2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506289

EDGE-AWARE SUPERPIXEL SEGMENTATION WITH UNSUPERVISED CONVOLUTIONAL NEURAL NETWORKS

Yue Yu, Yang Yang, Kezhao Liu*

School of Automation Science and Engineering
Xi'an Jiaotong University, Xi'an, Shaanxi, China

ABSTRACT

Superpixels provide an efficient representation of images, and are applicable for subsequent vision tasks. In this paper, we propose an edge-aware superpixel algorithm based on an unsupervised convolutional neural network (CNN). Noticing that to adhere the boundaries of objects is one of the most essential characteristics of superpixels, we propose an entropy-based edge-aware term, which helps fit the differential model of the pixel-superpixel soft-assignment matrix predicted from CNN to image gradients, i.e. generate boundary-aligning superpixels. The proposed algorithm yields more boundary-adhering superpixels, and experimental results on BSDS500 show the effectiveness of the proposed edge-aware term.

Index Terms— Edge-aware, Superpixel Segmentation, Unsupervised Convolutional Neural Networks

1. INTRODUCTION

Superpixels are non-overlapping image patches with low/mid-level semantics. Instead of redundant pixel representation, superpixel representation heavily reduces the number of image primitives, therefore it is more computational efficient to process superpixel-wise images.

Intuitively, as fundamental features, boundaries are important to discriminate objects. Observing superpixel algorithms in [1, 2, 3], we can deduce one principle: superpixels should have the ability to adhere to boundaries. Some algorithms have been proposed to satisfy this principle. Superpixel Lattices [4] over-segments images with a boundary map via minimizing boundary cost. BASS [5] sets more clustering centers on edges and extracts superpixels in *k*-means method.

In this paper, to tackle this principle, an unsupervised CNN is employed to yield superpixels. Inspired by [6, 7], we construct a CNN for processing a single image. In order to generate boundary-adhering superpixels, an edge-aware term is proposed to help train the CNN. Not requiring sophisticated edges, this term works only with image gradients and the pixel-superpixel soft-assignment matrix predicted by CNN. Since edges are on behalf of image gradients and able to be extracted by gradient operators, forward/backward difference on images can denote edges to some extent. Based on this view, we observe that forward/backward difference on soft-assignment matrix has similar variations to that of the corresponding image. Thus, it is rational to obtain boundary-adhering superpixels by fitting the differential model of soft-assignment matrix to image gradients. Noting that Kullback–Leibler divergence (KL divergence) measures the similarity between two distributions, we design this term in that way. Fig. 1 shows some superpixel segmentation instances. We will demonstrate the principal and the effectiveness of this term in following sections.

Fig. 1. Superpixel segmentation instances. Images from top to bottom are segmented into 200, 100 and 50 patches.

The remainder of this paper consists of following contents. Section 2 defines symbols to be used and introduce the fundamental clustering method for superpixel segmetation. Section 3 demonstrates the edge-aware mechanism and the corresponding framework for superpixel segmentation. In Section 4, we illustrate the experimental details and results, and we summarize our work in Section 5.

2. PRELIMINARIES

2.1. Notations

Let $I \in \mathbb{R}^{H\times W\times C}$ represent an $H \times W$ input with C channels (3 for RGB images), and $I_{h,w}$ or I_i represent the intensity vector of pixel i at (h, w). Let $P \in \mathbb{R}_{+}^{H\times W\times N}$ represents soft-assignment tensor, where a N-dimensional vector

*Corresponding author. E-mail: yyang@mail.xjtu.edu.cn. This work is supported by the National Key Research and Development Program of China under Grant No. 2018AAA0102500, and Natural Science Basic Research Program of Shaanxi (021JM-020).

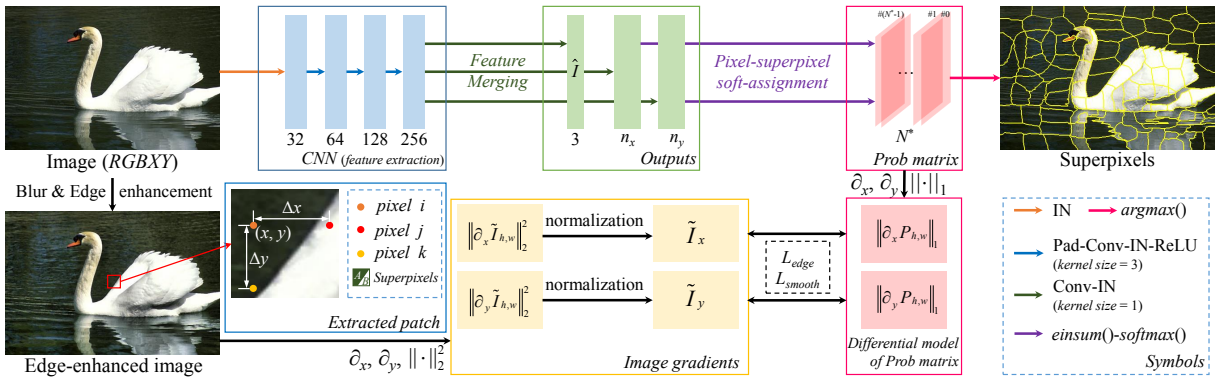

Fig. 2. Illustration of our framework and edge-aware mechanism. *IN* represents Instance Normalization, and values under the blocks denote the number of convolutional kernels, respectively. *#i* in *Prob matrix* represents superpixel *#i*, where the probability matrix *P* contains N^* channels, i.e. superpixels. The CNN only extracts features from a normalized *RGBXY* image. Edge-enhanced image is applied for calculating L_{smooth} and L_{edge}.

at (h, w) represents the probabilities that pixel at (h, w) belongs to N superpixels, and $P_{h,w}$ is a probability vector at (h, w) of P. Similar to I, $P_{h,w,n}$ or $P_{i,n}$ represents the probability that pixel *i* at (h, w) belongs to superpixel *n*. P can be reshaped into an $HW \times N$ matrix. Thus, we call tensor P as probability matrix or soft-assignment matrix for convenience. In addition, here is a constraint for P, i.e. $\sum_n P_{h,w,n} = 1$, indicating that the sum of probabilities that one pixel belongs to *N* superpixels is 1. These definitions are similar to [6].

2.2. Clustering method

One category of superpixel algorithms is based on clustering. Regularized Information Maximization (RIM) is an entropy-based clustering method, which has been proofed to be powerful to generate distinctive clusters [8].

To employ RIM, we regard it as a term of loss function to train the CNN. It denotes as:

$$L_{clustering} = \frac{1}{HW}\sum_{h,w}\sum_{n} -P_{h,w,n}\log P_{h,w,n} + \lambda\sum_{n} C_n \log C_n, \tag{1}$$

where $C_n = \frac{1}{HW}\sum_{h,w} P_{h,w,n}$, λ is a hyperparameter to control the amplitude of the second term. Minimizing the first term is to generate a sparse probability matrix, i.e. discriminative clusters; minimizing the second term, which represents cardinality, is to generate superpixels with uniform size.

3. EDGE-AWARE SUPERPIXEL SEGMENTATION

3.1. Edge-aware mechanism

Suppose here is a color image, and a pattern is extracted, as shown in *Extracted patch* of Fig. 2, where pixel *i* and *k* belong to superpixel *A*, and *j* belongs to *B*. Let pixel *i*, *j* and *k* be at (x, y), $(x + \Delta x, y)$ and $(x, y + \Delta y)$, respectively. Therefore, image gradients[1] at pixel *i* can be induced as $\partial_x I_i = ||I_i - I_j||_2^2$, $\partial_y I_i = ||I_i - I_k||_2^2$. Obviously, $\partial_x I_i$ is a large value, while $\partial_y I_i$ is close to 0. Let the index of superpixel *A*, *B* be a, b, respectively. So the probability differences at pixel *i* can be calculated as $\partial_x P_i = \sum_{s\in a,b} ||P_{i,s} - P_{j,s}||_1$, $\partial_y P_i = \sum_{s\in a,b} ||P_{i,s} - P_{k,s}||_1$. Since superpixels should adhere to boundaries, it is apparent that $\partial_x P_i$ should be larger, i.e. fit $\partial_x P_i$ to $\partial_x I_i$, indicating that pixel *i* and *j* belong to different superpixels. Similarly, $\partial_y P_i$ should fit to the corresponding $\partial_y I_i$ to show that pixel *i* and *k* belong to the identical region.

Based on these views, we propose a fitting principle: *forward/backward differential model of soft-assignment matrix should fit to image gradients*. To investigate how $\partial.P_i$ should fit to $\partial.I_i$, we define an indicator denoting as:

$$\mathbb{1}_{x,\theta} = \begin{cases} 1, & \text{if } x > \theta \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

where θ is a threshold. Only the amplitude of gradients greater than θ should be fit. Therefore, the edge-aware loss term denotes as:

$$L_{edge} = -\frac{1}{HW}\sum_{h,w}\Big(\mathbb{1}_{\tilde{I}_x,\theta}\tilde{I}_x \log||\partial_x P_{h,w}||_1 + \mathbb{1}_{\tilde{I}_y,\theta}\tilde{I}_y \log||\partial_y P_{h,w}||_1\Big), \tag{3}$$

where $\partial.$ denotes differential calculation. Since we observe that $||\partial_d P_{h,w}||_1$ ranges $[0, 2]$, we normalize $||\partial_d I_{h,w}||_2^2$ into $[0, 2]$ and define it as $\tilde{I}_d$, where $d \in \{x, y\}$. Observing that $\tilde{I}_d \log \tilde{I}_d$ is constant, we omit this term for convenience. Thus, edge-aware loss becomes a cross-entropy-like term de facto.

[1]Defining gradients in this way illustrates that the gradients are non-directional, so as $\partial_x P_i$ and $\partial_y P_i$.

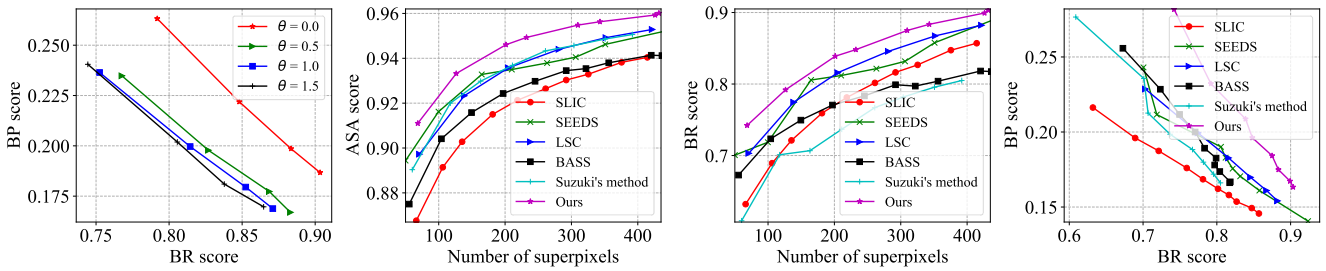

Fig. 3. Superpixel segmentation results on BSDS500. The first figure is PR curve of *Ablation study*; the second to fourth figures are ASA, BR and PR curve of *Comparison with baseline models*, respectively.

Another view is that $\tilde{I}_d$ can be regarded as *labels* derived from images, and minimizing this cross-entropy term is to make the distribution of $||\partial_d P_{h,w}||_1$ resemble that of $\tilde{I}_d$. It is similar to one-hot labels in supervised learning, while labels here are not manual-annotated. Therefore, the differential model of soft-assignment matrix is likely to approach the distribution of image gradients when minimizing this edge-aware term.

3.2. Framework

We design a multi-layer CNN to implement our superpixel algorithm, as shown in Fig. 2. Taking feasibility and efficiency into consideration, we make CNN generate n_x- and n_y-dimensional tensors, which represent the probabilities that one pixel belongs to horizontal n_x and vertical n_y superpixels, respectively. n_x and n_y is induced by the number of superpixels N, height and width of input image. In general, $N^* = n_x \times n_y \leq N$. It is convenient to employ `enisum()`[2] to combine them into an N^*-dimensional tensor.

As introduced in [7], CNN is a parameterized model of a desired output after training with a task-specific objective function. Based on this viewpoint, we define an overall loss function for superpixel segmentation. It consists of 4 terms:

$$L = L_{clustering} + \alpha L_{smooth} + \beta L_{recon} + \gamma L_{edge}, \quad (4)$$

where $L_{clustering}$ and L_{edge} have been induced in Eqs. 1 and 3, L_{smooth} and L_{recon} represent spatial smoothness loss and reconstruction loss, respectively. They have been introduced in [6, 9]. α, β and γ are hyperparameters controlling the strengths of these terms. CNN model should be optimized by minimizing this overall loss function to yield superpixels.

L_{smooth} and L_{recon} denote as:

$$L_{smooth} = \frac{1}{HW}\left(\sum_{h,w} ||\partial_x P_{h,w}||_1 e^{-||\partial_x I_{h,w}||_2^2/(2\sigma^2)} + ||\partial_y P_{h,w}||_1 e^{-||\partial_y I_{h,w}||_2^2/(2\sigma^2)}\right), \quad (5)$$

$$L_{recon} = \frac{1}{3HW}||I^{RGB}_{h,w} - \hat{I}_{h,w}||_2^2, \quad (6)$$

where σ is a hyperparameter that controls the local range for smoothing, $\hat{I}_{h,w} \in \mathbb{R}^{H\times W\times 3}$ is predicted from CNN, respectively. Minimizing L_{smooth} is to generate superpixels with smooth boundaries, and minimizing L_{recon} helps CNN extract more useful features from the input.

4. EXPERIMENTS AND RESULTS

The experimental results are yielded from the test set of BSDS500 [10], which contains 200 images. For each image, several manual-annotated labels are attached.

To demonstrate the effectiveness of our method, we compare our results with several unsupervised superpixel algorithms: SLIC [2], SEEDS [11], LSC [12], BASS [5] and the baseline method proposed by Suzuki [6], on the same dataset.

4.1. Metrics

We employ Achievable Segmentation Accuracy (ASA), Boundary Recall (BR) and Boundary Precision (BP) to evaluate the results and illustrate the effectiveness of our method[3]. ASA is a metric to calculate an upper bound of accuracy in superpixel-wise segmentation. The higher ASA indicates the segments are better. BR and BP measure the extent that the boundaries of superpixels align with that of groundtruth, and the corresponding PR (Precision-Recall) curve is better if it approaches upper-right-hand corner more, as introduced in [14]. For BR and BP, we set boundary tolerance $r = 2$ for all experiments. We utilize the metric scripts provided by [13].

4.2. Implementation details

We deploy our framework with PyTorch. To highlight edges, we employ filters provided in [15] and Contrast Limited Adaptive Histogram Equalization (CLAHE). So the input is an 8-dimensional tensor, where 3 for raw image, 2 for spatial coordinates and 3 for edge-enhanced image. In addition, each

[2]This function is contained in several frequently-used libraries and backends, e.g. PyTorch, NumPy and TensorFlow.

[3]More detailed definitions can be found in related works, e.g. [3, 13]

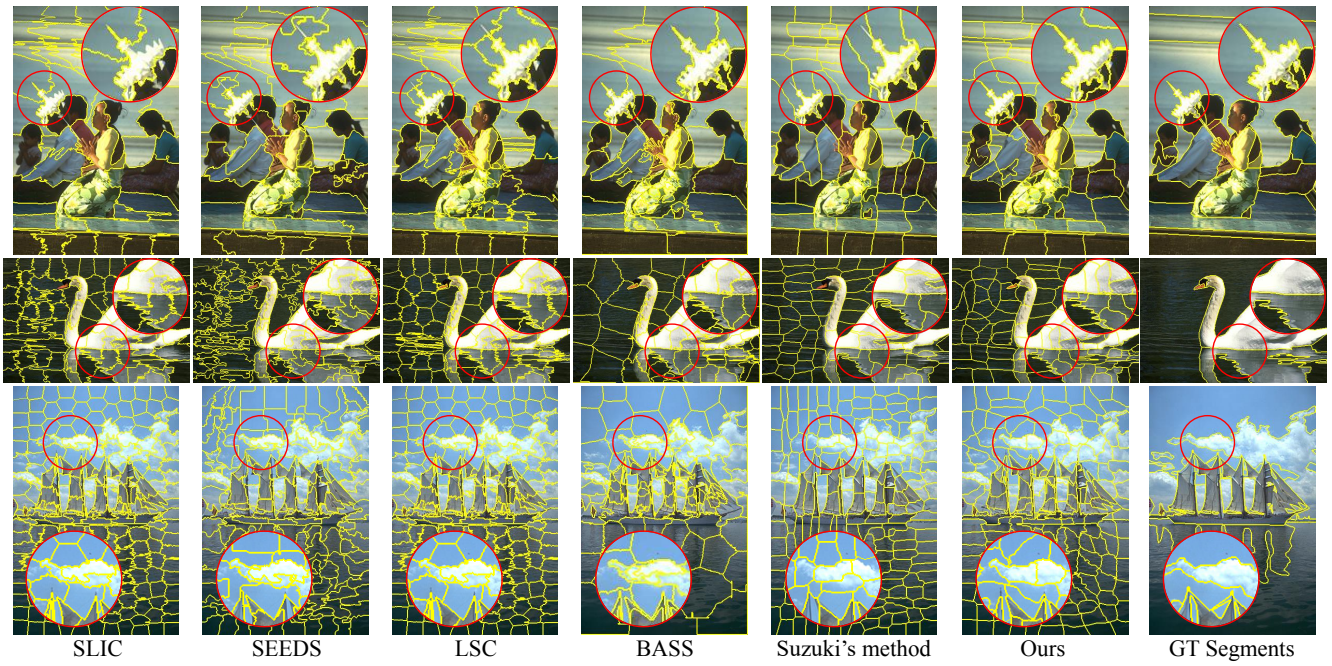

Fig. 4. Comparisons with GT segments and baseline superpixel algorithms. Rows from top to bottom are images divided into 50, 100 and 200 segments, respectively. Some details (highlighted with red circle) are zoomed in.

channel is normalized so that the mean and variance are 0 and 1. We optimize the model for 1,000 iterations with Adam in learning rate of 0.01. We set $(\alpha, \beta, \lambda, \sigma) = (2, 10, 1.5, 2)$ in Eqs. 1, 4 and 5. These are recommended in [6]. We set $\gamma = 1.5$ to balance L_{smooth} and L_{edge}[4].

4.3. Results

Ablation study. To investigate the extent that the differential model of probability matrix should fit to image gradients, we set $\theta = (0, 0.5, 1.0, 1.5)$. Generally, more detailed edges emerge when θ turns smaller. Here we measure the result via PR curve. As shown in the first figure of Fig. 3, the curve is more adjacent to upper-right corner when θ is smaller, indicating that superpixels adhere to boundaries better. It is acceptable that the boundaries of superpixels resembles that of groundtruth segments when more edges are emphasized. To this end, we set $\theta = 0$ as the baseline of our proposed framework and conduct subsequent experiments.

Comparison with baseline models. We compare our framework with aforementioned works via ASA, BR and PR curve. Since most models are well-performed when the number of superpixels is large, here we compare the performance of these models in ≤ 400 superpixels. The right 3 figures of Fig. 3 shows the score plots. Comparing with the baseline model proposed by Suzuki [6], BR score increases $\sim$10%, indicating that the edge-aware term is powerful to enforce superpixels to adhere to edges existing on images. Therefore, ASA score increases $\sim$1%. Benefiting from CNN with amount of low-level image features, our framework performs better than other models that yield superpixels in low-dimensional feature spaces, e.g. SLIC [2] in *LabXY* space.

Fig. 4 provides some instances. Comparing with other methods, BASS [5] and our method generate superpixels adhering to local boundaries better, since both exploit edges from images. In addition, SLIC, LSC [12] and Suzuki's method can generate regular superpixels. SLIC and LSC assign labels to pixels in a neighborhood region, and the second term of RIM is powerful to generate uniform superpixels. These ensure to generate regular superpixels. Since an edge-aware term is added, and most boundaries of objects are irregular, ours method yields more superpixels with inconsistent size and irregular boundaries than Suzuki's method does. It is adaptive to yield more regular superpixels by adjusting θ in Eq. 3 and λ in Eq. 1 larger, while superpixels are less boundary-adhering.

5. CONCLUSION

In this paper, we propose an edge-aware term and employ it in the superpixel algorithm based on an unsupervised CNN. Experimental results and comparisons demonstrate the effectiveness of our proposed algorithm.

One existing issue is that edge-aware superpixels are more irregular. We will explore the method to balance regularity and boundary adherence of superpixels in future works.

[4]Code is available at `https://github.com/yueyu-stu/EdgeAwareSpixel`.

6. REFERENCES

[1] X. Ren and J. Malik, "Learning a classification model for segmentation," in *IEEE International Conference on Computer Vision*, 2003, vol. 1, pp. 10–17.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[3] V. Jampani, D. Sun, M. Liu, M. Yang, and J. Kautz, "Superpixel sampling networks," in *European Conference on Computer Vision*, 2018, pp. 352–368.

[4] A. P. Moore, S. J. D. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[5] A. Rubio, L. Yu, E. Simo-Serra, and F. Moreno-Noguer, "BASS: Boundary-aware superpixel segmentation," in *International Conference on Pattern Recognition*, 2016, pp. 2824–2829.

[6] T. Suzuki, "Superpixel segmentation via convolutional neural networks with regularized information maximization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 2573–2577.

[7] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[8] A. Krause, P. Perona, and R. Gomes, "Discriminative clustering by regularized information maximization," in *Advances in Neural Information Processing Systems*, 2010, vol. 23, pp. 775–783.

[9] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 270–279.

[10] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[11] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, "SEEDS: Superpixels extracted via energy-driven sampling," in *European Conference on Computer Vision*, 2012, vol. 7, pp. 13–26.

[12] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1356–1363.

[13] D. Stutz, A. Hermans, and B. Leibe, "Superpixels: An evaluation of the state-of-the-art," *Computer Vision and Image Understanding*, vol. 166, pp. 1 – 27, 2018.

[14] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *International Conference on Machine Learning*, 2006, pp. 233–240.

[15] H. Yin, Y. Gong, and G. Qiu, "Side window filtering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.