fully-connected layer replaced by a layer two possible outputs. We used a pre-trained model on the ImageNet and fine-tuned the networks using mini-batches of 700 images with the initial learning rate of 0.0001. The learning rate was divided by 2 every 10 epochs and the training process stopped after 80 epochs. The binary cross entropy loss was used as the objective function and the Adam optimizer was adopted. A dropout layer with probability 0.5 was added to the networks to prevent overfitting during the training phase. Moreover, we adopted the early-stopping technique with validation the patience set to 400 iterations.

The source code of the proposed algorithm can be downloaded from the link in [13].

A. Performance Comparison

To compare the proposed DMMSS algorithm to existing methods, we evaluated the performance on the standard metrics of segmentation covering (SC) [19], the probabilistic rand index (PRI) [20], and the variation of information (VI) [21]. Higher SC and PRI values and a lower VI value mean better performance.

We compared the proposed algorithm to the state-of-the-art methods, including the methods of the W-Net [12], gPb-owt-ucm [3], DC-Seg-full [22], Taylor [23], Felzenszwalb and Huttenlocher (Felz-Hutt) [5], Mean Shift [2], Canny-owt-ucm [3], Normalized Cuts (NCuts) [4], fPb-owt-ucm [6], and cPb-owtucm [6].

TABLE I. COMPARISON FOR SEGMENTATION RESULTS.

Method	SC	PRI	VI
Ncuts	0.45	0.78	2.23
Canny-owt-ucm	0.49	0.79	2.19
Felz-Hutt	0.52	0.80	2.21
Mean Shift	0.54	0.79	1.85
Taylor	0.56	0.81	1.78
W-Net	0.57	0.81	1.76
fPb-owt-ucm	0.58	0.82	1.70
DC-Seg-full	0.59	0.82	1.68
W-Net+ucm	0.59	0.82	1.67
gPb-owt-ucm	0.59	0.83	1.69
cPb-owt-ucm	0.59	0.83	1.65
Proposed DMMSS(SEAL)	**0.62**	**0.85**	**1.50**
Proposed DMMSS(MS)	**0.63**	**0.85**	**1.51**
Proposed DMMSS(SSN)	**0.63**	**0.86**	**1.46**
Human Drawing	0.72	0.88	1.17

Table I shows the performance of the proposed DMMSS approach on the BSDS500 dataset. As the proposed algorithm, all the algorithms compared in Table I have not to assign the number of regions in prior.

From Table I, one can see that the performance of our proposed method is much better than that of state-of-the-art algorithms. One of the significant advantages of the proposed DMMSS algorithm is that it does not require lots of annotated training images to train a model. Compared to other learning based methods like the W-Net, which was trained on the PASCAL VOC2012 dataset that contains 11,530 images and 6,929 segmentations, better results can be achieved by the proposed DMMSS algorithm with almost 35 times fewer training images. About the computation time, when the superpixels of SEAL-ERS 100 is applied, the computation time of the proposed algorithm, DC-seg full, and gPb-owt-ucm are 15, 59, and 100 seconds, respectively.

Fig. 4 shows the segmentation results from the classical gPb-owt-ucm algorithm [3] and the proposed DMMSS algorithm. On can see that the proposed method can effectively merge background regions. These regions that are usually difficult to be merged using existing methods but can be successfully merged by the proposed r method. For example, in the 5th column, for the two people walking on the beach, the proposed algorithm can produce a segment that cover the whole person while gPb-owt-ucm produced a fragmentary result.

In Fig. 5, we show another visual comparison of the proposed DMMSS algorithm to DC-Seg-full [20], which is a famous learning-based method. Compare to the results of DC-Seg-full, the proposed method can produce more compact and reliable segmentation.

B. *Ablation Study for Different Contour Map*

In the proposed architecture, the RCN [16] is applied as the contour map, which plays an important role in the superpixel pairing procedure of Section II-C. The contour map highly

Fig. 4. Comparing the segmentation results. (Top) input images; (Middle) by gPb-owt-ucm [3]; (Bottom) by the proposed algorithms.

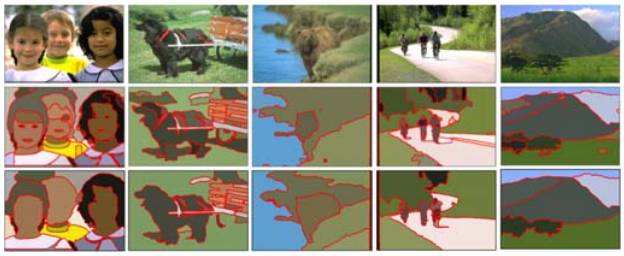

Fig. 5. Comparing the segmentation results. (Top) input images; (Middle) by DC-Seg-full [20]; (Bottom) by the proposed algorithms.

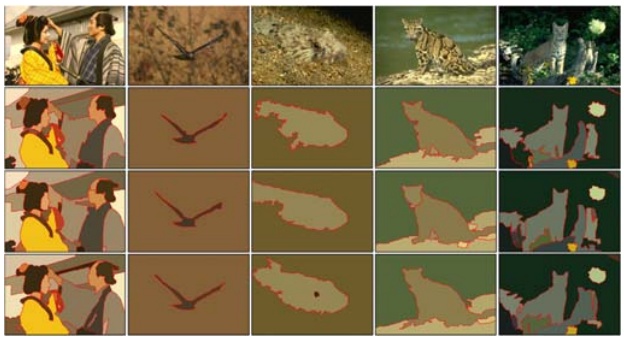

Fig. 6. Results of the proposed DMMSS algorithm using different superpixels. (1st row): original images; (Other rows): segmentation results produced by the proposed *DMMSS* algorithm using the superpixels generated by (2nd row): Mean-Shift; (3rd row): SEAL; (4th row): Superpixel Sampling Network (SSN).

TABLE II. RESULTS OF DIFFERENT CONTOUR/EDGE DETECTION.

Detection Method	SC	PRI	VI
Structure Edge	0.61	0.85	1.58
UCM	0.62	0.84	1.56
RCN	**0.63**	**0.86**	**1.46**

TABLE III. Results of Different Depth in the CNN.

Detection Method	SC	PRI	VI
ResNet18	0.58	0.84	1.72
ResNet50	0.57	0.84	1.71
ResNet101	**0.63**	**0.86**	**1.46**

affects the performance of image segmentation because vanishing of the contour might cripple the criterion, causing regions to be join undesirably. In this subsection, we further study the impact of different contour/edge detection algorithms on the proposed algorithm. In Table II, we reported the results that utilizing the classical contour detection of UCM [3] and the structure edge detector [24] instead of the RCN for contour map generation. As one can see, better segmentation results can be achieved if the RCN is adopted for contour map generation.

C. Ablation Study for Different CNN Architecture

We mainly implemented the proposed DMMSS algorithm by the ResNet [18] architecture. There are various versions of the ResNet with different numbers of layers within the networks. Therefore, we tested the proposed algorithms on the ResNet18, the ResNet50, the ResNet101 and showed the results in the right part of Table III. The results show that using ResNet101 can achieve the best performance.

D. Using Different Superpixels

In Table I, three different kinds of superpixels are adopted (SEAL [14], MS [2], and SSN [15] superpixels). Two of them (SEAL and SSN superpixels) are deep learning-based and adjustable to the number of superpixels.

In Fig. 6, we present the simulations that apply the proposed DMMSS algorithm to merge the superpixels generated from different algorithms. The results in Table I and Fig. 6 show that, no matter which type of superpixels is applied, with the proposed DMMSS algorithm, very high-quality segmentation results can be achieved.

IV. Conclusion

A novel image segmentation algorithm, *DMMSS*, that applies both deep learning architectures and superpixels was proposed in this work. The proposed algorithm converted the image segmentation problem into a series of decision problems about whether two adjacent superpixels should be merged or not. Since in the proposed algorithm the input of the network is a pair of adjacent superpixels and there are many adjacent superpixel pairs within an image, one can acquire huge amount of data to train the network and obtain highly accurate segmentation results. Experimental results showed that the proposed *DMMSS* algorithm outperforms state-of-the-art image segmentation techniques, including both learning-based and rule-based algorithms. Moreover, the proposed *DMMSS* algorithm is fully automatic and the number of regions has not to be assigned in prior.

References

[1] F. Y. Shih and S. Cheng, "Automatic seeded region growing for color image segmentation," Image and Vision Computing, vol. 23, issue 10, pp. 877–886, 2005.

[2] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, issue 5, pp. 603–619, 2002.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, issue 5, pp. 898–916, 2010.

[4] T. Cour, F. Benezit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," in Computer Vision and Pattern Recognition, vol. 2, pp. 1124–1131, 2005.

[5] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," Int. J. Computer Vision, vol. 59, issue 2, pp. 167–181. 2004.

[6] T. H. Kim, K. M. Lee, and S. U. Lee, "Learning full pairwise affinities for spectral segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, issue 7, pp. 1690–1703, 2012.

[7] Z. Li, X. M. Wu, and S. F. Chang, "Segmentation using superpixels: A bipartite graph partitioning approach," in Computer Vision and Pattern Recognition, pp. 789–796, 2012.

[8] Y. Yang, Y. Wang, and X. Xue, "A novel spectral clustering method with superpixels for image segmentation," Optik, vol. 127, issue 1, pp. 161–167, 2016.

[9] 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 Trans. Pattern Anal. Mach. Intell., vol. 40, issue 4, pp. 834–848, 2017.

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Computer Vision and Pattern Recognition, pp. 3431–3440, 2015.

[11] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in Int. Conf. Computer Vision, pp. 1520–1528, 2015.

[12] X. Xia and B. Kulis, "W-net: A deep model for fully unsupervised image segmentation," arXiv preprint arXiv:1711.08506, 2017.

[13] The source code is available from https://drive.google.com/drive/folders/16Q4zXldkUgnTQJI7lvouw6QcJ3YjNa82 .

[14] W. C. Tu, M. Y. Liu, V. Jampani, D. Sun, S. Y. Chien, M. H. Yang, and J. Kautz, "Learning superpixels with segmentation-aware affinity loss," in Computer Vision and Pattern Recognition, pp. 568–576, 2018.

[15] V. Jampani, D. Sun, M. Y. Liu, M. H. Yang, and J. Kautz, "Superpixel sampling networks," in European Conf. Computer Vision. pp. 352–368, 2018.

[16] A. P. Kelm, V. S. Rao, and U. Zolzer, "Object contour and edge detection with refinecontournet," in Int. Conf. Computer Analysis of Images and Patterns, Springer, pp. 246–258, 2019.

[17] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," JOSA A, vol. 4, issue 12, pp. 2379-2394, 1987.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Computer Vision and Pattern Recognition, pp. 770–778, 2016.

[19] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," in Computer Vision and Pattern Recognition, pp. 2294–2301, 2009.

[20] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: is a correction for chance necessary?" in: Proc. Annual Int. Conf. Machine Learning, pp. 1073-1080, 2009.

[21] M. Meilă, "Comparing clusterings—an information based distance," J. Multivariate Analysis, vol. 98, issue 5, pp, 873-895, 2007.

[22] M. Donoser and D. Schmalstieg, "Discrete-continuous gradient orientation estimation for faster image segmentation," in Computer Vision and Pattern Recognition, pp. 3158–3165, 2014.

[23] C. J. Taylor, "Towards fast and accurate segmentation," in Computer Vision and Pattern Recognition, pp. 1916–1922, 2013.

[24] P. Dollar and C. L. Zitnick, "Structured forests for fast edge detection," in: Int. Conf. Computer Vision. pp. 1841–1848, 2013.