Fast Superpixel Segmentation with Deep Features

Mubinun Awaisu[1], Liang Li[1,2(✉)], Junjie Peng[1], and Jiawan Zhang[1]

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
amubinun@yahoo.com, {liangli,jwzhang}@tju.edu.cn,
jiongjiongpeng@yeah.net

[2] Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering, China Three Gorges University, Yichang 443002, China

Abstract. In this paper, we propose a superpixel segmentation method which utilizes extracted deep features along with the combination of color and position information of the pixels. It is observed that the results can be improved significantly using better initial seed points. Therefore, we incorporated a one-step k-means clustering to calculate the positions of the initial seed points and applied the active search method to ensure that each pixel belongs to the right seed. The proposed method was also compared to other state-of-the-art methods quantitatively and qualitatively, and was found to produce promising results that adhere to the object boundaries better than others.

Keywords: Superpixel · Deep feature extraction · Active search

1 Introduction

Superpixels are becoming increasingly important in the field of computer vision. They are widely used in applications such as object detection [9], semantic segmentation [2], saliency estimation [3], and optical flow estimation [7]. Essentially, superpixel is a technique used to group image pixels into smaller sub-regions [1] based on the pixels similarity. State-of-the-art overview can be found in [8]. Superpixels can be used as the fundamental units instead of pixels to reduce computational complexity. Useful superpixels must produce high quality segmentations that adhere to the edges well. To fulfill this demand, researchers have tried many features. For instance, recently, the 5-D features consisting of the L, a, b values from CIELAB color space and the x, y pixel coordinate has been popular choice. However, relying only on appearance and spatial information is not enough to segment the edges accurately when the objects in the image share similar color with the background, which is very common when the scene is highly cluttered.

This work was supported by National Natural Science Foundation of China under Grants 61602338, Hubei Foundation for Innovative Research Groups under Grants 2015CFA025.

M. Gavrilova et al. (Eds.): CGI 2019, LNCS 11542, pp. 410–416, 2019.
https://doi.org/10.1007/978-3-030-22514-8_38

Fig. 1. Images segmented using our algorithm into 100, 200 and 400 superpixels. The superpixels are compact and adhere well to the region boundaries.

Therefore, researchers have recently resorted to the convolutional neural network (CNN) to extract features with more representability [5]. Due to the strong capability of CNN to learn the high-level representations for natural images, these methods often achieve better adherence to boundaries (Fig. 1).

It is important to note that superpixels are usually used as a preprocessing step in other applications. Therefore, running speed is a critical factor affecting the usefulness of superpixel algorithm. In this paper, we propose a superpixel segmentation algorithm that can produce superpixels with better boundaries adherence while being time efficient. There are two stages in our algorithm. In the first stage, we modified the initialization step in [10] by first measuring the distance between the seeds center and neighboring pixels. This makes each seed to contain almost similar pixels and also makes the active search less computational since most of the pixels are similar. Then we applied the active search to the initial superpixels to ensure that each pixel belongs to the right superpixel. The initial superpixels and modified active search approach have low computational cost and satisfy all the properties of good superpixels.

2 Proposed Method

In the proposed method, deep features were extracted from pre-trained network designed by [4] and concatenated with $LabXY$ for better representation of the image pixels. Superpixel Sampling Network (SNN) is mainly designed for feature extraction, and we extracted features for each image with a deep CNN originally trained over the BSDS500. Figure 2 shows an overview of the proposed method. Multidimensional vector is used to represent each pixel in our algorithm: $I_i = [l_i\ a_i\ b_i\ x_i\ y_i\ F_i]^T$, where $[l_i\ a_i\ b_i]$ is the pixel color vector in CIELAB color space, $F_i = [f_{i1}\ f_{i2}...f_{iT}]$ are the extracted features from deep network and $[x_i\ y_i]$ is the pixel position.

Details on getting the initial superpixels are explained in Algorithm 1. The nearest seed center is computed by a distance function D defined by the following equation,

$$D(I_i, S_k) = \sqrt{\lambda\left(d_c + \alpha d_s\right)^2 + d_F^2}, \tag{1}$$

where I_i represents the pixel, S_k represents the seed center, λ is the weight for controlling $LabXY$ and $\alpha = \frac{m}{N}$; m is the compactness variable and N is the

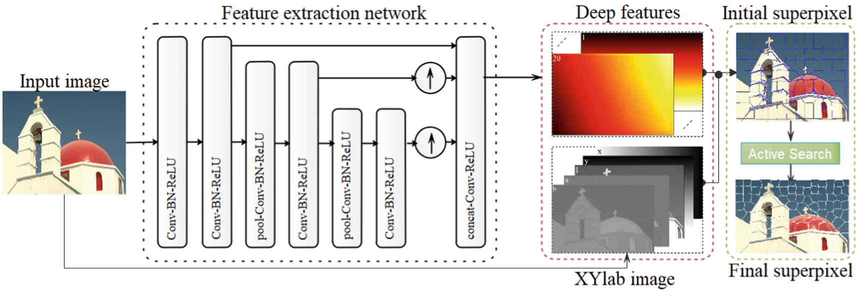


Fig. 2. Deep-FLIC flow chart. We used pre-trained network for extraction of features in each image, then we concatenated these features and the original image features in CIELAB color space. Finally we generated the initial superpixels and applied the active search method used in [10] to get the final superpixels segmentation.

number of pixels in an image. The variables d_c, d_s and d_F are the *lab* color distance, xy plane distance, and deep feature distance, respectively. They are defined by the following equations.

$$d_c = \sqrt{(l_i - l_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2}, \tag{2}$$

$$d_s = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \tag{3}$$

$$d_F = \sqrt{(F_i - F_j)^2} = \sqrt{\sum_{t=1}^{T} (f_{it} - f_{jt})^2}. \tag{4}$$

2.1 Active Search

The active search strategy enables each of the current pixel to actively search for the superpixel it should belong to, based on it's neighbouring pixels. We computed the distances between the current pixel and the seeds of its four adjacent pixels. The assignment principle for pixel I_i is given by the equation below,

$$L_i = \arg\min_{L_j} D(I_i, S_{L_j}), I_j \in A_i, \tag{5}$$

where A_i consists of I_i and its four neighboring pixels, S_{L_j} is I_j's corresponding initial superpixel seed. Equation (1) was used to measure the distance $D(I_i, S_{L_j})$.

The back and forth strategy [10] is applied to traverse the initial superpixels and get the pixels processing sequence. Minimum bounding box for all the initial superpixels is defined. The scanning process for all pixels in the corresponding minimum bounding box is performed, and then the pixels are processed within the initial superpixels. When the label of the current pixel L_i changes to any seed of its neighbouring pixels L_j, the seeds are updated instantly using the following equations,

$$S_{L_i} = \frac{S_{L_i} * |P_{L_i}| - I_i}{|P_{L_i}| - 1}, \tag{6}$$

Algorithm 1. Superpixel Initialization

Input: $I_i = [l_i, a_i, b_i, x_i, y_i, F_i]$ for each pixel.
Initialization: The initial seed points $S = [S_1, \cdots, S_k]$ and the centers in each of the seeds $S_k = [l_k\ a_k\ b_k\ x_k\ y_k\ F_k]^T$ with the corresponding feature F_{S_k};
Set label $L_i = -1$, distance $d_i = \infty$ for each pixel and $itr = 0$;
for *each seed center* S_k **do**
  for *each pixel in a* $2S \times 2S$ *region around* S_k **do**
    Compute the distance D between S_k and I_i;
    if $D < d_i$ **then**
      set $d_i = D$ and $L_i = k$
    end
  end
end
Output: Initial superpixels.

---

where $|P_{L_i}|$ is the number of pixels in the initial superpixel P_{L_i}, and S_{L_j} is updated using the below equation,

$$S_{L_j} = \frac{S_{L_j} * |P_{L_j}| + I_i}{|P_{L_i}| + 1}, \tag{7}$$

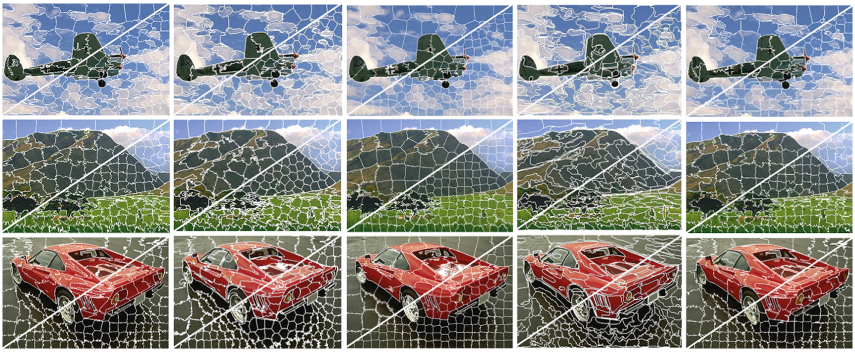
the bounding box is also updated. This process changes the seeds of the initial superpixels adaptively and allows the assignment and update to happen jointly.

3 Experiments

The proposed algorithm is implemented in C++ and runs on a PC with CPU, 4.0 GHz, 8 GB RAM, and 64 bit operating system. Our method is compared with existing four state-of-the-art algorithms, namely, FLIC [10], SLIC [1], LSC [6] and SSN [4]. Publically available implementations provided by the original authors are used for fair comparison. All the experiments are conducted on the BSD Berkeley Segmentation Dataset. This dataset consists of five hundred 321×481 images, together with human-annotated ground truth segmentations. The effectiveness of our method is demonstrated by providing visual and quantitative results with the existing superpixel methods. We experimented with the following default parameters: The number of superpixel, K, was set as desired, the spatial distance weight, $m = 5$ (as default), number of iterations, $itr = 2$, and $\lambda = 0.5$ (λ is a weighting value that balances the $LabXY$ features and deep features. Its value varies between [0; 1]).

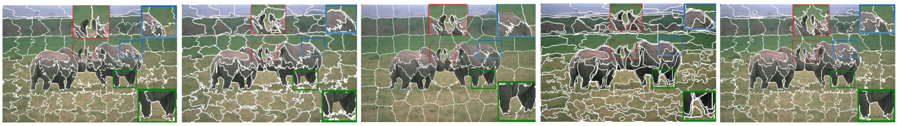
3.1 Visual Results

Figure 3 shows segmentation results of our method and the compared existing algorithms. Looking closely at the characteristics of good superpixel algorithm, it can be seen that, our method performed well compared to the competing



(a) FLIC [10] (b) LSC [6] (c) SLIC [1] (d) SSN [4] (e) Our result

Fig. 3. Visual comparison with SOTA algorithms with 100 and 300 superpixels.



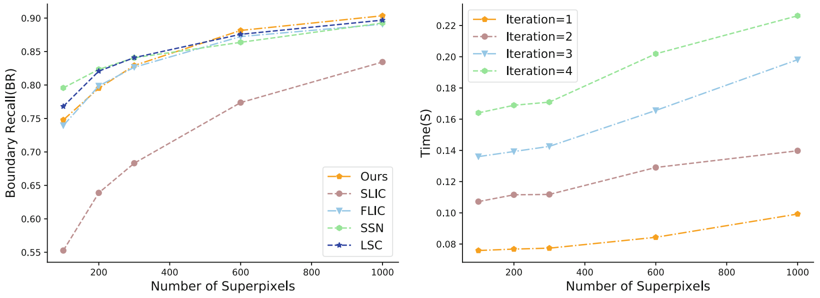
(a) FLIC [10] (b) LSC [6] (c) SLIC [1] (d) SSN [4] (e) Our result

Fig. 4. Superpixels segmentation results and magnified regions. The number of superpixels in all the results is 100.

state-of-the-art algorithms. Furthermore, in Fig. 4 more visual results with magnified regions can be seen, thereby indicating that our method obtained a certain improvement in boundary adherence compared to FLIC, LSC and SLIC methods.

3.2 Quantitative Results

The most important feature of good superpixel algorithm is the boundary adherence. To determine how well the superpixels adhere to the boundaries of an object, it is required to use some criteria for quantitative comparison. Boundary recall (BR) and under-segmentation error (UE) are the standard criteria for measuring the quality of the boundary adherence. Figure 5(a) shows a graph of the boundary recall as a function of the number of superpixels generated by the algorithms. Our method performs favorably in both higher and lower superpixel numbers. Under-segmentation error and Achievable segmentation accuracy of all the methods are illustrated in Table 1. Looking at Table 1, its clearly seen that SSN has the best UE of all the comparison algorithms. However our method is the best compared to FLIC, LSC and SLIC.



(a) (b)

Fig. 5. Quantitative comparison with the state-of-the-art methods. (a) Boundary recall vs superpixel number. (b) Time comparison in different iterations.

Table 1. Comparison for 1000 superpixels (BR, UE) and 300 superpixels (ASA).

Existing methods	FLIC [10]	LSC [6]	SLIC [1]	SSN [4]	OURS
Boundary Recall (BR)	0.890	0.897	0.834	0.892	**0.900**
Under-segmentation Error (UE)	0.181	0.252	0.200	**0.108**	0.154
Achievable Segmentation Accuracy (ASA)	**0.949**	0.923	0.925	0.947	0.948

4 Discussion

Good superpixel segmentation method should have high BR as well as low UE. It is seen from Table 1 that the method developed in this study performs well compared to several algorithms. In comparison to FLIC, LSC, and SLIC, our method has lower UE. This advantage is due to the addition of more features in to the distance measure. The BR measure of our method (Table 1) and Fig. 4 further indicate the favorable performance of our algorithm compared to other methods especially when the number of superpixels is high. However, SSN has the best BR below 600 supepixels. The initial superpixels introduced in our method increased the efficiency of the running time by making the active search less computational. It ensures that each of the initial superpixel contained almost similar pixels. In Fig. 5(b) we compared the time required for our method to generate superpixels for different iterations with different number of superpixels. When processing an image with 600 superpixels the time taken for first and second iterations are 0.0852 s and 0.125 s, respectively, which can be used efficiently in image preprocessing.

5 Conclusions

In this research, we proposed a modified superpixel algorithm, that takes into account images with weak object boundaries to increase the boundary adherence of the superpixels. FLIC algorithm is extended to incorporate deep features along with color and spatial properties of the pixels. Visual and quantitative results show that the developed method is able to generate more semantically-coherent superpixels compared to the other state-of-the-art methods. Future work will involve the use of end to end trainable deep network with the active search method to improve both the lower and higher superpixels and also reduce the computational time.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012)
2. Gould, S., Rodgers, J., Cohen, D., Elidan, G., Koller, D.: Multi-class segmentation with relative location prior. Int. J. Comput. Vision **80**(3), 300–316 (2008)
3. He, S., Lau, R.W.H., Liu, W., Huang, Z., Yang, Q.: SuperCNN: a superpixelwise convolutional neural network for salient object detection. Int. J. Comput. Vision **115**(3), 330–344 (2015)
4. Jampani, V., Sun, D., Liu, M.-Y., Yang, M.-H., Kautz, J.: Superpixel sampling networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 363–380. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_22
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 (2012)
6. Li, Z., Chen, J.: Superpixel segmentation using linear spectral clustering. In: CVPR (2015)
7. Lu, J., Yang, H., Min, D., Do, M.N.: Patch match filter: efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In: CVPR (2013)
8. Stutz, D., Hermans, A., Leibe, B.: Superpixels: an evaluation of the state-of-the-art. Comput. Vis. Image Underst. **166**, 1–27 (2018)
9. Yan, J., Yu, Y., Zhu, X., Lei, Z., Li, S.Z.: Object detection by labeling superpixels. In: CVPR (2015)
10. Zhao, J.X., Bo, R., Hou, Q., Cheng, M.M.: FLIC: fast linear iterative clustering with active search. In: AAAI (2018)